BÜYÜK VERİ İŞLEME VE ANALİZİNİN İLERİ TEKNOLOJİLERİ

Editör

Hasan YILDIZHAN

Çeviri Yazarları

Ayşe SEVİMLİ
Melek YOLCU

Yazarlar

Mihaela TINCA UDRIŞTIOIU
Adam DUDÁŠ
Alžbeta MICHALÍKOVÁ
Fatih KILIÇ
Önder TUTSOY
Jarmila ŠKRINÁROVÁ
Silvia PUIU
Slaveya PETROVA

Bu materyal, Hava kirliliği ile ilgili olarak öğretim ve araştırmada bazı ileri teknolojilerin uygulanması konusunda Erasmus+ projesi kapsamında Avrupa Komisyonu tarafından finanse edilmiştir.
Proje **Kodu: 2021-1-RO01-KA220-HED-000030286**

Avrupa Komisyonu'nun bu yayının üretimine verdiği destek, yalnızca yazarların görüşlerini yansıtan içeriğin onaylandığı anlamına gelmez ve Ulusal Ajans ve Komisyon, burada yer alan bilgilerin herhangi bir şekilde kullanılmasından sorumlu tutulamaz.

Avrupa Birliği Tarafından Finanse Edilmektedir

Craiova Üniversitesi

Paisiy Hilendarski Plovdiv Üniversitesi

Adana Alparslan Türkeş Bilim ve Teknoloji Üniversitesi

Banská Bystrica, Matej Bel Üniversitesi

ISBN
978-625-399-464-8

Kitap Adı
Büyük Veri İşleme ve Analizinin İleri Teknolojileri

Editör
Hasan YILDIZHAN
ORCID iD: 0000-0003-0272-980X

Proje Yöneticisi
Mihaela TINCA UDRISTIOIU
ORCID iD: 0000-0002-5811-5930

Yayın Koordinatörü
Yasin DİLMEN

Sayfa ve Kapak Tasarımı
Akademisyen Dizgi Ünitesi

Yayıncı Sertifika No
47518

Baskı ve Cilt
Vadi Matbaacılık

Bisac Code
BUS070030

DOI
10.37609/akya.2891

Kütüphane Kimlik Kartı
Tınca Udrıstıoıu, Mihaela ve diğer.
Büyük Veri İşleme ve Analizinin İleri Teknolojileri / Mihaela Tınca Udrıstıoıu, Adam Dudaš, Alžbeta Michalikova [ve başkaları] ; editör : Hasan Yıldızhan.
Ankara : Akademisyen Yayınevi Kitabevi, 2023.
171 s. : şekil, tablo. ; 195x275 mm.
Kaynakça var.
ISBN 9786253994648
1. Bilgisayar--Bilgi Teknolojisi.

GENEL DAĞITIM
Akademisyen Kitabevi A.Ş.
Halk Sokak 5 / A
Yenişehir / Ankara
Tel: 0312 431 16 33
siparis@akademisyen.com

www.akademisyen.com

İÇİNDEKİLER

GİRİŞ

Bu el kitabı, Erasmus+ projesi no. 2021-1-RO01 KA220-HED-000030286 kapsamında ortaya çıkan bir sonucu temsil etmektedir. Proje başlığı, "Hava kirliliği ile ilgili öğretim ve araştırma bağlamında bazı gelişmiş teknolojilerin uygulanması" olarak belirlenmiştir. Dört ortak (Matej Bel Üniversitesi - Banská Bystrica, Slovakya; Craiova Üniversitesi - Romanya; Paisii Hilendarski Üniversitesi - Plovdiv, Bulgaristan ve Alparslan Türkeş Bilim ve Teknoloji Üniversitesi - Adana, Türkiye) bu sonuca ulaşmak için birlikte çalışmıştır. Bu el kitabı, STEM (fen, teknoloji, mühendislik ve matematik) eğitmenlerinin öğrencilerin veri ile çalışma becerilerini geliştirmelerine yardımcı olmayı amaçlamaktadır.

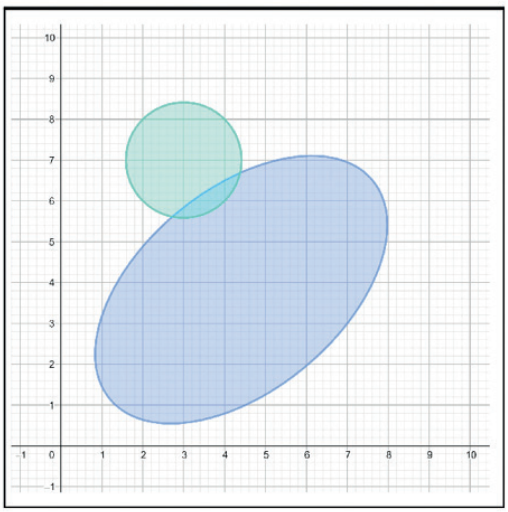
Çevremizde bulunan bilgi miktarıyla sıkça karşı karşıya kalıyoruz. Günümüzde her bir amaca yönelik olarak ilgili bilgileri çıkarmak için verileri işlemenin nasıl yapılacağını bilmek gereklidir. Her saniye, bilgisayarlar, sensör ağları ve uydular, fiziksel miktarlar ve parametreler için milyonlarca veriyi toplar. Veritabanları, verileri depolar ve düzenler, veri kalitesini artırır. Bilgi, her zamankinden daha fazla güçtür; bu noktadan hareketle, STEM öğrencileri veri ile nasıl çalışacaklarını öğrenmelidir. Şirketler, veritabanları tarafından verilen bilgilere dayalı sorunları çözebilecek uzman mezunlar sunabilmek için yüksek öğrenim gerektirir. Üniversitelerde, STEM öğrencilerinin veri setlerinin nasıl toplandığını, analiz edildiğini ve yorumlandığını öğrenmeleri gerekir. Ayrıca veri sınıflandırmaları, yaklaşımları ve tahminlerini yapmayı anlamaları gereklidir. Son olarak, İş piyasası STEM mezunlarından süreçlerin zaman ve mekanda nasıl gelişeceğini tahmin etmelerini veya kararlar vermelerini ister. Makine Öğrenimi ve Yapay Zeka, öğrencilerin günlük kelime hazinelerinin standart terimleridir.

Bu el kitabı on bölümden, eklerden ve referanslardan oluşmaktadır. İlk bölüm, farklı veri türleri, özellikleri, veri örnekleme yöntemleri ve verilerin nasıl işleneceği ve analiz edileceği hakkında bilgi içermektedir. Ardışık bölümler büyük veri setleri ile ilgili en önemli sorunlardan birine yaklaşmaktadır, yani veri analizi. Büyük verileri analiz ederken uygun istatistiksel analiz yöntemlerini, veri görselleştirmeyi ve diğer keşifsel, tahminsel ve tahmin yöntemlerini nasıl kullanılacağını bilmek gereklidir. Farklı bölümler, makine öğrenimi, bulanık çıkarım ve sinir ağı sistemleri gibi yaklaşımlara odaklanmaktadır. Ekler, Iris veri kümesinin açıklamasını, bazı sorunlara çözüm örneklerini, iklim değişikliği veya hava kirliliği ile ilgili veri setlerini ve hava kirliliğinin insan sağlığına etkisi hakkında bilgileri içermektedir. "Büyük veri işleme ve analizi için gelişmiş teknolojiler" adlı bir ders müfredatı örneği bu el kitabını kapatmaktadır.

BÖLÜM 1

VERİ VE ÖZELLİKLERİ

Bu el kitabının bu bölümü, Slovakya'nın Banská Bystrica kentindeki Matej Bel Üniversitesi, Fen Bilimleri Fakültesi, Bilgisayar Bilimleri Bölümü'nden Adam Dudáš tarafından yazılmıştır.



	A	B	C	D
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1
11	5.4	3.7	1.5	0.2
12	4.8	3.4	1.6	0.2
13	4.8	3.0	1.4	0.1
14	4.3	3.0	1.1	0.1
15	5.8	4.0	1.2	0.2
16	5.7	4.4	1.5	0.4
17	5.4	3.9	1.3	0.4
18	5.1	3.5	1.4	0.3

48° 44' 10.597" N 19° 8' 46.291" E

Bu el kitabı, yapay zeka, makine öğrenimi veya sinir ağları gibi bilgisayar bilimi tekniklerini kullanarak basit veri analizi yöntemlerini göstermektedir. Çalışmanın ilerleyen bölümlerinde, veriyle ilgili temel terimler ve kavramlar, verinin özellikleri, işlenmesi ve analizi ele alınacaktır.

Veri, teknik, istatistiksel, ekonomik veya diğer türdeki mesajlar veya bilgilerdir ki bu bilgiler teknik araçlar yardımıyla işlenebilir. Bizim durumumuzda, bu teknik araçlar bilgisayarlardır. Veriyi bir sistem içinde bütünleşik ve paylaşılan nesneler olarak ele alıyoruz:

- **Veri bütünleştirme** - Veri, tekrarı en aza indirgemek amacıyla birden fazla dosyaya yerleştirilebilir ve birden fazla dosyadan gelen verilere aynı anda erişilebilir hale getirilebilir.
- **Veri paylaşımı** - Her veri nesnesi, birden fazla kullanıcı tarafından (tekrar tekrar ve aynı anda) paylaşılabilir.

Verinin en önemli özelliği **sürekliliktir** - sürekli veri, program sonlandırıldıktan sonra bile varlığını sürdüren veridir. Giriş verileri sürekli veriye dönüştürülebilir, ve çıkış verileri sürekli veriden dönüştürülebilir veya ondan türetilebilir. Diğer verilerden türetilen veriler sürekli olmamalıdır (bu, sistem işletme maliyetlerini artırır) - ancak bazen bu gereklidir.

Verilerin kayıtlarda nasıl temsil edileceğine karar vermek, belirtilen türlere göre (mümkün olan en verimli depolamayı düşünerek) oldukça önemlidir. En yaygın **veri türleri** şunlardır:

- **Sayısal veri** - çeşitli şekillerde depolanabilir (ikilik, karakter, yarı-logaritmik form, ...). Genellikle sayı için gereken bit/byte sayısını belirlemek gereklidir.
- **Diziler** (Strings) - farklı karakter setlerinde depolanabilir (ASCII, UNICODE, EBCDIC, ...).
- **Numaralandırıcılar** - dizeler yerine karakter kodları kullanır (örneğin, mükemmel yerine A, ...).
- **Birimler** - belirli duruma uyacak şekilde ayarlanmalıdır (örneksiz: milimetre cinsinden ölçülen uçuş mesafesi).

Veri analizi bağlamında, bu veri türlerinin uygulanmaları bizim için pek ilginç değil gibi görünüyor. Genel olarak, içeriklerine göre ayrılan iki tür veriden bahsedeceğiz:

- **Sayısal veri, sayısal değerlerden (yükseklik, mesafe, sayı, vb.) oluşan verilerdir.** Bu tür veriler, matematiksel modellerde doğrudan kullanılabilir ve makine öğrenimi yöntemlerine dayalı veri analizi açısından kritiktir.
- **Kategorik veri, özelliklerin dil tabanlı tanımlamalarını** içerir (cinsiyet, renk, tür, vb.). Bu, belirli veri analizi yöntemlerinin gerekliliğini ima eder. Bazı kategorik veriler sayısal verilere dönüştürülebilir, ancak bu tür bir işlem her zaman anlamlı olmayabilir. Bu tür bir kodlamanın bir örneği, *cinsiyet = erkek İSE cinsiyet = 1, cinsiyet = kadın İSE cinsiyet = 2* gibi bir şey olabilir. Bu bir açıdan mantıklı olabilir, ancak böyle bir kodlamayı engelleyen bazı sorular vardır:

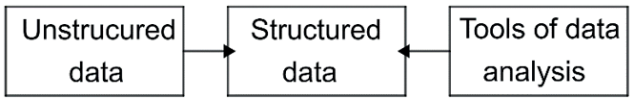
2 - 1 = 1 olduğuna göre kadın - erkek = erkek midir?
Cinsiyetin maksimum değeri nedir?

Herhangi bir veriyi analiz etmeden önce, onu düzgün bir şekilde yapılandırmak önemlidir. **Yapı açısından üç tür veriyi ayırt ediyoruz**:

- **Yapılandırılmış veri**, her kaydedilen nesne (satır) için aynı özellikleri aynı sıra (sütunlar) ile tanımlayabilen tablolarda veya bir dosyada saklanan veridir. En sık kullanılan yapılandırılmış veri formatları arasında csv (comma-separated values), Excel dosyası, basit metin dosyası veya SQL veritabanı bulunmaktadır.
- **Yarı-yapılandırılmış veri,** verinin yapısı vardır ancak bu yapının sabit olmadığı veridir. Bu nedenle her nesne (satır) için aynı özellikleri belirleyebiliriz, ancak bu özellikler her kayıt için kaydedilmemiş olabilir veya tüm kayıtlar için aynı sıra içinde olmayabilir (sütunlar belirlenemez). Bu, sensörlerin verileri, mobil uygulamalar ve benzeri kaynaklardan gelen veriler için tipiktir. Bu veri türü için kullanılan formatlar arasında XML, JSON veya MongoDB bulunur. Yarı-yapılandırılmış veri, belirli bir düzeni takip eder, ancak bu düzen kayıtlar arasında değişebilir veya eksik bilgiler içerebilir.

- **Çok yapılandırılmış/ya da Yapısız Veri** - farklı formatlardaki ham verilerdir. Örneğin, ham sensör verileri, web günlükleri, sosyal ağlardan gelen veriler, ses, video, görüntüler, 3D modeller, koordinatlar vb. gibi çeşitli formatlardaki ham veriler bu kategoriye girer. Bu tür veriler, belirli bir yapı veya düzen içermeyebilir ve farklı kaynaklardan elde edilen karmaşık bilgileri içerebilir. Bu tür veriler genellikle özel işleme ve analiz gerektirir çünkü yapıları belirsiz veya çok çeşitli olabilir.

  Yapısız veriyle çalışırken, genellikle aşağıdaki iş akışını kullanırız:



Örnek: Yapısız veri formundan yapılandırılmış veriye (tablo) dönüşüm. Üç öğrencinin temel bilgilerini içeren bir veri koleksiyonuna sahip olalım:

1. Martin, erkek, 28.6.1983, öğrenim yılı 2, 40 yaşında

2. Jane, 1994-9-13., K, 1. yıl st., 29 y.

3. Miriam, kadın, 5 Nisan 1992, öğreniminin ikinci yılı, yaş: 31

Bilgiler tutarsız bir biçimdedir - insanların bireysel özelliklerinin sırası farklıdır

öğrencilerin her biri için ve bireysel özelliklerin formatı da farklıdır. Veriler ayrıca kolayca hesaplanabilir fakat depolanmayan veriler içerir (yaş). Bu nedenle, verileri yapılandırılmış bir biçimde saklarken formda (tablo) tüm özelliklerin sırasını ve formatını birleştirmemiz gerekir (örneğin, YYYY-AA-GG formatındaki veriler).

Numele	***Data naşterii***	***Anul de studii***	***Sexul***
Martin	*1983-6-28*	*2*	*M*
Jane	*1994-9-13*	*1*	*F*
Miriam	*1992-4-5*	*2*	*F*

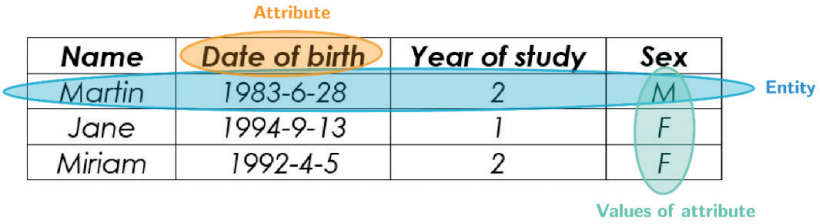
Bu el kitabı bağlamında, veri kümelerinde aynı nesneler için farklı başlıklar kullanacağız. Bu nedenle, aşağıdaki terimlerin kısa bir açıklamasını sunuyoruz.

Varlık (Entity), Nesne (Object) veya Kayıt (Record): Diğer nesnelerden açıkça ayırt edilebilen, bağımsız varlık yeteneğine sahip gerçek dünya nesnesidir.

Özellik (Attribute) veya **Mülkiyet** (Property): Bir varlığa bir değer atayan ve varlığın bazı temel özelliklerini belirleyen bir işlevidir (örneğin, yükseklik, yaş, ...).

Veri Kümesi (Dataset) veya **Tablo** (Table): Aynı özellik kümesinden oluşan varlıkların bir kümesidir.

Toplanan verilerin yapılandırılması, veri işleme işleminin temel bir yöntemidir (bkz. Bölüm 2). Bu terimler, veri yönetimi ve analizi süreçlerinde kullanılırken bir anlam bütünlüğü sağlamak için önemlidir.



1.1 BÜYÜK VERİ KONUSUNDA BİRKAÇ SÖZ

İlkesel olarak, az veri yerine çok veriye sahip olmak her zaman daha iyidir (bazı kayıtları her zaman atabiliriz). Verileri büyük olarak adlandırabiliriz, işlenmesi ve analiz edilmesi geleneksel araçlarla pratik bir süre içinde mümkün değilse. Tabii ki, sorular şunlar olabilir: *Pratik süre nedir? Geleneksel bir araç nedir?*

Bu nedenle, büyük verinin tanımını belirleyen özellikleri sıralayarak büyük veriyi tanımlıyoruz. Veri, genellikle 3V olarak adlandırılan özellikleri taşıdığında Büyük Veri olarak kabul edilir (bazı literatürlerde 5V, büyük verinin en temel görünüm modeli olarak kabul edilir):

- **Veri Miktarı** (Volume) - Birkaç kaynaktan elde edilebilecek veri miktarı, basit ilişkisel veritabanı modelleri kullanmayı düşünülemez hale getirir. Veriyi basit bir tablo veya tablo kümesi ile temsil edemez ve tek bir makinede çalışamazsınız. Bu nedenle, daha sofistike bir hesaplama altyapısı geliştirme ihtiyacı ve optimize edilmiş algoritmaları uygulama gerekliliği önem kazanır. Bu sistem, yüksek performanslı, dağıtılmış ve bulut hesaplama prensiplerini uygulamalı, homojen ve yüksek derecede bağlantılı verileri depolayabilen ilişkisel olmayan veritabanları ve veriyi işlemek ve analiz etmek için yapay zeka modellerini içermelidir.
- **Veri Çeşitliliği** (Variety) - Gerçek büyük veri sorunlarının ele alındığı durumda düşünülen veri kümesi homojen olmadığı için, birçok dosya türü ve formatı ile çalışabilmemiz gerekmektedir. Örneğin, basit metin belgeleri, ses dosyaları, video dosyaları, koordinatlar veya iki boyuttan daha fazlasını içeren bilgisayar modelleri gibi. Bu veri türlerinin çoğu, ilişkisel veritabanlarında saklanamaz ve daha sonra kullanmak üzere başarılı ve uygun bir şekilde saklanması ve işlenmesi için yüksek hesaplama gücü ve depolama alanı gerektirir.
- **Veri Hızı** (Velocity) - Büyük veri kümeleri genellikle canlı (veya dinamik) veri kümeleridir ve zaman içinde değişirler. Verinin zaman içinde değişimi, sürekli aktif olan ve veri toplayan bölgeler veri kaynaklarını uygulayan tüm sistemlerde meydana gelir. Bu tür kaynaklar, aynı veri kümesinin ilerlemesini ölçen Internet of Things (IoT) sensörleri gibi her zaman etkin olan ve veri toplayan kaynaklardır. Verinin canlılığı ve işlenmesi, depolanması ve analizi, sistem içine akış halindeki veri akışları oluşturur. Bu, büyük veri sistemlerinin temel gereksinimlerinden birini oluşturur - veriyi (neredeyse) gerçek zamanlı olarak toplama, depolama, işleme ve analiz etme yeteneği.

Canlı veri sorununun üzerine, büyük verinin dinamik ve statik kısımlarını birleştirerek sistemde birkaç sorunlu olayı tanımlayabiliriz.

Verinin miktarı, çeşitliliği ve hızının yanı sıra, kaynakların sayısı doğruluk (veracity) ve değer (value) ile çalışır:

- **Datanın doğruluğu:** Veri kaynaklarından gelen verilerin güvenilirliği ve kesinliği sorunlu olabilir. Veri kaynaklarından gelen bilgiler eksik, hatalı veya belirsiz olabilir. Bu nedenle, büyük veri sistemleri, güvenilirliği artırmak ve doğrulukla başa çıkmak için özel yöntemler ve veri kalitesi kontrolleri içermelidir.
- **Datanın değeri:** Büyük veriden elde edilen verilerin anlamlı içgörüler ve değer sağlaması önemlidir. Veri analizi ve işleme, verilerden anlamlı sonuçlar çıkarmak ve iş kararlarına yol göstermek için yapılmalıdır. Bu nedenle, büyük veri sistemleri, veriden değer çıkarma yeteneğini vurgulamalıdır.

Büyük verinin bu özellikleri, **işlenmesi ve analizi ile ilgili birçok sorunu ortaya çıkarır**

Bu sorunların ilki, **verinin kendi boyutudur**. Bu boyut, sadece verinin kendisinin depolanması için gereken bellek alanı bağlamında değil, aynı zamanda verilerde arama yapma ve analiz etme açısından da önemlidir. Bu tür verilerle çalışırken, bu veri türünden bilgi elde etmek için yüksek performanslı, dağıtılmış veya bulut hesaplama yöntemlerini, yapay zeka algoritmalarını (makine öğrenimi, bulanık çıkarım sistemleri ve sinir ağları gibi) kullanmak gereklidir. Büyük veri ile başa çıkmak için geleneksel tek bir makine yetersiz olabilir ve bu nedenle büyük veri işleme ve analizinde özel çözümler ve altyapılar gerekebilir. Büyük verinin boyutu, analitik işlemler için gelişmiş yöntemlerin ve teknolojilerin kullanılmasını gerektirebilir.

Bu büyük verilerin sadece büyük olması değil, **aynı zamanda genellikle farklı yönlerde farklılık gösteren heterojen bölümlerden oluştuğu bir gerçektir** - verinin boyutu, verinin bileşimi, verinin yapısı, aynı zamanda kullanılan ölçümler gibi birçok açıdan farklılık gösterebilir. Bu uyumsuzluk, büyük veri kümelerinin genellikle bir arşivde birleştirilen birkaç uyumsuz kaynaktan toplandığı gerçeğinin bir sonucudur. Bu nedenle, veri bölümlerinin yeniden biçimlendirilmesi gerekmektedir (veya daha kesin bir ifadeyle, bireysel veri biçimlerinin bir ölçüde birleştirilmesi gerekmektedir). Bu, veriler üzerinde gerçekleştirilmesi gereken basit görevlerin bir dizisini temsil eder (gerektiğinde ölçü birimlerinin dönüştürülmesi gibi), ancak aynı zamanda daha karmaşık görevleri de içerebilir. Örneğin, aykırı değerlerin ve eksik verilerin tanımlanması gibi. Eksik veriler durumunda, eksik verilerin hesaplanması için önlem alınabilir - değerlerin tahmini veya veri sınıflandırması için makine öğrenimi ve sinir ağları yöntemleri kullanılabilir.

Büyük veri kümelerinin heterojenliği ile yakından ilişkili olan sorun, **bu veri kümelerinin canlılığıdır**. Bu durumda, çevresel veri kaynaklarını (örneğin sensör ağları) kullanarak veri ölçümleri yapmaktayız ve bu ölçümler, küçük zaman aralıklarında yeterince fazla sensörde yapılır. Bu, sistem içinde işlenmesi ve analiz için hazırlanması gereken veri akışlarını oluşturur. Bu nedenle, bu sistem zaman içinde değişen veri kümelerini işleme yeteneğine sahip olmalıdır. Bu tür canlı veri akışları, büyük veri sistemlerinin özellikle veri akışlarını sürekli olarak yakalayabilme, depolama, işleme ve analiz etme yeteneğini gerektirir. Bu, gerçek zamanlı veri analizi ve işlemeye olan ihtiyacı vurgular ve büyük veri sistemlerinin bu tür verileri verimli bir şekilde işlemesi için özel olarak tasarlanmasını gerektirir.

Büyük veri kümeleri ile ilgili en önemli sorunlardan biri **verinin analizidir**. Analiz, yüksek performanslı, dağıtılmış veya bulut hesaplama, doğru problem ayrıştırma ve makine öğrenimi, bulanık ve sinir ağı hesaplama modelleri tarafından desteklenmelidir. Büyük veri kümelerini analiz ederken, istatistiksel analiz yöntemleri, verinin görselleştirilmesi ve veri keşif analizi veya makine öğrenimi, bulanık çıkarım sistemi veya sinir ağı yaklaşımları kullanarak keşifsel veri analizi, öngörü ve tahmini veri analizi gibi diğer yöntemler kullanabiliriz (Bölüm 2'ye bakınız).

1.2 VERİ SETLERİNDE ORTAK SORUNLAR

Büyük veri ile ilgili özellikle bahsedilmemiş bazı yaygın sorunlar aşağıda belirtilmiştir:

Yukarıda belirtildiği gibi, veri miktarı sürekli olarak arttıkça, depolanan ve işlenen verilerin analizi daha uzun sürer. Ancak analiz, veri depolamanın kendisi için esastır ve bu nedenle kaçınılmazdır. Bu, büyük veri kümesi bağlamında bilgi edinmemizi ve karar verme sürecini desteklememizi gerektirir. Verileri analiz etmek için yüksek performanslı, dağıtılmış veya bulut hesaplama yöntemlerine ihtiyaç vardır.

Belirli görevler için oluşturulan veri kümesi genellikle birkaç kaynaktan gelen verilerin birleştirilmesiyle oluşturulur. Bu kaynaklar, bireysel veri birimlerinin biçim ve bileşimi açısından çeşitlilik gösterebilir. Bu nedenle, bu çeşitli verileri toplamak ve birleştirmek için bir yol gereklidir. Farklı kaynaklardan gelen verilerin birleştirilmesi durumunda, bazen bireysel kayıtların birbirine ters düşebileceği (veya birbiriyle tutarlı olmayabileceği) bir durum ortaya çıkabilir.

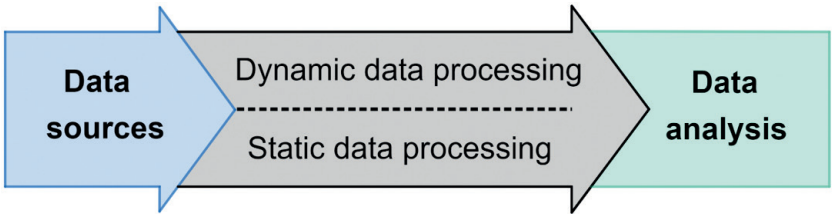
Veri güvenliği büyük bir sorundur, ancak bu elkitabının bağlamında temel bir öneme sahip değildir. Her veri kaynağı güvenli değildir ve hatta kullanmak isteyen şirketin politikasına uygun olmayabilir. Genel olarak, veriye erişim yetkilendirme ve kimlik doğrulama, veri üzerinde çalışan kullanıcıların izlenmesi, ham ve edinilen verilerin güvenliği ve iletişimin korunması, yani veri transferi, dikkate alınması gereken konular arasındadır.

Tahminler veya tahminler oluştururken özellikle önemli olan sorunlar eksik veri ve veri aykırılarıdır. Eksik veri sorunu, verilerde gereken ölçülen değerlerden birinin eksik olduğu durumlarda ortaya çıkar. Aykırı değerler durumunda, bazı ölçülen değerlerin veri kümesinin ana gövdesinin çok dışında olduğu doğal bir durumdur. Bu iki sorun, bu elkitabının Bölüm 2'sinin ana konusudur.

BÖLÜM 2

VERİ İŞLEME VE ANALİZİ

Bu elkitabının bu bölümü, Slovakya'nın Banská Bystrica şehrinde bulunan Matej Bel Üniversitesi Doğa Bilimleri Fakültesi Bilgisayar Bilimleri Bölümü'nden Adam Dudáš tarafından yazılmıştır.



Veriyle çalışırken, bu elkitabının bakış açısından en azından iki ana etkinlik belirleyebiliriz: veri işleme ve veri analizi. Ana hedefimiz, veri analizi için giriş olarak uygun veri olmadan etkili bir şekilde gerçekleştirilemeyen bilgi ve veri analizi örneklerini sunmaktır. Bu nedenle veriyi analiz için uygun bir forma hazırlama işlemine, herhangi bir sorun olmadan veri işleme adı verilir. Veriyi analiz için uygun hale getirme ihtiyacı, modern verilerin birkaç özelliğinden kaynaklanır:

- **Veri Kaynakları**: Günümüzde, çeşitli kaynaklardan toplanan daha küçük veri kümelerinin birleştirilmesi ile oluşturulan veri kümeleri ile çalışmak yaygındır. Bu şekilde üretilen veri kümeleri farklı veritabanlarından, aynı ağdaki sensörlerden veya bu iki yaklaşımın bir kombinasyonundan kaynaklanabilir. Bu şekilde oluşturulan veri kümesi, bu veri yapısı türü ile ilişkilendirilen çok doğal sorunları beraberinde getirir:
  - Veri yapısının homojenleştirilmesi,
  - Eksik verilerle çalışma,
  - Aykırı değerlerle çalışma.
- **Statik Veri İşleme**: Modern sistemlerde, belirli bir sistemde görünebilecek veya sistemde işleyebileceğimiz iki tür veriyi algılarız - ilk tür veri statik veridir. Statik veri, zaman içinde değişmeyen verilerdir. Bu tür verileri işlemek için yaygın bir yaklaşım, batch işlem olarak adlandırılan yaklaşımdır.

Bu işlem setleri varsayılan olarak bir dosyanın yüklenmesini, dosyanın işlenmesini ve kullanıcının manuel müdahalesi olmadan yeni bir dosyaya çıktı yazmayı içerir.

- **Dinamik veri işleme** - sistem ambiyans veri kaynaklarını (sürekli aktif sensör veya bir dizi sensör gibi) kullanıyorsa, bu veriyi neredeyse gerçek zamanlı olarak yakalayabilmeli ve depolayabilmelidir. Bu verinin depolanma türü önemli değildir - büyük veri hacmi nedeniyle ölçeklendirmeyi desteklemelidir. Bu tür dinamik değişen veri kümelerine veri akışları adını veririz ve bunları işleyebilmeli, süzebilmeli, birleştirebilmeli ve analiz için hazırlayabilmeliyiz. Bu şekilde işlenen veriler daha sonra analiz için gönderilir. Dinamik veri işleme sorunu bu elkitabının kapsamının ötesindedir, ancak modern veri sistemlerinin oldukça önemli bir parçasıdır.
- **Veri analizi** - veri analizini, belirli bir problem alanı bağlamında daha iyi karar verme ihtiyacı, toplanan verilere dayalı olarak değerlerin tahmin edilme olasılığı veya ölçülemeyen verilerin tahmin edilme olasılığı için bilgi edinme etkinliği olarak görüyoruz. Bu bölümün ikinci kısmında, veri analizi türlerini ve veri analizi ile ilgili ana sorunları açıklıyoruz.

Bu elkitabı bölümü, veri işleme ve bu eylemle ilgili seçilmiş sorunlara odaklanmıştır. Bölümün ikinci kısmı, daha sonra sunulan elkitabının sonraki bölümlerinde daha ayrıntılı olarak ele alınan veri analizi konusuna odaklanır.

2.1 VERİ İŞLEME

Her zaman doğrudan veri analizi için hazır bir veri kümesiyle çalışma fırsatına sahip olmayabiliriz. Sıklıkla (özellikle kendi verilerimiz söz konusu olduğunda), bu kelimenin tam anlamıyla toplanmış bilgi kümeleridir. Bu nedenle, veriyi analiz etmeden önce veriyi **temizlemek** ve **biçimlendirmek** gereklidir.

Bu süreç hakkında bir not - veri işleme ve metinde bu bölümde açıklanan tüm adımlar her zaman orijinal veri kümesinin bir kopyası üzerinde gerçekleştirilmelidir, veri kümesi üzerinde değil. Ayrıca, mümkünse sistematik ve tekrarlanabilir yöntemler kullanmalıyız. Sonuçta, kazanılmış verilerimizi kaybetmek istemeyiz.

Veri kümesinin iç tutarlılığı

Bu elkitabının bu bölümünün başında belirtildiği gibi, modern veri kümelerinin birleştirilerek oluşturulması, veri kümelerinin kendi iç tutarlılığı ile ilgili sorunlar yaratır. Bu tutarsızlık iki düzeyde algılanabilir - verinin kendisinin tutarsızlığı ve veri kümesi yapısının tutarsızlığı.

Varsayılan olarak, birkaç tipik sorun veri tutarsızlığına neden olabilir:

- **Birim dönüşümleri** - farklı birimleri kullanarak özellik değerlerini ölçen iki veri kümesini birleştirirken (örneğin, santimetreler ve milimetreler), ölçüm birimini birleştirmek gereklidir. Aynı miktarı ölçmek için farklı ölçüm birimlerini kullanan kıtalar üzerinde ölçülen veri kümelerini birleştirmek de önemlidir örneğin, aynı miktarı ölçmek için Avrupa'da santimetreler ve ABD'de inçler kullanılacaktır.
- **Sayısal dönüşümler** - sözlü olarak kaydedilen sayısal değerlerin sayılara dönüştürülmesi gereklidir. Bu dönüşümlerin gerekli olduğu alan, bir özellik değeri içinde birimleri belirleyen tipik sorunları içerir.

- **İsim dönüşümleri** – Kişilerin isimlerini kaydederken adlar ve soyadların kayıt şekillerini birleştirmek gerekir. Ad niteliklerini kullanan veri kümeleri durumunda en büyük sorun farklı kıtalar aksanlı karakterlerdir (örneğin, š, č, ä).
- **Tarih ve saat dönüşümleri** – zaman bilgisi içeren analizler durumunda, zaman kayıt formatını, özellikle de verilen veri kümesindeki tarihi birleştirin.
- **Finansal ve para birimi dönüşümleri** – veri kümesinde halihazırda mevcut olan para birimlerinden birine farklı para birimlerinde listelenen değerler eklenmelidir.

Başka bir tutarsızlık türü de veri kümesi yapısının tutarsızlığıdır. Veriyi daha sonra analiz için saklamanın ideal yöntemi, bu elkitabının 1. bölümünde açıklandığı gibidir - veriyi bir tablo olarak saklamak. Ancak, bunu basitçe elde etmek her zaman mümkün değildir - bu durumda sorun çoğunlukla eksik veriler olacaktır.

Eksik veriler içeren bir veri kümesi, standart araçlar ve yazılım araçları kullanılarak analiz etmek için sorunlu olabilir. Değerin eksik olduğu sanal tablonun hücreleri, değeri istatistiksel olarak değerlendiremeyen ve aynı zamanda kendisi olarak alınamayan *NULL* değerleri ile doldurulur (*çünkü 0 ≠ NULL*). Bu nedenle, bu tür bir sorunla belirli bir şekilde başa çıkmak gereklidir.

Eksik ve hasarlı veriler

Bu elkitabı için veriler, gerçek dünya özelliklerinin ölçümleri olarak görülmektedir. Bu ölçümler iki faktörden etkilenir: veri toplama aracı ve veri işleme yöntemi. Her iki faktörde de bir sorun ortaya çıkabilir ve bunun sonucu **veri kaybı veya veri hasarı** olabilir. Eğer veri toplama aracında bir sorun varsa (sensörün yanmış bir kısmı, sunucu arızası sonrası kayıplar vb.), bu veri kaybından bahsediyoruz ve bu kayıplar genellikle yeniden oluşturulamaz. Buna karşılık, verilerin işlenmesi sırasında veri kaybı veya hasarı meydana gelir. Ham veri mevcutsa, hatayı düzeltmek sorun değildir - bu tür veri kaybına veya hasarına bir sanat eseri adını veriyoruz.

Eğer veri kümemiz eksikse, eksik değerleri tanımlamak ve uygun şekilde telafi etmek gereklidir. Sorun, eksik değerlerin bazılarının belki de hiç var olmaması olabilir. Bir örnek, belirli bir konumda varış zamanını içeren bir özellik için bir değer olabilir, ancak henüz o konuma varmadığımız bir durumda bu tür bir değer eksik olacaktır.

Ham veriler mevcut olmadığında eksik değerlerle çalışma yöntemleri birkaç tür telafiye ayrılabilir:

- **Eksik değerleri başka bir değerle (0 / -1 / anlamsız) değiştirme** - bu yaklaşımla, her eksik değeri (NULL) seçilen özel bir değerle değiştiririz. Bu yaklaşım önerilmez - yerine geçen değerler sıklıkla doğru kabul edilebilir ve veri kümesinin analizinde yanlış yorumlanabilir. Örneğin, bir çalışanın maaşı için belirli bir değerimiz yoksa, bunu 0 veya -1 değeriyle değiştirmeyiz, çünkü çalışan ücretsiz çalışmaz veya işe gelmek için ödeme yapmaz
- **Eksik varlıkları atma** - bir öncekine göre biraz daha iyi bir durum olabilir, eksik her kaydı veri kümesinden kaldırdığımız yaklaşım olabilir. Bu yaklaşım, yeterli veriye sahipsek kabul edilebilir ancak yine de yanıltıcı sonuçlara yol açabilir.
- **Eksik değerlerin hesaplanması** (tamamlama) - eksik değerleri içeren kayıtları kullanmamız gerekiyorsa, bu değerleri aşağıdaki yöntemlerden birini kullanarak hesaplayabiliriz. Bu yaklaşıma değer tamamlama da denir.
    - Eğer veri kümesi ve ilişkileri hakkında yeterince bilgi sahibiyseniz, bazı özelliklerin değerini tahmin edebilmelisiniz. Bu durumda, sezgisel yaklaşımı kullanarak eksik değerleri doldurabilirsiniz.

- Eksik değerleri, verilen özellik için ortalama değerle doldurma - bu yöntem, eksik değerleri verilen özelliğin ortalaması ile değiştirir. Bu tür bir değeri kullanmanın birkaç nedeni vardır, en önemlisi, özelliklerin ortalamalarının her iki yönde de güçlü olmaması ve bu nedenle veri kümesinin tahmin potansiyeline çok az etkisi olmasıdır. Ancak eksik değerleri verilen özelliğin ortalaması ile değiştirmek her zaman uygun değildir. Bu yaklaşım, ortalama bir maaş için uygun olurdu, ancak belirli bir konumda ortalama varış tarihi mantıklı değildir.
- Özelliğin rasgele bir değeri ile tamamlama - eksik değer için veri kümesinde kaydedilen verilen özelliğin rasgele bir değerini seçeriz.
- Makine öğrenimi yöntemlerini kullanarak tamamlama - eksik verilerin hesaplanmasının en sofistike yaklaşımı, makine öğrenimi yöntemlerini kullanmaktır. Ancak bu yöntemler her veri kümesiyle kullanılamaz - veya daha doğru bir ifadeyle, her veri kümesi için etkili bir şekilde kullanmak mümkün değildir. Makine öğrenimi yöntemleri, veri kümesindeki bireysel değerler arasındaki ilişkilere dayalı olarak çalışır. Bu ilişkiler zayif veya yoksa, veri kümesi değerlerinin tahminleri yanlış olacaktır. Bu yaklaşım, bu elkitabının 4. bölümünden itibaren daha ayrıntılı olarak açıklanmıştır.

Aykırı Değerler

Aykırı değerler, veri kümesinin vücudunun dışında bulunan değerlerdir. Normal dağılım gösteren bir veri kümesinde, veri kümesinin ortalama değerinden uzaklık arttıkça bir değerin veri kümesinde bulunma olasılığı azalır. Ancak normal olmayan bir dağılıma sahip veri kümeleriyle sorunlar ortaya çıkar.

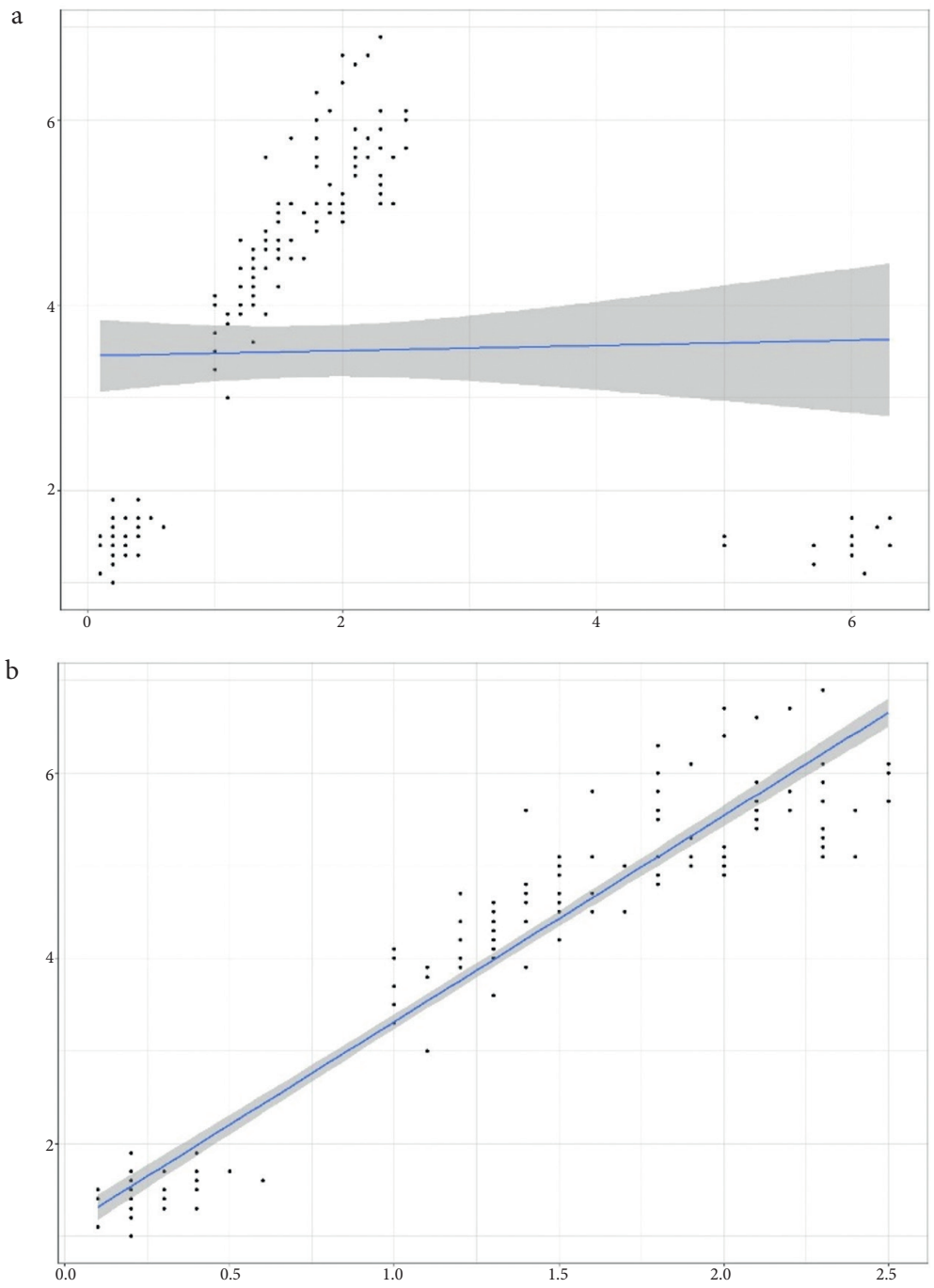
Aykırı değerler birkaç şekilde ortaya çıkabilir:

- Bir ölçüm hatası,
- Veri işleme sırasında yazım hatası,
- Güvensiz bilgi ki bu daha fazla güvensiz kaydın mevcudiyetini gösterebilir.

Ancak, bu genellikle standart durumlardan sapmış gerçek bir değer olabilir (örneğin, hava kirliliği dönemleri gibi), bu nedenle kaydın tamamını analiz etmek gereklidir.

Aykırı değerlerle ilgili sorun, aykırı değerler içeren verilere dayalı herhangi bir **genelleme** yapmaya çalışırken ortaya çıkar. Aşağıdaki şekil, verilen veri kümesini düz bir çizgi kullanarak açıklama girişimini göstermektedir. Sol tarafta, veri kümesi içeren 162 kayıttan oluşan bir veri kümesi bulunur, bunların 12'si veri kümesinin vücudunun dışında önemli bir şekilde bulunur. Bu durumda, veri kümesinin merkezinin geçmesi gereken mavi çizginin, bir nokta hariç tamamen kaçırdığını görüyoruz. Sağ tarafta, verilen on iki aykırı değeri çıkardıktan sonra aynı veri kümesini görebiliriz. Bu durumda genelleme sonucu çok daha tatmin edicidir.

Veri kümesi üzerinde herhangi bir genelleme yapmak istiyorsak, aykırı değerler rahatsız edici bir etken olarak hareket edeceklerdir ve bu nedenle bu tür özellik değerlerini (ve içerdikleri kayıtları) hesaba katmamak önerilir, hatta doğru olsalar bile. Aşağıdaki şekilde görüleceği gibi - bir veriyi bir çizgi (aslında doğrusal bir fonksiyon) kullanarak açıklamak istiyoruz. a alt şeklinde, aykırı değerlerin varlığı nedeniyle çizgi sapmıştır (düşünülen uzayın sağ alt köşesi). Bu aykırı değerleri kaldırdıktan sonra, bu genelleme doğruluğunda drastik bir artış görüyoruz (b alt şekil).



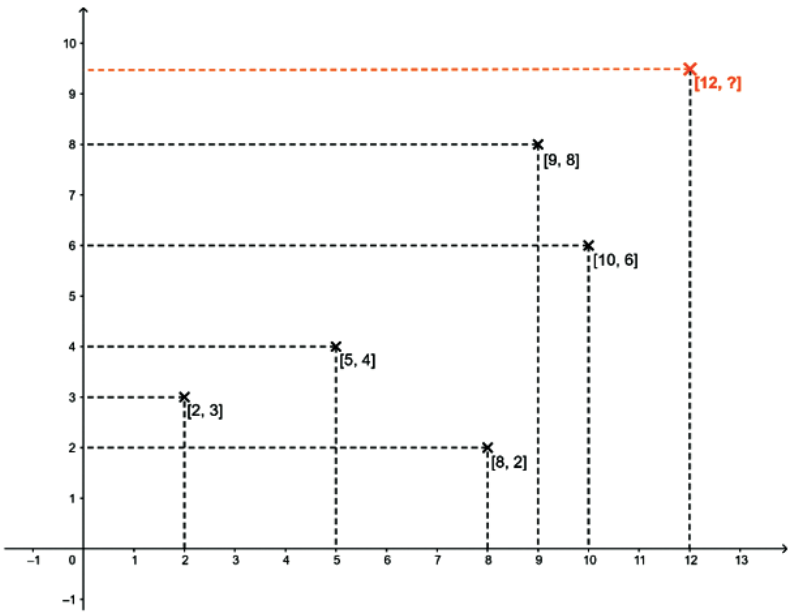
2.2 VERİ ANALİZİ

Veri analizi, işlenmiş verilere dayalı olarak sorun hakkında bilinçli karar verme, olayların tahmin edilmesi ve seçilen nesnelerin davranışı hakkında **yararlı bilgi elde etmeyi** amaçlar. Birkaç tür veri analizi tanıyoruz, ancak bu kılavuz içinde sadece üç temel ve en sık kullanılan türle ilgileneceğiz:

- **Tanımlayıcı ve Teşhis Edici Veri Analizi** - veri kümesi analizinin en basit (ve aynı zamanda en sık kullanılan) yöntemidir. Tanımlayıcı analiz, verilerden sonuçlar çıkarma (veya bilgi edinme) odaklanır. Genellikle veri kümesinin açıklaması ve veri kümesinin tanımladığı temel özelliklerin ölçümü bağlamında kullanılır - örneğin, organizasyondaki planların yerine getirilmesi. Teşhis Analizi, tanımlayıcı analizde tanımlanan olayların neden gerçekleştiğini açıklamayı amaçlar. Teşhis analizi genellikle veriler arasında bağlantılar kurduğu için kullanılır ve veri nesnelerinin davranışındaki tekrarlayan desenleri tanımlamak için kullanılabilir. Bu tür veri analizi, daha sonra benzer sorunları çözerken tekrar tekrar kullanılabilecek ayrıntılı bilgiler oluşturmaya dayanmaktadır.
- **Keşifsel Veri Analizi** - insanlar için en doğal veri analizi türü keşifsel veri analizidir. Bu, veri keşfi ile odaklanan ve genellikle veri görselleştirmesi kullanılarak gerçekleştirilen bir veri analizidir. Bu analiz, verilerdeki desenleri ve bağımlılıkları etkili bir şekilde tanımlar, ancak diğer analizlerin sonuçlarını sunma açısından da önemlidir. Keşifsel veri analizinin görsel yönünün yanı sıra, veri kümesini basitleştirmeye veya temsil etmeye yönelik eylemleri de burada dahil ediyoruz - örneğin, boyutsal azaltma, bir n-boyutlu veri kümesini m-boyutlu bir veri kümesine yansıttığımız bir işlemken m < n'dir.
- **Tahminsel Veri Analizi** - tahminsel analiz, yukarıda bahsedilen analiz türlerinin bir uzantısıdır. Amacı, toplanan verileri kullanarak olay sonuçlarının mantıklı tahminlerini oluşturmaktır veya aslında ölçmediğimiz değerleri tahmin etmektir. Bu tür veri analizinde, istatistiklere dayalı modelleme yöntemleri kullanılır ve bu da tahmin modelleri oluşturmak için hesaplama teknolojilerinin kullanılma ihtiyacını beraberinde getirir. Unutmayın ki tahmin analizi sırasında oluşturulan modellerin sonucunda elde edilen tahminler, yalnızca veri kümesi için tahmini değerlerdir ve doğrudan verilerin kalitesine bağlı olarak doğruluklarına bağlıdır.

Tüm bu veri analizi türleri, genellikle çözülmesi gereken iki temel sorunla çalışır - regresyon problemi ve sınıflandırma problemi. Bu bölümün geri kalanı, bu iki problemi açıklamak üzerine odaklanmıştır.

Regresyon sorunu

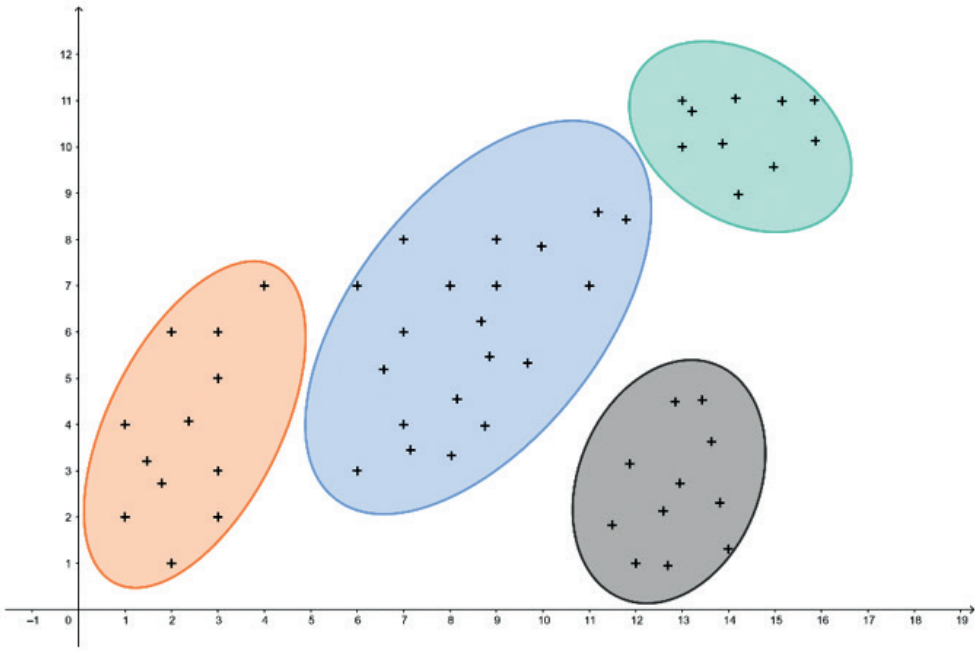


Yukarıdaki şekilde, iki özellik - bu değerler x ve y özelliklerinin değerleri olarak ölçülen beş noktayı içeren bir veri kümesini görebiliriz. Bu veri kümesi, [2, 3], [5, 4], [8, 2], [9, 8] ve [10, 6] noktalarından oluşur.

Bu durumda regresyon problemi, x değerini bilirsek ve önceki beş noktadan elde edilen deseni bilirsek, $y \in R$ özelliğinin gerçek değerini tahmin etme görevi olacaktır. Dolayısıyla, $x = 12$ özelliği için bir değer içeren ve hesaplamamız gereken bilinmeyen bir y özelliği içeren bir varlığımız var.

Genel olarak, bu tür bir problemi, $x, y \in R$ değişkenlerinin değerine dayalı olarak değişken y'nin sayısal değerini tahmin etmek veya öngörmek olarak tanımlayabiliriz.

Sınıflandırma sorunu



Bu tür bir problemi genel olarak şöyle tanımlayabiliriz: Verilen x deseni ve X uzayı ile, x deseninin hangi y değerini alacağını tahmin etmek için y ile ilişkilendirilmiş bir nitelik $y \in \{1,\ldots, n\}$ tahmin edilir.

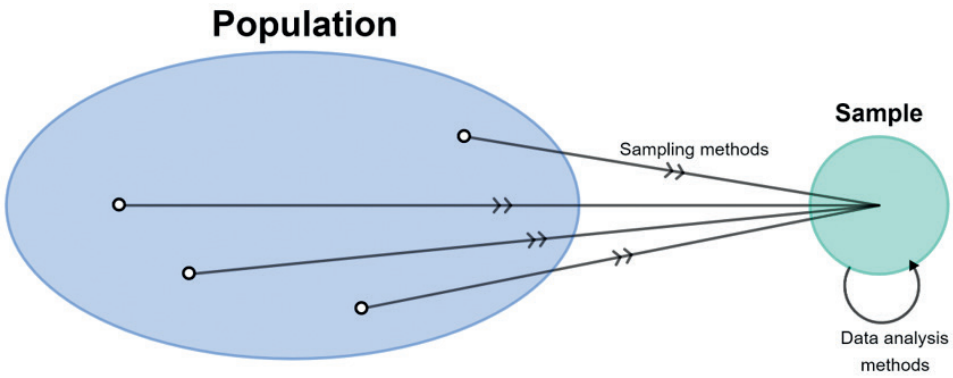
Bu açıklama biraz belirsizdir, daha anlaşılır olabilirdi ki sınıflandırma problemi, belirli bir anlamda benzer olan bireylerin önceden tanımlanmış sınıflara tahsis edildiği bir dizi sınıfa tahsis etmeyi içerir. Genel olarak, üç tür sınıflandırma prosedürü tanımlayabiliriz:

- **Hiyerarşik sınıflandırma**, sınıfların kendilerinin sınıflandığı bir sistemdir ve bu işlem farklı seviyelerde tekrarlanarak bir ağaç oluşturur.
- **Bölümleme**, sınıfların birbirleriyle çakışmayan ve böylece varlıkların bir bölünmesini oluşturan bir parçalandığı bir sistemdir.
- **Kümelenme**, sınıfların veya kümelerin üst üste gelebileceği ve bir kümenin ve onun tamamlayıcısının farklı sınıf türleri olarak ele alındığı bir sistemdir.

BÖLÜM 3

VERİ ÖRNEKLEME YÖNTEMLERİ

Bu bölüm, Matej Bel Üniversitesi Fen Bilimleri Fakültesi Bilgisayar Bilimleri Bölümü'nden Adam Dudáš tarafından yazılmıştır ve Slovakya'nın Banská Bystrica şehrinde bulunmaktadır.



Örneklem, bir nüfusun (veri kümesi) en temsilci parçasının seçilerek, nüfus hakkında bilgi edinmek ve analiz yapmak amacıyla kullanılabilmesi için seçildiği bir işlem olarak tanımlanabilir. Nüfustan örneklem alırken kullanılan teknik, örnekleme yöntemi olarak adlandırılır.

Bu nedenle örneklem, bir **nüfusun belirli bir bölümü** veya kesiri olarak tanımlanırken, nüfus, incelenen tüm konuların veya birimlerin toplamıdır.

Birçok bilinen örnekleme yöntemi bulunmaktadır. Genellikle bu yöntemleri olasılığa dayalı ve olasılığa dayalı olmayan örnekleme yöntemleri olarak **iki gruba** ayırırız. Ancak, hangi tipin kullanılacağı, çözülmesi gereken probleme tamamen bağlıdır. Genel olarak şunu söyleyebiliriz:

- **Olasılığa dayalı olmayan yöntemler**, örneklemi oluşturan kişiye bağlı olduğu için sonuçları elde etmek bir kişinin beklediği sonuçları almak oldukça kolaydır (bunlar tüm popülasyon için geçerli olmayabilir).
- **olasılığa dayalı yöntemler** bu sorunu daha az yaşama eğilimindedir.

3.1. OLASILIĞA DAYALI OLMAYAN ÖRNEKLEME YÖNTEMLERİ

Örneklemenin popülasyondan seçilmesi, örneklemi oluşturan kişinin değerlendirmesine büyük ölçüde bağlıdır. Bu nedenle, bu yöntemler bazı değerlerin popülasyona göre bozulmasına neden olabilir. Bazı olasılıksız örnekleme yöntemleri, sadece örneklemi oluşturan kişinin kolaylığına bağlıdır - örneğin, Kolaylık örnekleme olarak adlandırılan örnekleme yöntemi, popülasyonun üyeleri, örneklemi derleyen kişinin kolaylığına dayalı olarak seçilir. Benzer şekilde, Bir değerlendirme örnekleme adı verilen bir yöntemde, örneklem, derleyici tarafından yapılacak değerlendirmeye dayalı olarak oluşturulur - örneğin, örneklemi derleyen kişinin veriler dışı bilgisine dayalı olarak.

Birkaç olasılıksız örnekleme yöntemi açık ve sadece bir tür "sağduyu" örnekleme gibidir. Bu nedenle, bu kılavuzda yalnızca olasılıksız örnekleme yöntemlerinden yalnızca birinin daha ayrıntılı bir açıklamasını sunuyoruz.

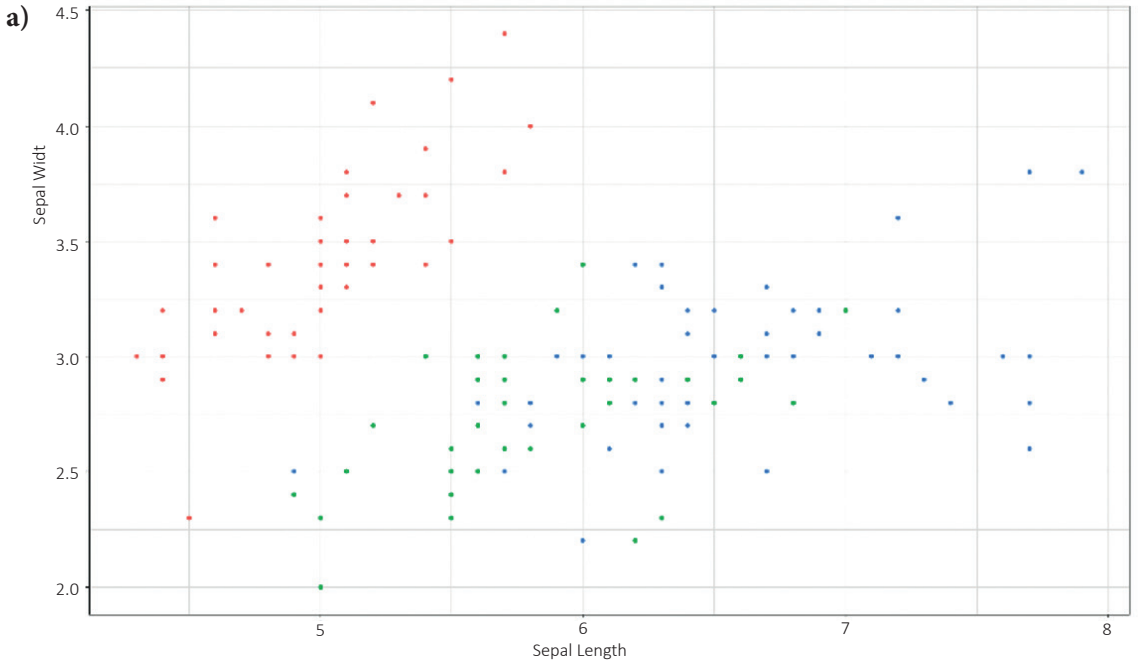
Amaçlı örnekleme yöntemi

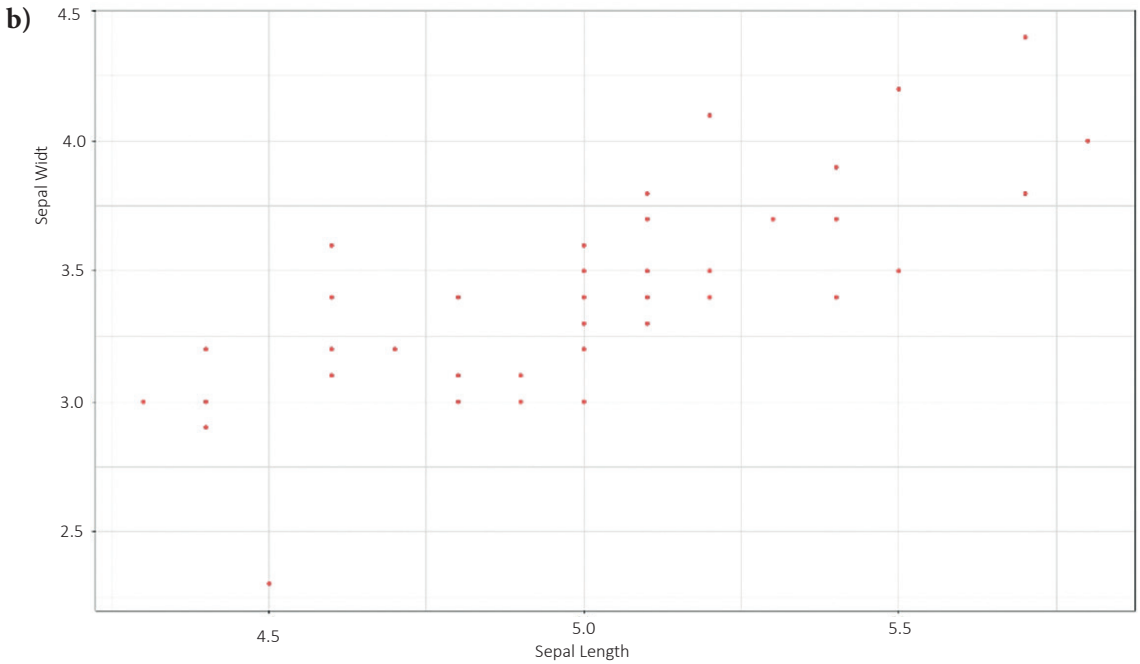
Bu örnekleme yönteminde, örnekleme yapan kişi, örnekleme çekilen özel bir amaç için popülasyon üyelerini seçer. Çünkü popülasyonun tüm üyelerinin örnekleme dahil olma aynı şansa sahip olmadığı bir örneklemeye olasılıksız bir örnekleme yönteminden bahsediyoruz.

Bu tür bir örnekleme örneği, bilgisayar bilimi alanında üçüncü sınıf öğrencilerinin analizini derlemek, yani tüm yılların tüm alanlarında öğrencilerin popülasyonundan bilgisayar bilimi öğrencilerinin üçüncü sınıf öğrencilerinin örneklemini oluşturmak gerektiğinde olabilir. Açıktır ki, birinci, ikinci, dördüncü veya beşinci sınıf öğrencilerini örnekleme dahil etmek istemeyiz. Aynı şekilde, uygulamalı matematik, biyoloji veya adli kimya gibi alanlarda okuyan öğrencileri örnekleme dahil etmeyeceğiz.

Bu bölümün bu noktasından itibaren bireysel örnekleme yöntemlerinin sonuçlarını tanımlamak için bu kılavuzun Ek A'sında açıklanan Iris veri kümesinden örnekler kullanacağız. Amaçlı örnekleme yöntemi için görev şu şekilde olabilir: *Belirli bir çiçek türü için sepal uzunluğu ve sepal genişliği değerlerini analiz etmek gereklidir - Iris Setosa.*

Şekil, a) tam iris veri kümesi (her iris çiçeği sınıfı kendi rengiyle işaretlenmiştir) ve b) yalnızca bir sınıf olan Iris setosa'dan oluşan bir örnekleme dahil olmak üzere sepal uzunluğu ve sepal genişliği değerlerinin karşılaştırmasını temsil eder.





3.2 OLASILIĞA DAYALI ÖRNEKLEME YÖNTEMLERİ

Bu başlıkla, düşünülen popülasyonun tüm üyelerinin örneklemin bir parçası olarak seçilme şansının eşit olduğu yöntemlere atıfta bulunuyoruz. Bu yöntemler, örnekleme yapılırken örnekleyicinin öznel hatalarını (veya azaltmayı) önler ve bu, olasılıksız yöntemler bölümünde bahsedildi. Farklı tipte olasılıksal örnekleme yöntemleri, farklı popülasyonlardan örnekler seçmek için farklı durumlarda kullanılır.

Bu yöntemler, araştırmacının düşünülen popülasyonu, uygun örnekleme yöntemini ve her karşılaşılan durumda nasıl kullanılacağını bilmesini gerektirir.

Birkaç olasılıksal örnekleme yöntemi vardır - çok aşamalı örnekleme, küme örnekleme, sistemli örnekleme vb. Biz, basit ve birçok çözülen soruna uygulanabilir dört olasılıksal örnekleme yöntemine odaklanacağız.

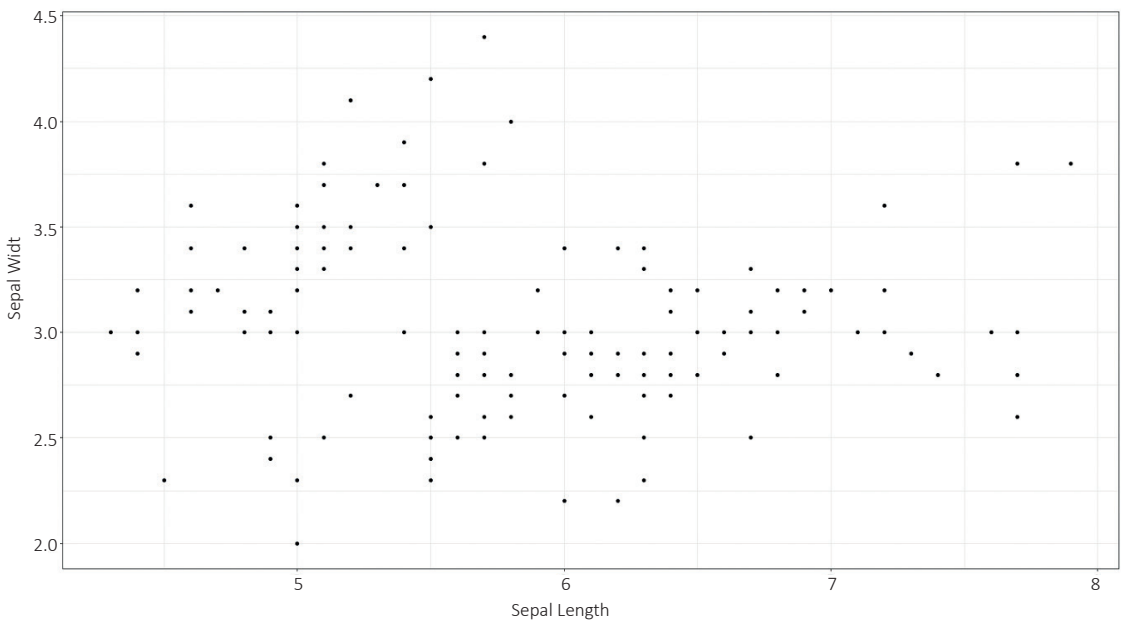
Basit rastgele örnekleme yöntemi

Bu yöntem, popülasyondan bireylerin rastgele seçilmesine dayanır. Başka bir deyişle, herhangi bir matematiksel model veya mantıklı karar verme olmadan, herhangi bir popülasyondan belirli bir örnek seçilir. Her bireyin (kaydın) örneğin bir parçası olma şansı aynı olduğundan, bu yöntem olasılıksal örnekleme yöntemlerinin en temsilcisidir.

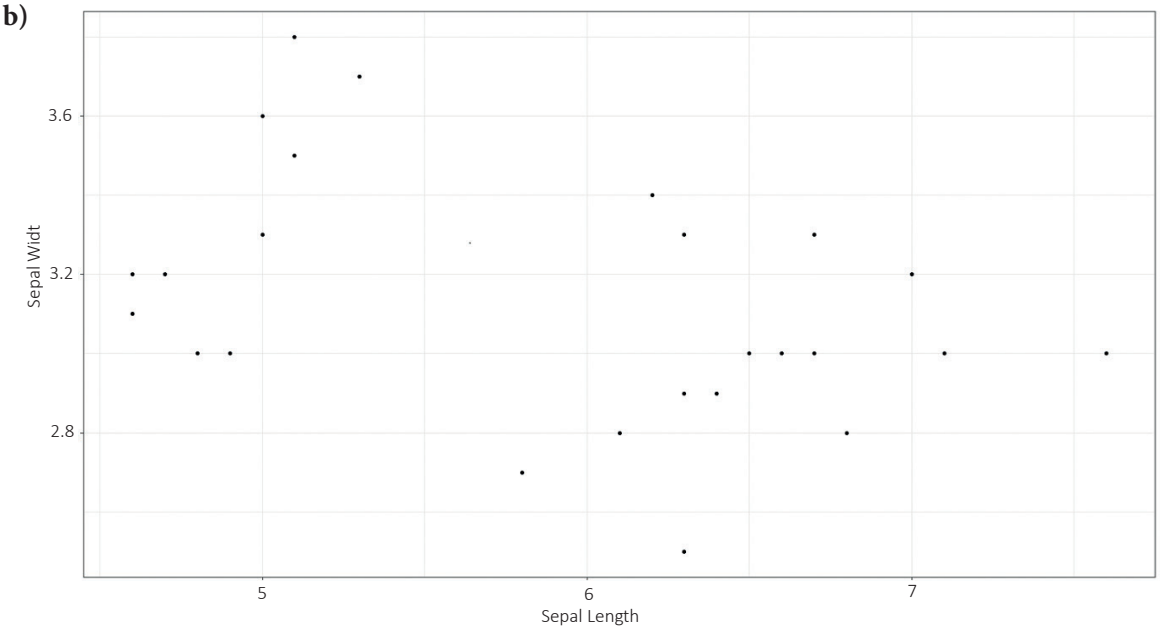
Basit rastgele bir örnekleme yönteminin yalnızca bir giriş parametresi vardır - örneklem için istenen boyut.

Örnek: Popülasyonumuz (alt şekil a), 150 birey (kayıt) içeriyor ve ondan rastgele 25 temsilci seçiyoruz (alt şekil b) - bu küme bizim için basit rastgele bir örnek temsil ediyor.

a)



b)



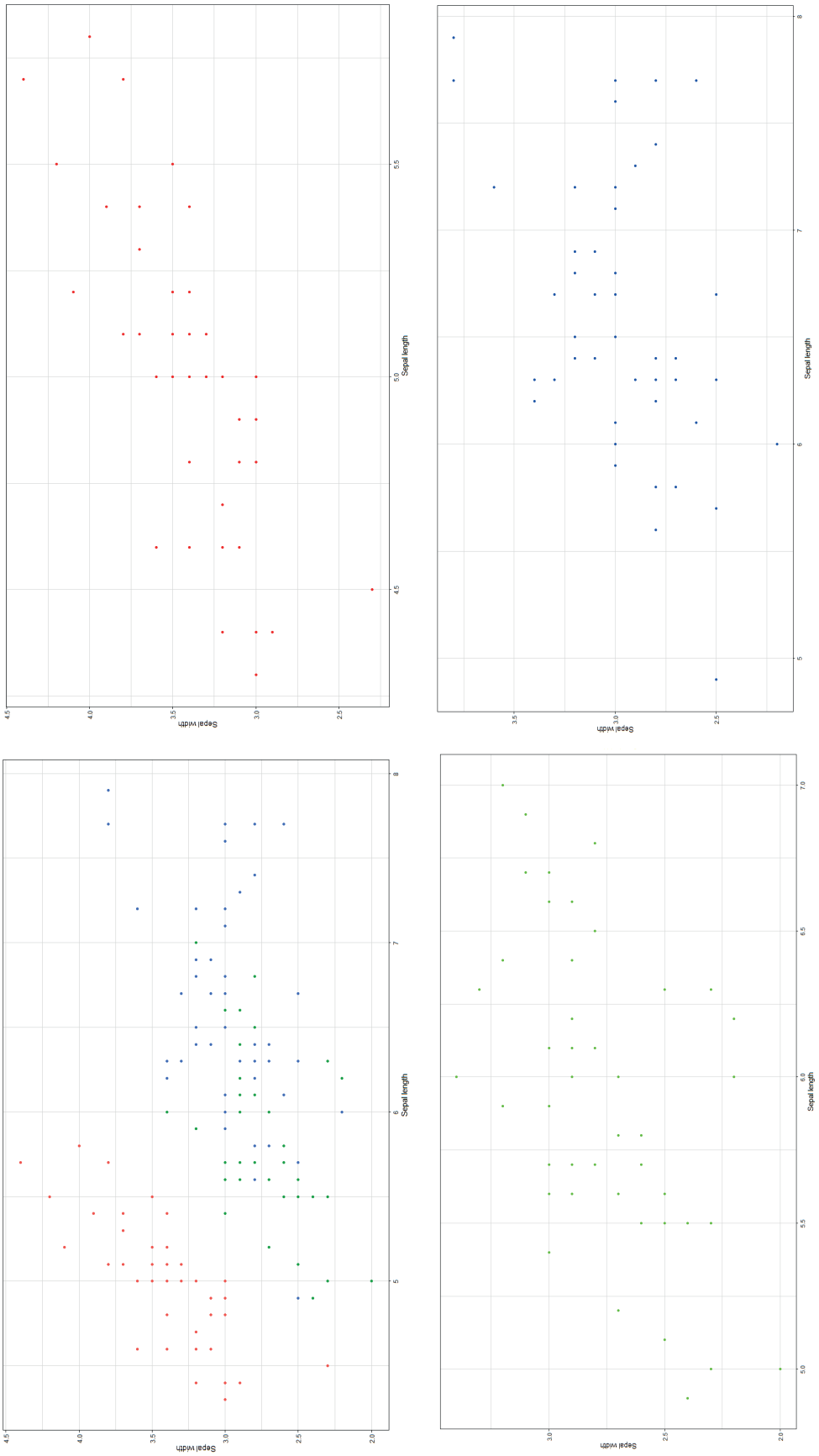
Küme örnekleme yöntemi

Küme örnekleme, bir veri kümesinin tamamının bölümlere veya kümelere ayrıldığı bir yöntemdir. Kümeler, belirli bir özniteliğe dayalı olarak tanımlanır ve örneğe dahil edilir. Bu öznitelik genellikle kategorik olup saç rengi, cinsiyet vb. gibi örneklerdir. Bu yöntem, veri içinde zaten var olan alt küme analizi için uygun örnekler oluşturmak için uygundur.

Küme örnekleme yönteminin yalnızca bir girdi parametresi vardır - veri kümesini kümelendirmek için kullanılacak bir öznitelikdir.

Örnek: Veri kümemizi (en üst, sol) sınıf özniteliğine göre üç değer alabilen - iris setosa, iris versicolor ve iris virginica olan bir şekilde böldük. Küme örnekleme yöntemini kullanarak, veri kümesindeki bireylerin özelliklerini analiz etmek için kullanılabilecek üç örnek oluşturabiliriz. Örneğin, sample2 (en üst, sağ) sadece kendi sınıf özniteliği değeri sample2 olan alt küme hakkında sonuçlar çıkarmak için uygun değildir. Bu yöntemin uygun bir kullanımı, iris çiçeği sınıfının özelliklerini karşılaştırmamıza olanak tanıyan bireysel kümeleri tanımlayan istatistiksel bir analiz hazırlamaktır.

Sistematik örnekleme yöntemi

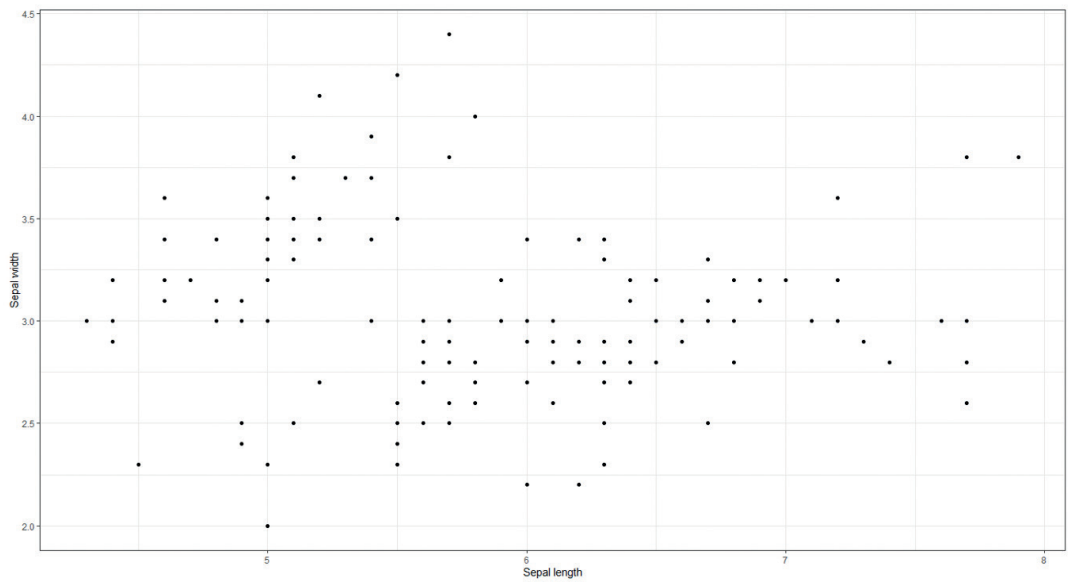

Sepal width
Sepal length
Sepal width
Sepal length
Sepal width
Sepal length
Sepal width
Sepal length

Bu yöntem, populasyondan örnek üyelerini düzenli aralıklarla seçmek için kullanılır. Bu örnekleme yöntemi önceden tanımlanmış bir kapsama sahiptir ve bu nedenle en az zaman alan örnekleme tekniğidir.

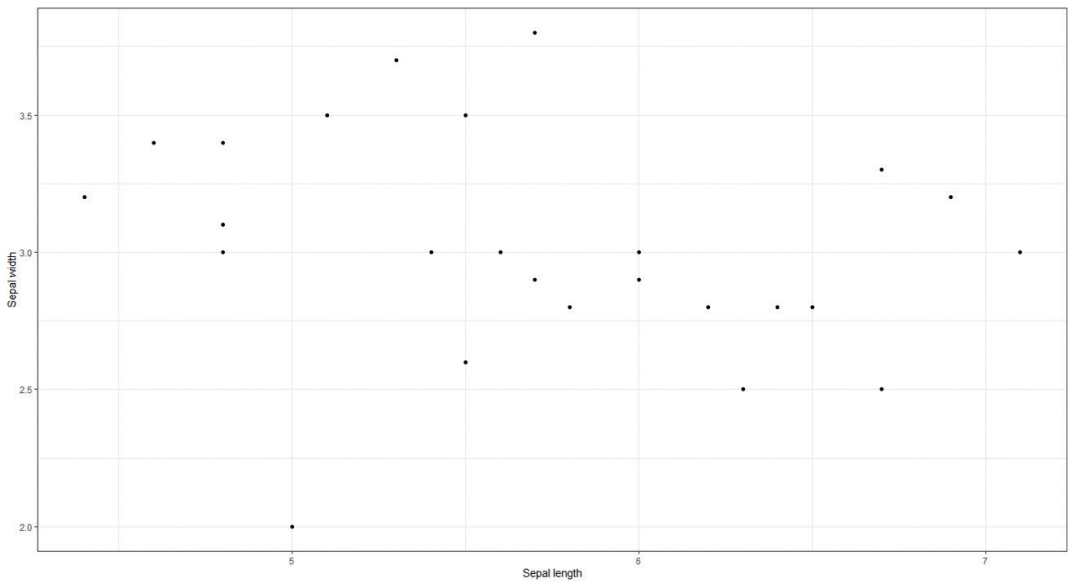
- Sistematik örnekleme yönteminin iki veya üç girdisi vardır:
- Örnek oluşturmak için seçilen başlangıç noktası (örneğe ait ilk birey),
- Bireylerin örneğe eklenmesiyle oluşturulan örtük örnek boyutunu oluşturan aralık,
- veya bireylerin örneğe eklenmesiyle ve oluşturduğumuz örnek boyutu aralığında.

Örnek: Basit rasgele örnekleme yöntemi ile 25 birey seçtiğimizden ve bu bireyler, kullandığımız populasyonu temsil ettiğinden, sistematik örnekleme yöntemini kullanarak da 25 bireyden oluşan bir örnek oluşturmak istiyoruz. Orijinal set 150 temsilciden oluşuyor ve 150/25 = 6 olduğu için her altıncı bireyi seçeceğiz (veri kümesindeki ilk kayıttan başlarsak). Aşağıdaki şekilde, tüm populasyonu (şekil a) ve yukarıda açıklanan prosedürle seçilen 25 bireyden oluşan örneği (şekil b) görebiliriz.

a)



b)



Tabakalı örnekleme yöntemi

Küme örnekleme yöntemi, veri kümesinin tamamını, toplam populasyonu temsil eden daha küçük ayrık gruplara böler. Küme örnekleme yöntemi ile karşılaştırıldığında, bu yöntem, verideki grupları orijinal veri kümesinde bulunan bir özellik üzerinde yeni tanımlanan bir sınır kullanarak oluşturur. Küme örnekleme yöntemi bu sınırları oluşturmaz, ancak verideki grupları belirlemek için bir (kategorik) özellik kullanır.

Örnek: Aşağıdaki şekilde stratified örnekleme yöntemi kullanılarak oluşturulan örnekler, Sepal uzunluğu özelliğinde tanımlanan aralıklar olarak tanımlanabilir. Her örnek farklıdır, ancak her örnekte seçilen yöntemin bakış açısından benzer sepal uzunluğu değerine sahip temsilciler bulunmaktadır. Bizim durumumuzda, örnekleri en küçükten en büyüğe 1 cm'lik aralıklarla böldük:

$$sepal\_length \in (4, 5]$$
$$sepal\_length \in (5, 6]$$
$$sepal\_length \in (6, 7]$$
$$sepal\_length \in (7, 8]$$

Bu nedenle, şekil kırmızı ile işaretlenen örnek1, yeşille işaretlenen örnek2 ve benzerleri şeklinde dört örnek içerir.

3.3 NUMUNE KALİTESİ KONUSUNDA BİRKAÇ KELİME

Örneklemenin doğru bir şekilde yapılması, büyük verilerle çalışırken gereken bir tekniktir (bkz. başvuru için bölüm 1.1). Yukarıda bahsedilen yöntemler, kalitesi birkaç açıdan değerlendirilebilen örnekler oluşturur. Bu kılavuz için, örnek kalitesini tanımlamak için sadece iki kriter sunuyoruz ve bunlardan yalnızca biri sıradan kullanıcılar için gerçekten kritik olarak kabul edilmelidir (bu terim, özellikle bilgisayar bilimi alanı dışındaki kullanıcılar için kullanılır):

- **Örnek toplama hızı** - günümüzde genellikle optimize edilmiş işlevler ve paketler içeren modern yazılımlar kullanmaktayız. Bu tür işlevler, seçilen araçta uygulanmış örnekleme yöntemlerini içerir ve bu yöntemlerin etkinliğini artıran optimizasyonlar ve yöntemlerden geçer (bu neredeyse garanti edilir). Standart büyük veri benzeri olmayan bir veri kümesinden örnek oluşturmak gerekiyorsa, kullanıcı, örnek oluşturmayı uzatan yetersiz sistem performansı sorunuyla karşılaşmaz (veya ölümcül durumlarda örnek oluşturamaz). Ancak, gerçekten büyük veri kümesiyle çalıştığımızda, standart sistem yeterince verimli olmaktan çıkar. Kendi deneyimlerimizden örnek vermek gerekirse, 100 milyon kayıt içeren, her bir kaydın on altı özellik içerdiği bir veri kümesinden (popülasyon) örnek oluşturma ihtiyacı olduğunu düşünelim (bu büyük bir veri kümesi değil). R dili üzerinde standart bir kullanıcı bilgisayarında giriş parametresi 4 olan (nüfusun %25'ini temsil eden bir örnek oluşturmak için) sistemli örnekleme yöntemi işlevini kullanarak, örnek oluşturamadık. Bu sorun, yüksek performanslı hesaplama ve bulut hesaplama yöntemlerinin kullanılması gibi birkaç şekilde çözülebilir.
- **Örneğin temsilciliği** - örnek kalitesini değerlendirmek için örnek toplama hızından daha önemli bir sorundur ve bu, oluşturulduğu popülasyonu tanımlama yeteneği ile ilgilidir. Yukarıda verilen küme örnekleme yöntemi açıklamasında belirtildiği gibi, bir kümenin (verinin belirli bir alt kümesi) örneği, tüm nüfus hakkında bir sonuç çıkarmak için uygun değildir. Örnek ve nüfusun özelliklerini karşılaştırmak uygun olduğunda, amaçlarımıza göre birkaç şekilde ilerleyebiliriz:
  - **Örneklemi istatistiksel olarak açıklama** - Verileri az sayıda değerle açıklamak istediğimiz durumlarda, kritik istatistiksel ölçümleri hesaplayabiliriz. Bu sayısal değerlerden, veri ile ilgili daha fazla çalışma için uygun bilgiler elde edebiliriz.
  - **Örneğin görselleştirilmesi** - büyük veriler, görselleştirmenin karmaşıklığı ile ünlüdür. Bu nedenle, daha az birey içeren ve bu nedenle daha kolay görselleştirilebilen temsilci bir örnek oluşturmak iyi bir fikirdir.
  - **Örneklemenin tahmin potansiyelinin analizi** - amacımız, makine öğrenimi temelli tahmin veya tahmin modelleri oluşturmaksa, karar ağacı modelleri veya korelasyon analizi gibi yöntemleri kullanarak bireysel özniteliklerin tahmin potansiyelini analiz etmek uygundur.

  Tüm bu yaklaşımlar, bu el kitabının 4. bölümünde daha detaylı olarak açıklanmıştır.

3.4 ÖRNEKLEM BÜYÜKLÜĞÜ KONUSUNDA BİRKAÇ KELİME

Verilen bir populasyondan bir örneklem derlememiz gerektiğinde, **istediğimiz sonuçları elde etmek için bu örneklem ne kadar büyük olmalı sorunu ortaya çıkabilir** - ihtiyacımız olan bilgileri doğru bir şekilde çıkarabilmek için. Bu sorunun cevabı kullanılan yönteme ve hedeflerimize bağlıdır:

- Küme örnekleme veya tabakalı örnekleme gibi örnekleme yöntemleri durumunda, **cevap kullanılan yöntem tarafından verilir**. Bu yöntemler, veride belirli bir değerin görünmesi tarafından tanımla-

nan bir örneklem oluşturur ve bu nedenle bu durumda kümenin boyutundan farklı bir örneklem boyutunu düşünmek standart değildir.

- Diğer durumlarda, özellikle rastgele örnekleme durumunda, **ihtiyaçlarımıza uygun örneklem boyutunu tanımlayan bir model kullanmak gereklidir**. Bu model, varsayılan olarak sınırlı sayıda birey içeren populasyonlar veya birey sayısı sınırsız populasyonlar için iki türde tanımlanmıştır. İhtiyaçlarımız için, bu sınırlı bir populasyonun daha doğal olanını düşüneceğiz:

$$\bar{n} = \frac{\frac{z^2 \bar{p}(1-\bar{p})}{\varepsilon^2}}{1 + \frac{z^2 \bar{p}(1-\bar{p})}{\varepsilon^2 N}}$$

- n, örneklem büyüklüğüdür,
- z, güven düzeyini yansıtan ve genellikle %90, %95 veya %99 olarak ayarlanan, aşağıdaki tabloda sunulan z-puanıdır:

Nivel de încredere	Scor-z
90%	1,65
95%	1,96
99%	2,58

Bu tablo, önceden hesaplanmış z-puanı değerlerini içeren ve diğer güven düzeyi değerleri için çevrimiçi olarak aranabilen tipik bir z-puanı tablosudur.

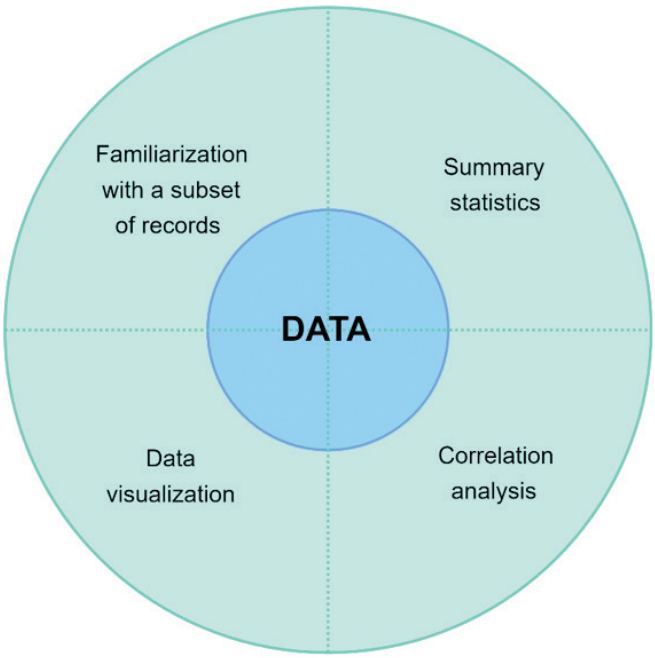
- n, nüfus oranıdır - nüfusla ilişkili nüfusun yüzdesi (veya kesri) araştırılan problem (bilinmeyen bir popülasyon için standart değer p = 0,5 olarak ayarlanmıştır).
- ε kullanıcı tarafından belirlenen hata payıdır.
- N, kullanılan popülasyonun büyüklüğüdür.

Örnek boyutunu hesaplamanın en kolay yolu, ücretsiz olarak sunulan çevrimiçi örnek boyutu hesaplayıcılarını kullanmaktır. Yukarıda sıralanan ilkelere göre çalışırlar.

BÖLÜM 4

KEŞFEDİCİ VERİ ANALİZİNİN TEMELLERİ

Bu bölüm el kitabının bir parçasıdır ve Adam Dudáš tarafından, Slovakya'nın Banská Bystrica şehrinde bulunan Matej Bel Üniversitesi'nin Fen Bilimleri Fakültesi, Bilgisayar Bilimleri Bölümü'nde yazılmıştır.



Bu el kitabının önceki bölümlerinde belirtildiği gibi, veri analizi sürecinde veri kümesinin tamamı veya bu metindeki 3. bölümde bahsedilen yöntemlerle oluşturulan bir örneklem ile çalışabiliriz. Temel, tanımlayıcı veri analizi için, veri kümesi veya örneklem verilerinin özelliklerini yakalamak için araçlar sunan tanımlayıcı istatistik yöntemlerini kullanırız. Tanımlayıcı istatistik yöntemleri, veri alt kümesini temsil etmek için birleştirme yöntemlerine dayanır, örneğin, özellik ortalama değeri, minimum, frekans veya değerlerin toplamı. Bu tür yöntemlere veri indirgeme yöntemleri olarak da adlandırılabilir.

Tanımlayıcı istatistik yöntemlerini kullanarak verileri analiz ederken, genellikle aşağıdaki üç kavramı kullanırız:

- **Merkezi eğilim ölçüleri**, verilerin etrafında toplandığı veya dağıldığı merkezleri aradığımız ölçülerdir.
- **Değişkenlik ölçüleri**, verilerin dağılımını açıklar, yani bireysel ölçümlerin merkezden ne kadar uzak olduğunu gösterir.
- **Korelasyon analizi**, veri kümesindeki bireysel özellikler arasındaki tahmin potansiyelini açıklayan katsayıların hesaplanmasına dayanır. Bu katsayılar, makine öğrenimi modellerinin oluşturulmasında önemlidir, aynı zamanda bu el kitabının bu bölümünün temel bir parçası olan veri görselleştirmesinde de önemlidir.

Bu el kitabının bu bölümünde, veri analizinin insanlar için oldukça doğal olan ilk versiyonu olan Keşifsel Veri Analizi'ne (EDA) odaklanacağız. Adından da anlaşılacağı gibi, verileri keşfetmek için veri analizi yaparız ve bir veri kümesinde veya örneklemede desenleri ve eğilimleri buluruz. Bu tür bir analizin en temel biçiminde, bu tür bir analiz için önemli bir parça olan veri görselleştirmeleri ile yapılmaktadır.

Hangi veri kümesi bölümlerinin görselleştirilmesinin uygun olduğunu ve hangilerinin olmadığını bilmeniz için tanımlayıcı istatistikler yöntemleri aracılığıyla edinilen temel bilgileri kullanırız.

4.1 TEMEL İSTATİSTİK YÖNTEMLERİ

Temel istatistik yöntemleri açısından, verilerdeki merkezi ölçen yöntemleri tanımlayabiliriz - verilerin etrafında toplandığı veya yoğun bir şekilde dağıldığı merkezleri ararız. Bu yöntemlerden en yaygın olanları şunlardır:

- **Ortalama**, dizi öğelerinin ortalama değeridir. Ortalama, simetrik dağılımlı verileri (örneğin, boy *n*, kilo) karakterize etmek için uygundur. Simetrik dağılımlı veriler, veri kümesindeki öğe sayısının ortalama sınırın altında ve üstünde benzer olması gereken ve ideal olarak aynı olan verilerdir. Ortalama değer hesaplama ilişkisi aşağıdaki gibidir:

$$\mu_A = \frac{\sum_{i=1}^{n} Ai}{n},$$

Burada μA, A özelliğinin ortalama değeridir, ise A özelliğini içeren varlıkların sayısıdır ve Ai, bu özelliğin i. değeridir.

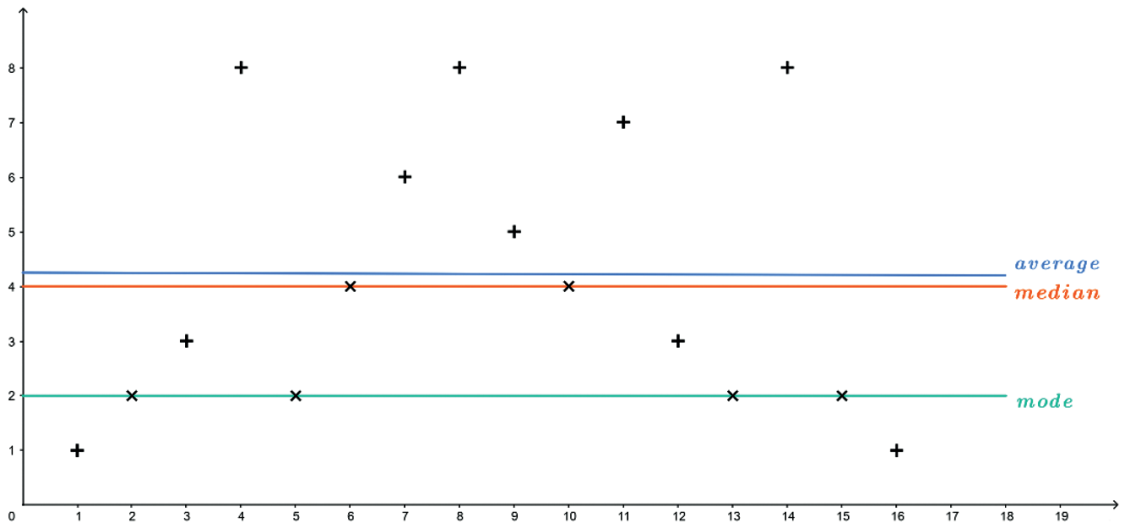
- **Orta değer**, sıralanmış bir dizinin orta değeridir. Medyan son derece simetriktir, yani medyanın altında ve üstünde aynı sayıda öğe bulunur. Bu kuralın istisnası, çift sayıda öğe içeren veri kümesidir (bu durumda medyanı, makul bir veri kümesinde birbirlerine oldukça yakın olmaları gereken iki orta değerden birini seçeriz). Ortalamanın aksine, medyan bir özelliğin gerçek değeridir, bu nedenle veri aykırı değerler içeriyorsa ve asimetrik olarak dağılmışsa (örneğin, belirli bir bölgedeki çalışanların maaşları), medyan daha uygundur. Medyan, aşağıdaki ilişki kullanılarak hesaplanır:

  n, öznitelik A içeren öğelerin sayısını temsil eder.

$$mediana_A = \frac{(n+1)}{2} \text{ sıralı bir kümenin bir elemanını temsil eder,}$$

- **Mod**, bir özniteliğin içerisinde en sık görülen değerdir. Ancak, modu kullanmak ve pek çok analitik görevde doğru sonuçlar elde etmek zor olabilir. Örneğin, gelirler için mod genellikle 0 olabilir, çünkü birçok insan (işsizler, çocuklar, emekliler vb.) gelir elde etmez.

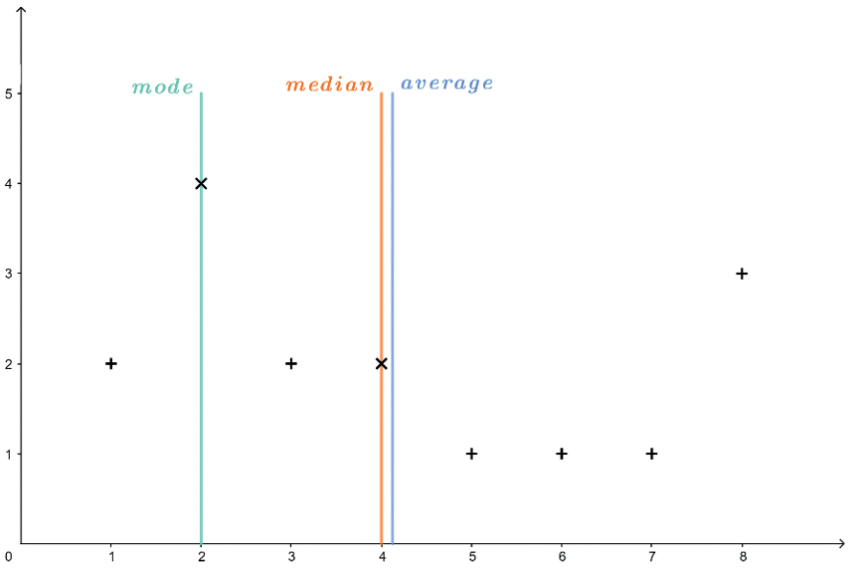
Aşağıdaki şekilde, basit bir veri kümesi için merkezi eğilim ölçülerinin görselleştirmesini sunuyoruz. Normal dağılıma sahip verilerde ortalama ve medyan değerlerinin birbirine yakın olduğu tipik bir davranışı gösterir. Ancak, mod değerini tahmin etmek zor olabilir çünkü bu, özniteliğin içerisinde en sık görülen öğe değerini temsil eder ve bu değer yüksek, düşük veya ortada bir yerde olabilir.



Bu standart ölçülere ek olarak, bir özniteliğin değerlerinin frekansı oldukça önemlidir. Frekans dağılımı başlığı altında, bir örnekte (veri kümesinde) çeşitli çıktıların sıklığını gösteren bir liste, tablo veya grafik anlamına gelir. Tablonun her girdisi, belirli bir grup veya aralıktaki değerlerin sıklığını (veya sayısını) içerir. Yukarıda kullanılan basit veri kümesi için böyle bir frekans dağılımı örneği aşağıdaki gibidir:

değer	sıklık
1	2
2	4
3	2
4	2
5	1
6	1
7	1
8	3

Bu tablo, aşağıdaki grafikle eşleştirilebilir. Frekans grafiklerinin böyle bir görselleştirilmesi, özellikle verileri tanıma ve aykırı değerlerin tespiti açısından önemlidir.



Düşünülen uzaydaki veri değişkenliğinin diğer yönü, standart istatistiksel ölçülerle birlikte gelir. En yaygın değişkenlik ölçüsü, toplam kareler farklarının ve ortalama değerin bir araya getirilmesiyle tanımlanan standart sapmadır (σ):

$$\sigma_A = \frac{\sqrt{\sum_{i=1}^{n}(A_i - \mu_A)^2}}{n-1},$$

μ_A, özellik A'nın ortalama değeridir, n, özelliği A içeren varlık sayısıdır ve A_i, bu özelliğin i. değeridir. Standart sapmaya benzer şekilde hesaplanan varyans:

$$V = \sigma^2$$

Örnek: Aşağıdaki beş ölçümden oluşan basit bir veri kümesimiz olsun: A = [20, 60, 40, 70, 50]. Bu veri kümesi için ortalama, medyan ve standart sapmayı hesaplayalım.

$$\mu_A = \frac{\sum_{i=1}^{n} A_i}{n} = \frac{20+60+40+70+50}{5} = \frac{240}{5} = 48$$

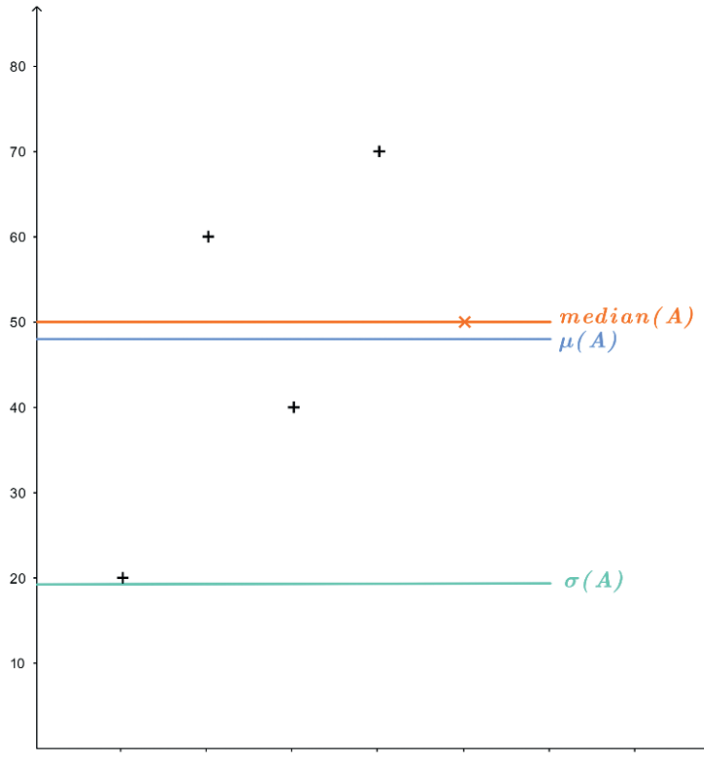
$$mediana_A = \frac{(n+1)}{2} = \frac{6}{2} = 3rd$$ *sıralanmış kümenin elemanı* → [20, 40, 50, 60, 70] = 50

$$\sigma_A = \frac{\sqrt{\sum_{i=1}^{n} (A_i - \mu_A)^2}}{n-1}$$

$$= \frac{\sqrt{(20-48)^2 + (60-48)^2 + (40-48)^2 + (70-48)^2 + (50-48)^2}}{4} = \frac{\sqrt{1480}}{4}$$

$$\sqrt{370} \approx 19.235$$

Bu standart sapma nispeten yüksektir, ki bu doğal bir durumdur çünkü veri kümesi kendisi oldukça dağılmıştır. Bu ölçümlerin görselleştirmesi aşağıdaki şekildedir.



Merkezcilik ve değişkenlik hesaplama yöntemleri, veri kümesini az sayıda değer kullanarak tanımlamak için kullanılır. Bu tür bir veri kümesi tanımı, **özetleme yoluyla veri kümesi** tanımı olarak da adlandırılabilir.

Örneğin: Aşağıdaki öz niteliğe sahip veri kümesine sahibiz A = [1, 2, 3, 8, 2, 4, 6, 8, 5, 4, 7, 3, 2, 8, 2, 1] (4.1bölümünün başında sunulmuştur). Bu veri kümesini, üç toplam değer kullanarak tanımlayabiliriz, örneğin (min(A), μ(A), max(A)), bu nedenle A = (1, 4.125, 8).

Son gösterge, **veri kümesinin verilerinin uzaya dağılımıdır**. Bu metriği karakterize etmek için özniteliğin ortalama değeri ve standart sapma kullanılır. Normal olarak dağılmış bir veri kümesinde, en azından $(1 - 1/k^2)$-si noktaların ortalama değerden kσ veya daha az bir uzaklıkta bulunur. Bu tür bir veri kümesi aykırı değer içermez ve makine öğrenme ve yapay zeka veri analizi yöntemleri için mükemmel bir adaydır.

Örnek: Tanıdık öznitelik A = [20, 60, 40, 70, 50] için dağılımı şu şekilde hesaplayabiliriz:

$\mu A = 48$

$\sigma A \approx 19.235$

$2\sigma = 2 * 19.235 \approx 38.47$

Görüyoruz ki en az 3 değer ortalama değerden (48) en fazla 38.47 birim uzakta olmalıdır. Bu, öznitelik A'nın tüm ölçümleri için geçerlidir.

4.2 KORELASYON ANALİZİ

Önceki bölümde tanımlanan temel istatistiksel değerler, veriyi açıklamak için önemli göstergelerdir. Bununla birlikte, veri analizi hedeflerine göre, söz konusu korelasyon analizi olarak adlandırılan daha güçlü bir ölçüt vardır.

Veri kümemiz birden fazla sayısal öznitelik içeriyorsa, bu veri kümesinin iki elemanlı alt kümeleri arasındaki korelasyonu ölçebiliriz. Diyelim ki veri kümemizin A_1 ve A_2 adlı iki özniteliği var. Bu öznitelikler, A_1'in A_2 özniteliği için tahmin edici potansiyele sahip olduğunda birbiriyle korele olurlar. Bu tür tahmin edici potansiyel, **veri kümesindeki eğilimlerin ve desenlerin varlığını** ve veri ile çalışan analitik modellerin oluşturulma olasılığını gösterir.

İki değişkenin korelasyonunu, iki değişken arasındaki **korelasyon katsayısı** $r(A_1, A_2)$ ile ölçeriz ve bu katsayı, A_1 özniteliğinin A_2 özniteliği için bir işlev olup olmadığını ve tersinin de geçerli olduğunu gösterir. Bu korelasyon katsayısı, *[-1, 1]* aralığında değer alabilir ve:

- **1**, iki özniteliğin **tam korelasyonunu** gösterir. Başka bir deyişle, bir özniteliğin değeri arttığında, diğer özniteliğin değeri de artar. İki değişken arasında tam bir korelasyon olduğunda, güçlü bir tahmin potansiyelinden bahsediyoruz ve bu nedenle bu öznitelikler birbirlerini tahmin etmek için uygundur.
- **0**, iki değer arasından korelasyon açısından en kötü durumu gösterir, bu durumu **korelasyonsuzluk** olarak adlandırırız. İki özniteliğin korelasyon katsayısı birbirine yakın veya sıfıra eşit olduğunda, bunlar bağımsız değerlerdir ve analitik modeller oluşturma açısından kullanışlı değildir.
- **-1**, tam korelasyonun zıttıdır ve bu **durumu zıt korelasyon** olarak adlandırırız. Bu durumda, bir özniteliğin değeri arttıkça, diğer özniteliğin değeri azalır veya tam tersi bir eğilim belirleyebiliriz. Tam korelasyon durumunda olduğu gibi, bu, analitik modeller oluşturmak için uygun bir koşuldur.

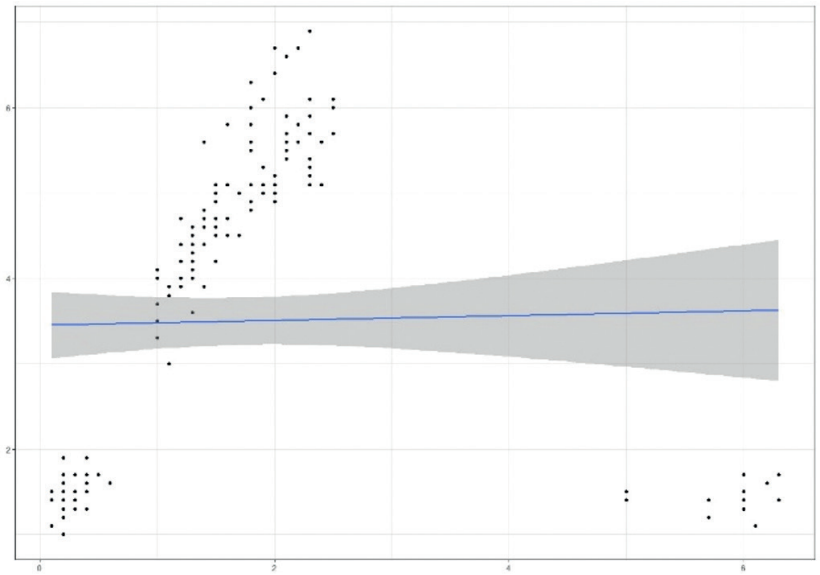
İlişkileri analiz etmek ve korelasyon katsayılarını ölçmek için kullanılan iki yaygın yöntem vardır - tabii ki daha birçok böyle yöntem bulunmaktadır. Ama amacımız için Pearson korelasyon katsayısı ve Spearman sıralama korelasyon katsayısı üzerinde odaklanacağız.

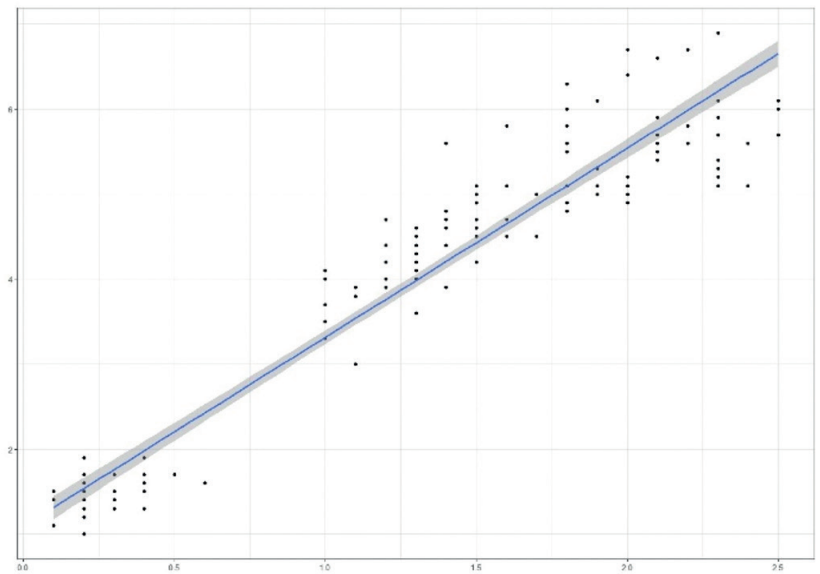
Pearson korelasyon katsayısı

İki veri kümesi özniteliği arasındaki ilişkiyi ölçmek için kullanılan ilk ve en güçlü katsayı, Pearson korelasyon katsayısıdır. Bu katsayı, **değerlerin lineer tahminine** ve A ve B öznitelikleri arasındaki ilişkiye odaklanır ve aşağıdaki ilişkiyle açıklanır:

$$r = \frac{\sum_{i=1}^{n}(A_i - \mu(A))(B_i - \mu(B))}{\sqrt{\sum_{i=1}^{n}(A_i - \mu(A))^2}\sqrt{\sum_{i=0}^{n}(B_i - \mu(B))^2}},$$

Burada $\mu(A)$, *A* özniteliğinin ortalamasıdır, benzer şekilde $\mu(B)$ *B* özniteliğinin ortalama değeridir ve n ölçüm sayısıdır (veri kümesinin dikey boyutu). Pearson korelasyon katsayısının en büyük dezavantajı olan bu ortalama değere olan hassasiyeti getiren açık bir bağımlılıktır.





Pearson korelasyon katsayısı, verilen özniteliklerin değerlerini tanımlayan bir çizgi arıyoruz. Sol taraftaki görüntüde, altta ve sağda aykırı değerlerin bulunduğu bir veri kümesiyle iki öznelliğin değerlerini karşılaştıran bir görselleştirmeyi görebiliriz. Ayrıca, verileri uydurduğumuz çizginin onu tamamen ve önemli ölçüde kaçırdığını görebiliriz (tek bir veri noktası hariç) - bu nedenle bu veri kümesinde tahmin edici potansiyeli ölçmek için Pearson korelasyon katsayısının uygun olmadığını sonuçlayabiliriz. Sağ tarafta, aykırı değerleri kaldırdıktan sonra veri kümesinin aynı özniteliklerini sunuyoruz. Bu durumda, çizginin verilerde mevcut olan eğilimleri tanımladığını görebiliriz.

Bu nedenle, **Pearson korelasyon katsayısı,** öznitelikler *A ve B*'nin şunları içerdiği durumlar için kullanılabilir:

- Doğrusal ilişkiler,
- Normal (Gaussian) dağılım,
- Aykırı değerlerin olmaması.

Spearman sıra korelasyon katsayısı

Nonlinear ilişkilere sahip ve aykırı değerler içeren veri kümesiyle başa çıkmak için farklı bir korelasyon katsayısı türü kullanabiliriz - özellikle, Spearman sıralama korelasyon katsayısı. Bu öznitelikler arasındaki korelasyonu ölçmenin bu yöntemi, öznitelik değerlerinin **sıralamasını** oluşturur (sıralama).

Örnek: şöyle bir özelliğimiz olsun A = [a0 = 4, a1 = 8, a2 = 2, a3 = 6]. Yukarıda bahsedilen hiyeraşi ve sıralama şöyle görülebilir.

a1 > a3 > a0 > a2 olduğundan sıra(a1) = 1, sıra(a2) = 4, gibidir.

Bu şekilde, özelliğin içindeki **değerlerin monotonluğunu** ölçeriz. Dolayısıyla, Spearman sıra korelasyon katsayısı, özellikler arasında monoton ilişkilerin olduğu veri kümeleri için en uygun olanıdır - bir özellik değeri arttığında diğer özellik asla azalmaz veya tam tersi olur. Öte yandan, bu tür bir korelasyon katsayısının, veri kümesinde tekrarlanan değerlerin olduğu durumlarda kullanılması önerilmez (aynı sıraya sahip anlamına gelir). Bu etki, veri kümesi büyüdükçe azalır.

Spearman sıra korelasyon katsayısı şu şekilde hesaplanır:

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)},$$

Burada d = rank(ai) – rank(bi) ve n, düşünülen özelliklerdeki varlıkların sayısıdır.

Örnek: Uyarı - aşağıdaki örnek öğrenciler arasında popülerdir; öğrenciler, belirli bir konseptin kendisinden daha çok bu örneği hatırlayabilirler. İki özellik düşünelim - dönerin fiyatı ve döner yerinin üniversiteden uzaklığı.

dizin	*Metre cinsinden mesafe*	*Euro cinsinden fiyat*
1	*10*	*4*
2	*70*	*3,50*
3	*85*	*3,30*
4	*100*	*3,20*
5	*130*	*3,80*
6	*195*	*2,90*
7	*215*	*3,10*
8	*300*	*3,90*
9	*420*	*3,15*
10	*505*	*3*

İlk olarak, Spearman sıra korelasyon katsayısının değerini hesaplarız. Bu yöntem, her iki özellik için de bir sıralama oluşturmayı ve d ve d2 değerlerini hesaplamayı gerektirir.

dizin	Metre cinsinden mesafe	*sıra (uzaklık)*	Euro cinsinden fiyat	Rand (fiyat)	*d*	d^2
1	*10*	*10*	*4*	*1*	*9*	*81*
2	*70*	*9*	*3,50*	*4*	*5*	*25*
3	*85*	*8*	*3,30*	*5*	*3*	*9*
4	*100*	*7*	*3,20*	*6*	*1*	*1*
5	*130*	*6*	*3,80*	*3*	*3*	*9*
6	*195*	*5*	*2,90*	*10*	*-5*	*25*
7	*215*	*4*	*3,10*	*8*	*-4*	*16*
8	*300*	*3*	*3,90*	*2*	*1*	*1*
9	*420*	*2*	*3,15*	*7*	*-5*	*25*
10	*505*	*1*	*3*	*9*	*-8*	*64*

Tablodaki değerler, Spearman sıra korelasyon katsayısını hesaplamak için kullanılabilir.

$$\sum d^2 = 256$$

$$n = 10$$

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} = 1 - \frac{6 * 256}{10(100-1)} = 1 - \frac{1536}{990} = 1 - 1.55 = -0.55$$

Bu nedenle, bu iki özellik arasında -0.55 olarak adlandırılabilecek ılımlı güçlü bir zıt-korelasyon olduğunu bulduk.

Pearson korelasyon katsayısının değerini de hesaplayalım. Bu tür bir katsayıyı hesaplamak için, ortalama mesafe değerini μ(distance) ve ortalama kebap fiyatı değerini μ(price) belirlememiz gerekmektedir:

- *μ(distance) = 203, bundan böyle μ(d) olarak adlandırılacaktır.*
- *μ(price) = 3.39, bundan böyle μ(p) olarak adlandırılacaktır*

Tablomuz daha fazla sütun içerecektir, ancak tüm bunlar, son Pearson korelasyon katsayısı ilişkisinin gerektirdiği parçaların önceden hesaplamalarıdır.

dizin	d	p	d – μ(d)	p – μ(p)	$(d - \mu(d))^2$	$(p - \mu(p))^2$	(d – μ(d)) (p – μ(p))
1	*10*	*4*	-193	0.61	37 249	0,3721	-117,73
2	*70*	*3,50*	-133	0,11	17 689	0,0121	-14,63
3	*85*	*3,30*	-118	-0,09	13 924	0,0081	10,62
4	*100*	*3,20*	-103	-0,19	10 609	0,0361	19,57
5	*130*	*3,80*	-73	0,41	5 329	0,1681	-29,93
6	*195*	*2,90*	-8	-0,49	64	0,2401	3,92
7	*215*	*3,10*	12	-0,29	144	0,0841	-3,48
8	*300*	*3,90*	97	0,51	9 409	0,2601	49,47
9	*420*	*3,15*	217	-0,24	47 089	0,0576	-52,08
10	*505*	*3*	302	-0,39	91 204	0,1521	-117,78

Bu nedenle Pearson korelasyon katsayısını aşağıdaki gibi hesaplayabiliriz:

$$\sum((d - \mu(d))^2) = 232\,710$$

$$\sum((p - \mu(p))^2) = 1.3905$$

$$\sum((d - \mu(d))\,(p - \mu(p))) = -252.05$$

$$r = \frac{\sum_{i=1}^{n}(A_i - \mu(A))(B_i - \mu(B))}{\sqrt{\sum_{i=1}^{n}(A_i - \mu(A))^2}\sqrt{\sum_{i=0}^{n}(B_i - \mu(B))^2}} = \frac{-252.05}{\sqrt{232\,710}\sqrt{1.3905}} \approx \frac{-252.05}{569.232} \approx -0.44$$

Bu nedenle, bu iki özellik arasında -0.44 Pearson korelasyonu olduğunu bulduk, bu da orta derecede güçlü bir karşıtlık olarak adlandırılabilir. Korelasyon katsayısı sonuçlarının yorumu hakkında daha fazla bilgi için bu elkitabının son bölümüne bakınız.

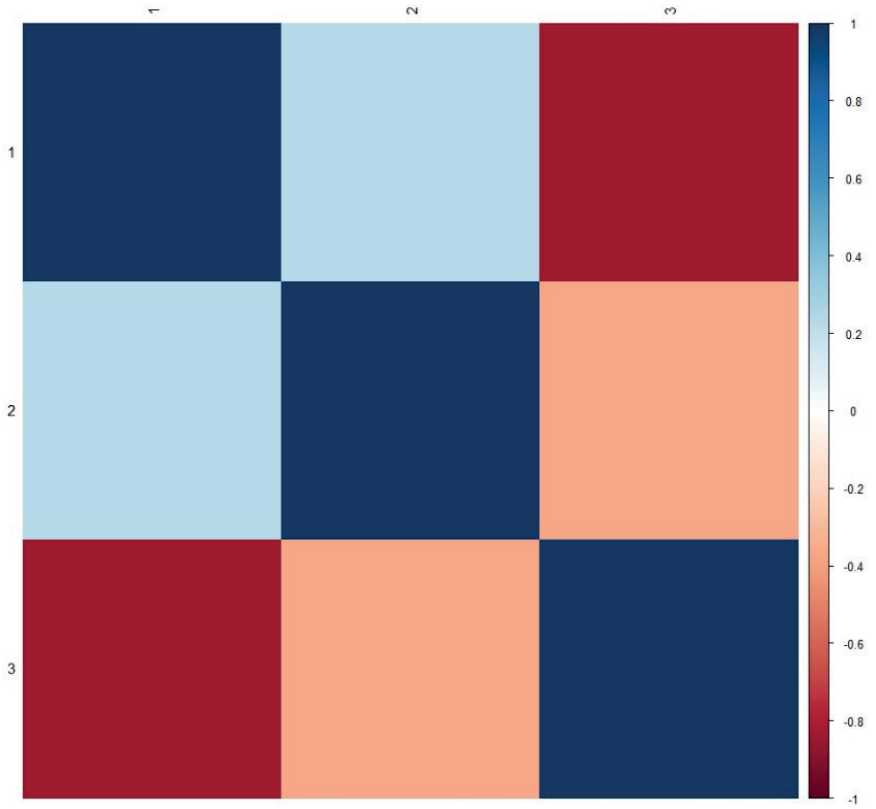
Korelasyon matrisi ve korelasyon ısı haritası

Veri kümesi genellikle yalnızca iki özellik içermez, bu da tüm özellikler arasındaki korelasyon katsayılarını ölçme ihtiyacını doğurur. Bu amaçlar için bir korelasyon matrisi kullanırız - veri kümesindeki tüm olası özellik çiftleri arasında ölçülen bir korelasyon katsayısını içeren bir tablo. Aşağıdaki tabloda, *A1, A2, A3* özellikleri arasında ölçülen korelasyon katsayısını görmekteyiz (korelasyon matrisi), bu sırada *r(A1, A2) = 0.238, r(A1, A3) = −0.834* ve benzeri değerleri görebiliriz.

	A_1	A_2	A_3
A_1	1	0,238	-0,834
A_2	0,238	1	-0, 362
A_3	-0,834	-0,362	1

Bu matrisin **iki doğal özelliği** vardır - çapraz simetriktir ve çaprazın her zaman kendisiyle olan korelasyon katsayısı değeri 1'dir - yani A_1 özniteliğinin kendisiyle olan korelasyonu her zaman $r(A_1, A1) = 1$'dir, kullanılan yönteme bakılmaksızın, ki bu da doğaldır çünkü A_1 özniteliği A_1 özniteliği değerine tamamen bağımlıdır.

Bu tür bir korelasyon analizi yöntemi, incelenen veri kümesindeki belirli sayıda özellik için uygundur. Açıkça, onlarca özelliği içeren bir veri kümesi için böyle bir matris kafa karıştırıcı ve okuması zor olurdu. Bu nedenle, genellikle bir ısıl harita veya korelasyon grafiği olarak adlandırılan bir görüntüleme yöntemi tarafından değiştirilir. Yukarıda sunulan korelasyon matrisi için ısıl harita şu şekilde oluşturulabilir:

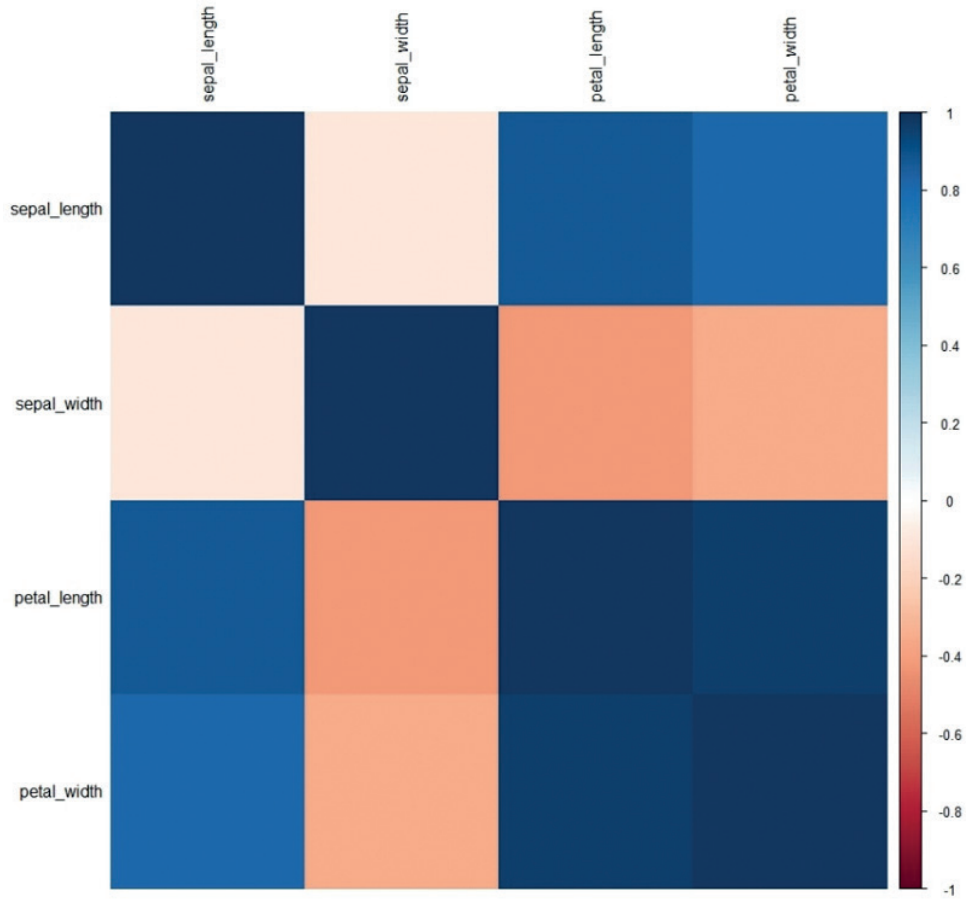


Bir korelasyon ısıl haritası, korelasyon matrisini sadece bir renkli ızgara içine yansıtan basit bir projedir. Burada, bir alanın rengi, verilen iki özellik çifti için korelasyon katsayısının değeri tarafından tanımlanır. Daha iyi okunabilirlik için, korelasyon ısıl haritasında mümkün olan korelasyon katsayısı değerleri aralığını içeren bir ölçek (sağda) belirtilir. Korelasyon matrisinde [1, -1] aralığının uçlarına yakın sayıları aramak yerine, korelasyon ısıl haritasında, aynı özelliği gösteren ve çoğu insan için daha kolay tanımlanabilen koyu kırmızı veya koyu mavi ızgara alanlarını ararız.

Örnek: Bu el kitabının Ek A'sında açıklanan Iris veri kümesine sahip olalım. Bu veri kümesi, 150 varlık üzerinde ölçülen beş özelliği içerir, bunlardan dörtü sayısal özelliklerdir ve beşincisi verilen varlık için sınıfı belirten bir dil özelliği içerir. Korelasyon analizinin bir parçası olarak, dil özellikleriyle çalışmak mümkün değildir, bu nedenle yalnızca 150x4 boyutundaki veri kümesini ele alacağız. (Biraz kırpılmış) Iris veri kümesinin Pearson korelasyon matrisi aşağıdaki gibi değerleri içerir:

	Sepal uzunluğu	Sepal genişliği	Sepal uzunluğu	Sepal genişliği
Sepal uzunluğu	1	-0.1093692	0,8717542	0,8179536
Sepal genişliği	-0,1093692	1	-0,4205161	-0,3565441
Sepal uzunluğu	0,8717542	-0,4205161	1	0,9627571
Sepal genişliği	0,8179536	-0,3565441	0,9627571	1

Tabii ki, bu veri kümesi çok sayıda özelliği içermediği için bir korelasyon ısıl haritası oluşturmak, korelasyon analizi işlemi için gerekli değildir. Ancak buna rağmen, korelasyon katsayısı değerlerini korelasyon ısıl haritasına yansıtma işlemini göstermek amacıyla bu ısıl haritayı sunuyoruz.



Korelasyon katsayılarının sonuçlarının yorumlanması

Korelasyon katsayısı, seçilen veri örneği içindeki A_2 özelliği değerlerini A_1 özelliğine dayalı olarak ne kadar iyi tahmin edebileceğimizi gösterir. Bu katsayının değeri, bu katsayının değeri 1 veya -1 değerlerine ne kadar yakınsa, verilen A_1 özelliğinin isteğe bağlı A_2 özelliği değerlerini tahmin etmek için ne kadar uygun olduğunu gösterir.

Açıkçası, 0 değeri, iki özellik arasındaki ilişkinin olmadığı ve üzerinde çalıştığımız veri kümesi için matematiksel modeller oluşturulmasında kullanılamayan **en kötü durumdur.**

Literatür, korelasyon değerlerinin kabul edilebilir düzeyi konusunda biraz farklı görüşlere sahiptir - yüksek korelasyonun daha iyi olduğu genel bir kabul olsa da. Genel olarak, iki özellik arasında güçlü bir korelasyon olduğundan bahsediyoruz, korelasyon katsayısı aralarındaki değerler **0.8'den yüksek** değerlere ulaştığında. İki özellik arasında güçlü bir karşıt korelasyon bulunurken, korelasyon katsayısı **-0.8'den düşük** bir değere ulaştığında mevcuttur. Tahmin potansiyelinin kabul edilebilir sınırları, **0.7 veya -0.7 değerlerine** yaklaşıldıkça gevşetilebilir, ancak daha fazlası önerilmez.

4.3 KEŞİF AMAÇLI VERİ ANALİZİ VE VERİ GÖRSELLEŞTİRME

Exploratory Data Annalysis (EDA), genellikle EDA olarak anılan, verileri keşfetmek için kullanılan bir veri analizi yöntemidir. Bir verilen populasyon veya örnek içindeki desenleri ve eğilimleri bulmak için kullanılır. Temel olarak, bu tür bir analiz, verilerin görsel keşfi ile gerçekleştirilir. Görselleştirme işleminden önce, verilerde gizli olan bilgileri aramada faydalı olacak birkaç adım atmak gereklidir:

- **Veri kümesine aşinalık kazanma** - verilerle ilgili bazı soruları yanıtlama yeteneği, verileri analiz etmeden önce gerekir:
  - **Veri kümesini kim, ne zaman ve neden derledi?** Bu nokta, verilerin geçerliliği, güncelliği ve kullanışlılığı açısından önemlidir. Veri kümesi, uzmanlar tarafından derlendi ise, başka birine göre daha geçerli olabilir, eğer verileri ucuz bir tüketici sınıfı sensörü ile ölçen bir acemi tarafından derlendi ise. Eğer veri kümesi 93 yıl önce derlendi ise, verilerin güncel olmayabileceği, ölçümlerin bugün yapabileceğimizden daha az doğru olabileceği vb. olasılıkları göz önünde bulundurulmalıdır. Veri kümesi ayrıca belirli bir amaçla oluşturulur ve bu nedenle evrensel değildir (her görev için uygun olmayabilir).
  - **Bu veri kümesi ne kadar büyük?** Veri kümesinin büyüklüğü ile, veri kümesinde ölçülen varlıkların ve özniteliklerin sayısını anlarız. Çalışmak için çok büyük bir veri kümesi olduğunda (bkz. Bölüm 1), bu el kitabının önceki bölümünde belirtilen prensiplere dayanarak bir örnek seçmemiz gereklidir. Ters problem - çok küçük bir veri kümesi - daha büyüktür. Bununla birlikte, bazı algoritmalar küçük veri kümeleriyle çalışabilir ve mevcut verilere dayalı olarak yeni varlıklar oluşturan ve aşırı örnekleme olarak adlandırılan yaklaşımlar vardır.
  - **Veri kümesinin bileşimi nedir?** Bu nokta, veri kümesinin derlenme nedeni ile yakından ilişkilidir. Tüm veri kümesi özniteliklerini gözden geçirmek ve amaçlarını anlamak önemlidir. Ayrıca, verilen özniteliğe kaydedilen verilerin sayısal mı yoksa kategorik mi olduğuna ve bireysel özniteliklerin değerlerinin hangi aralıkta olduğuna odaklanmak gereklidir.
- **Özet istatistiklerin hesaplanması**: Her öznitelik için temel özet istatistikler derlemek tavsiye edilir. Önerilen değerler, aşırı değerler (min, max), medyan veya ortalama, standart sapma ve diğerleri olabilir. Bu, verilen özniteliklerin ortalama değerlerini, en küçük ve en büyük değerlerini elde ettiğimiz ve öznitelikleri toplayarak ayrıntılı olarak tanımlayabileceğimiz çok önemli ve bilgilendirici bir adımdır.

- **Korelasyon analizi yapma**: Herhangi bir veri kümesi için korelasyon katsayılarının matrisini veya ısı haritasını oluşturmak tavsiye edilir. Bu matris, veri kümesinin tüm özniteliklerinin değerleri arasındaki ilişkileri ölçer ve bu nedenle veri kümesi üzerinde matematiksel modeller oluşturmanın ne kadar zor olacağını gösterir. Korelasyon analizi ayrıca görselleştirme için uygun öznitelikleri ve veri kümesi alt kümelerini tanımlamada da yardımcı olur.

Veri keşfi analizinin bir sonraki adımı, veri kümesinin gerçek görselleştirmesidir. Ancak, bu, bu elkitabının sonraki bölümünde sunulan çeşitli prensiplere dayalı etkili veri görselleştirmesinden bahsediyoruz.

Etkili veri görselleştirme

Veri görselleştirmesini etkili kılan şu üç unsur vardır:

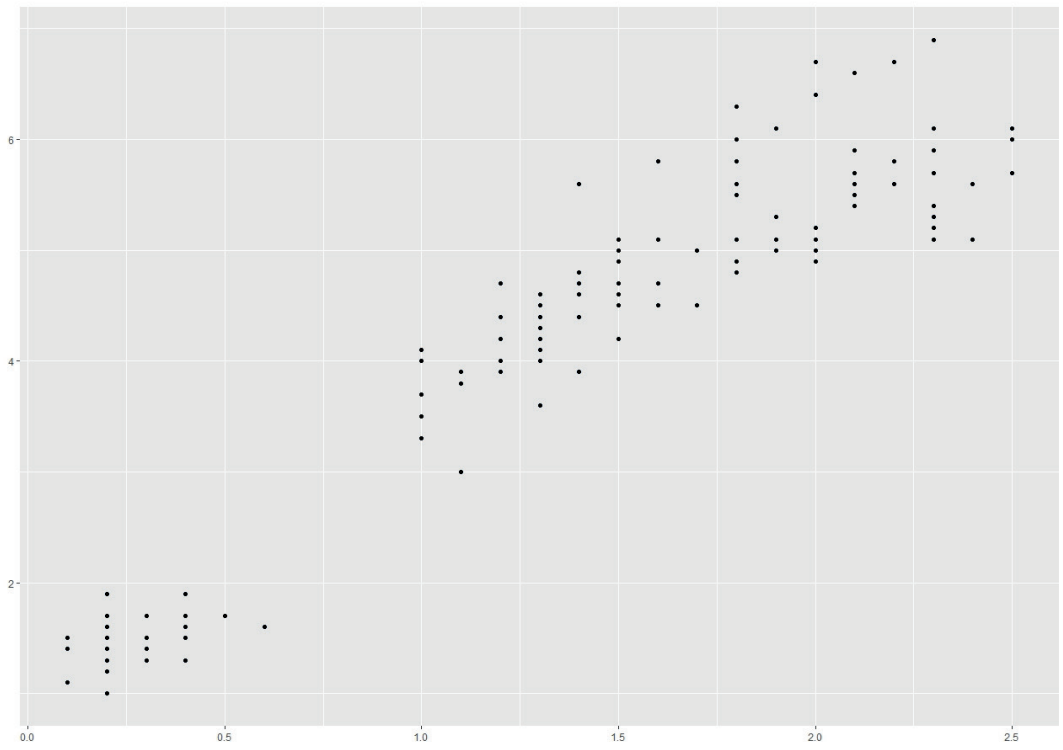
- Doğru veriyi görselleştirmek
- Doğru şekilde yapmak
- Doğru grafik türünü kullanmak

Bu üç unsur oldukça doğaldır. Veri analizi açısından görselleştirmeye uygun olarak adlandırılabilecek veriler, genellikle öngörü potansiyeli taşıyan verilerdir. El kitabının önceki bölümünden zaten biliyoruz ki, veri setlerinde öngörü potansiyelini kolayca araştırabiliriz ve bunu korelasyon analizi yöntemleriyle yapabiliriz. Bu nedenle, görselleştirmeye uygun veri setinin parçaları, özellik değerleri arasında güçlü korelasyonları veya ters korelasyonları tespit ettiğimiz yerler olacaktır (bkz. Korelasyon katsayılarının sonuçlarını yorumlama bölümü).

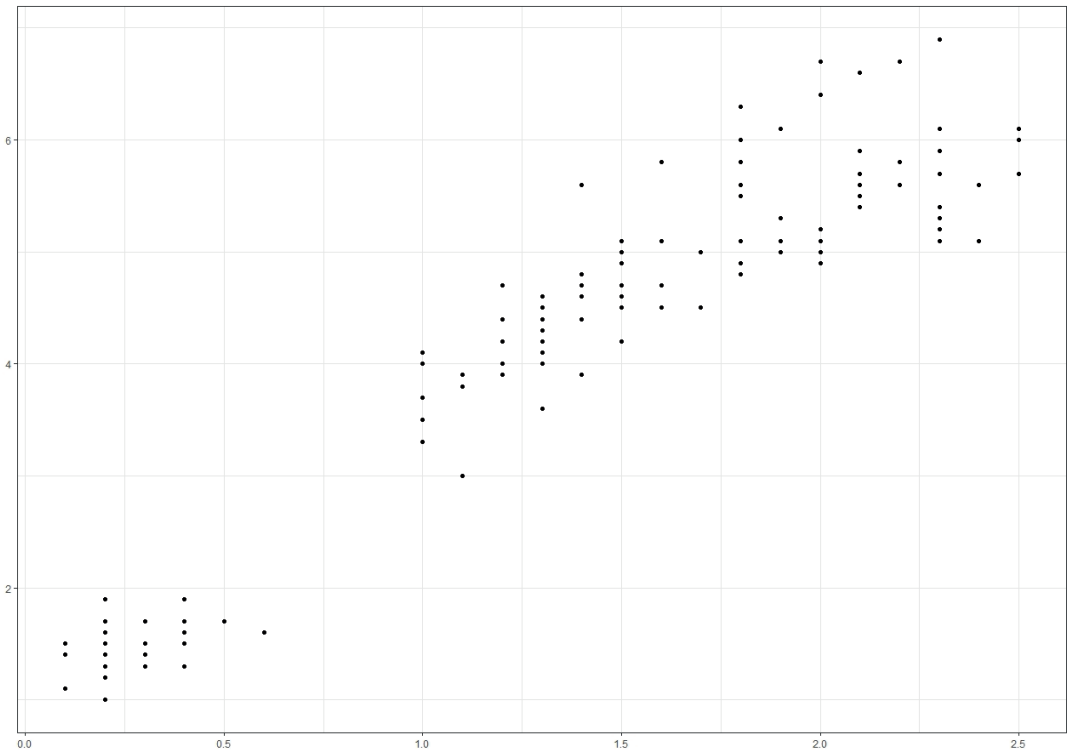
Bizim durumumuzda, **doğru veri görselleştirmesi ile**, veri setlerinin görselleştirmesinde yaygın olarak görülen iki sorunu elemeği kastediyoruz:

- **Verinin ve renk kullanımının oranını en üst düzeye çıkarmak** - Veriyi görselleştirmek istediğimiz için, ideal olarak, grafik minimum diğer grafik öğeleri içerecek (örneğin, bir arka plan rengi, belirgin bir ızgara vb.). Özellikle büyük nokta kümelerini görselleştirirken (potansiyel olarak birleşebilir veya çok küçük noktalarla veya başka bir nesne türüyle temsil edilebilirler) renk ve veri arasındaki oranı maksimize etmek önemlidir. Aşağıdaki şekil, R dilinde çizilmiş standart bir nokta grafiği içerir (alt şekil a) ve verileri daha görünür hale getirmek için bu grafiğin değiştirilmiş bir sürümünü içerir.

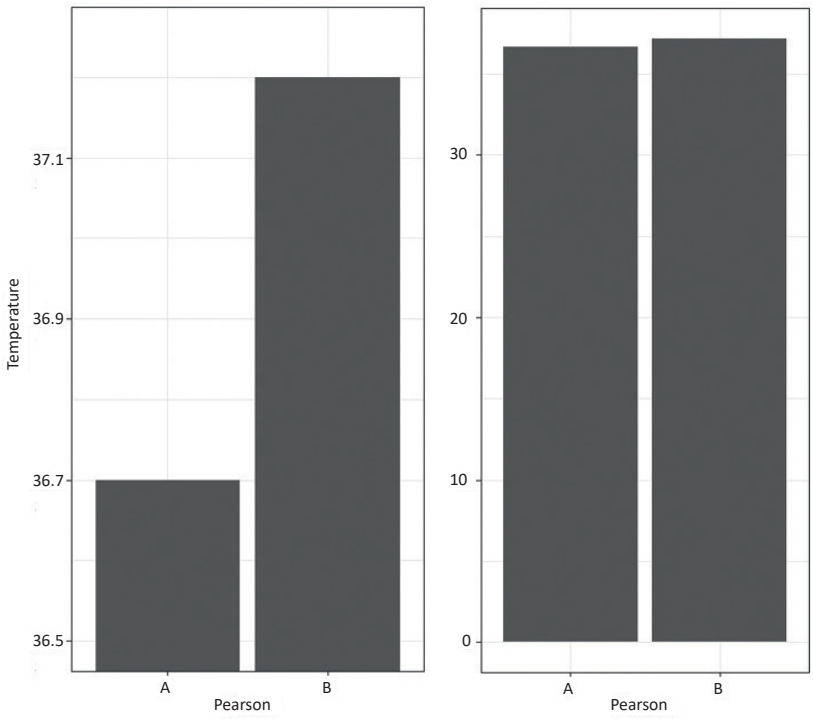
a)



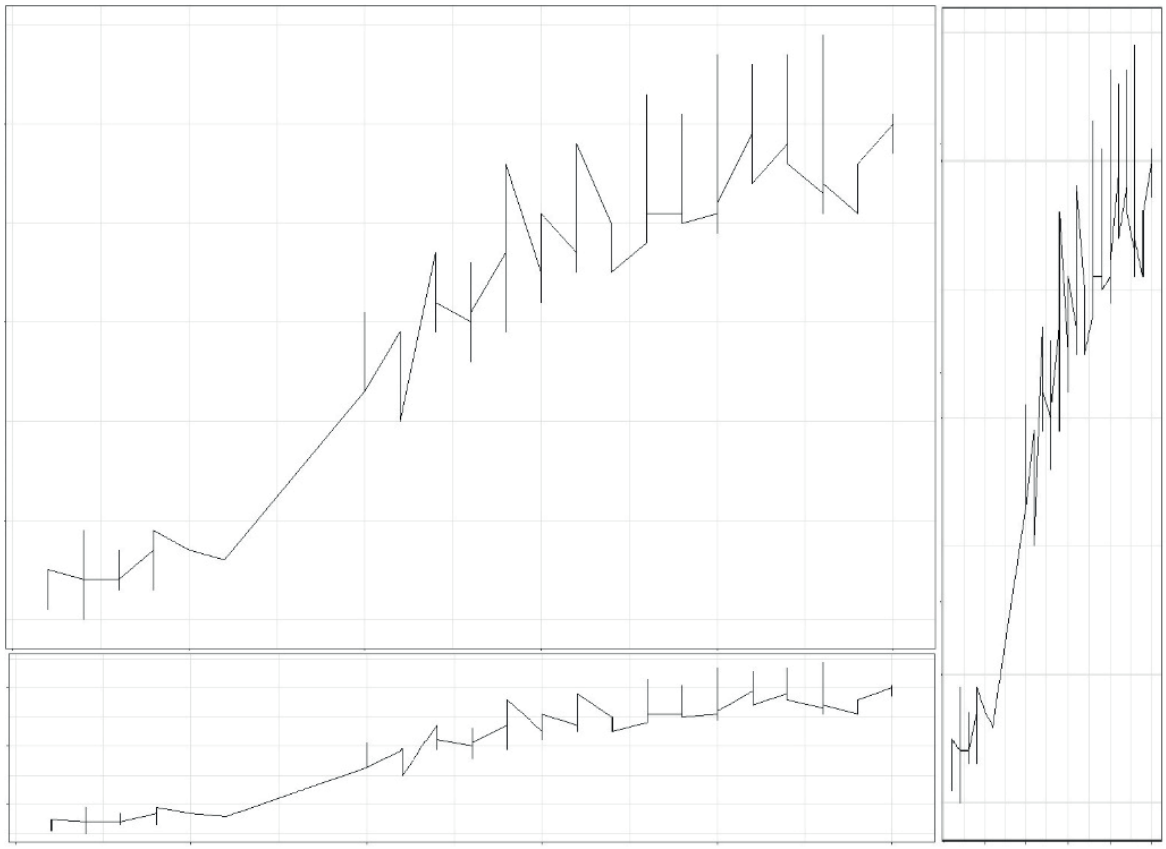
b)



- **Distorsiyonların ortadan kaldırılması** - Verilerin görsel sunumunda ve sonraki yorumlanmasında, eksiklikler sıklıkla eksik veya yanlış ekseni ayarlamadan veya görüntü dosyasının kendisini bozmadan kaynaklanır. Aşağıdaki şekiller her iki sorunu da göstermektedir. İlk çubuk grafiği iki kişinin (A ve B) sıcaklığını karşılaştırır. Sol grafikteki değerlerin önemli ölçüde farklı göründüğüne dikkat edin, sağdaki değerler ise çok benzer görünmektedir, bu da iki farklı eksen ayarıyla sunulan iki aynı ölçüm olmasına rağmen. Sol şekilde y yüzeyini sadece 36.5 ila 37.3 derece değerlerinden sunuyoruz. Sağ grafikte ise y değerlerini 0 ila 40 derece arasında sunuyoruz. Bu, veri görselleştirmesindeki bir karmaşa faktörüdür.



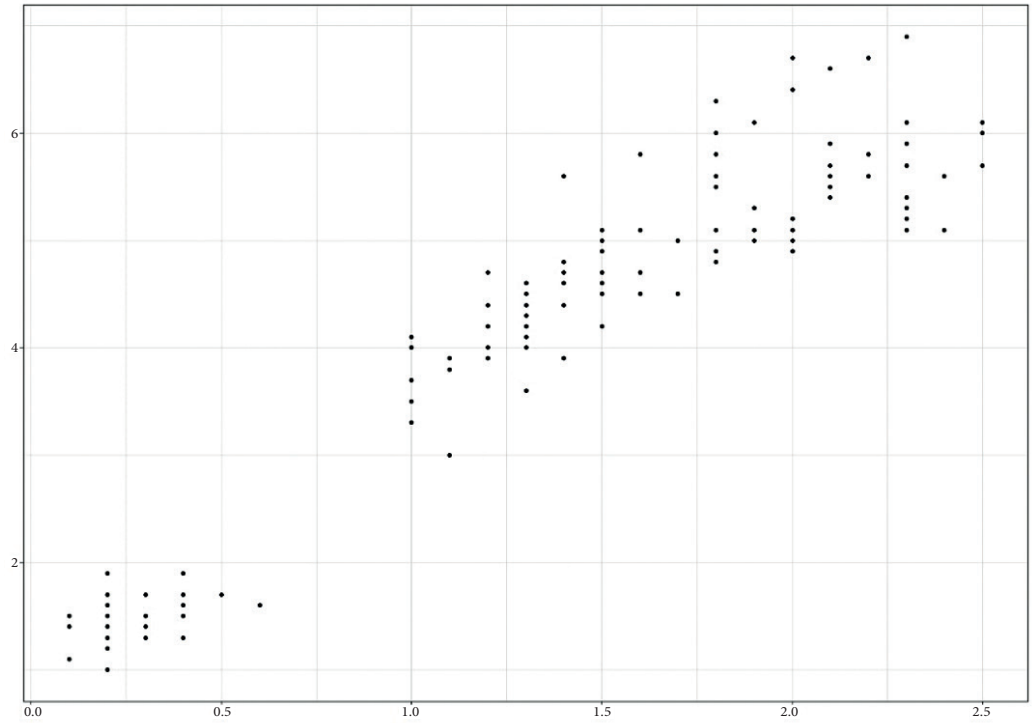
İkinci çarpıtma sorunu, resmin kendisinin çarpıtılmasıdır. Aşağıdaki görüntüde, bir grafiğin üç farklı en-boy oranında sunulduğunu görebiliriz. Sağ alttaki görüntü, verilen grafiğin gerçek şeklinin çarpıtılması nedeniyle uygunsuzdur ve bu nedenle bu veriler için ideal oran, en modern projektörlerin standart görüntü oranına yakın olan sol üstteki grafik olacaktır - 16:9.



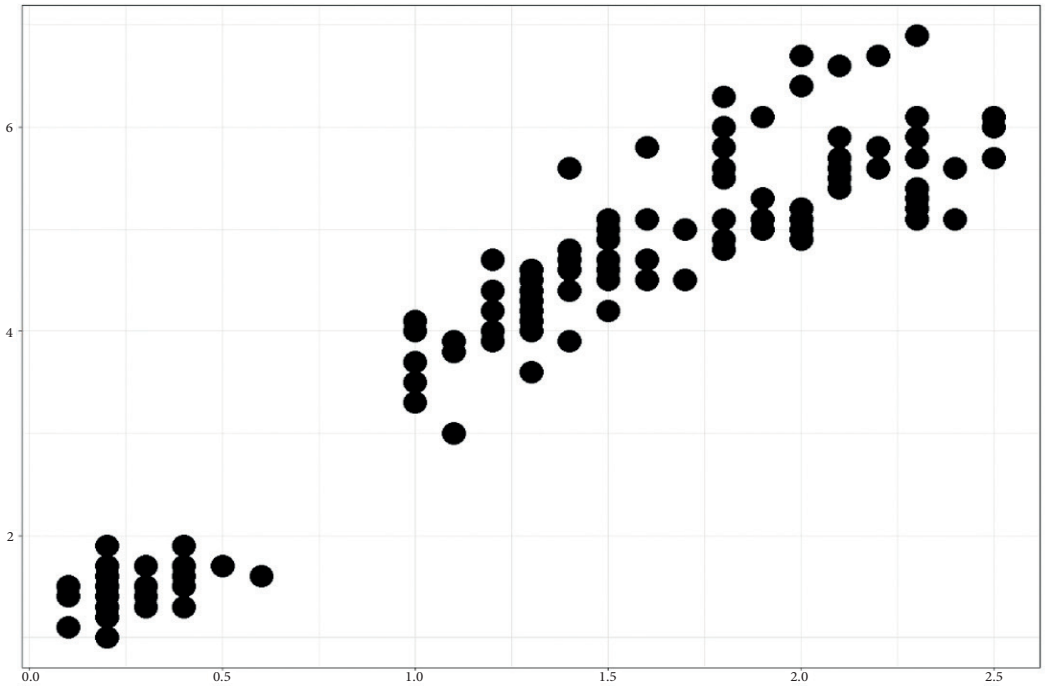
Veri görselleştirmenin etkinliğinin son derece önemli bir unsuru, doğru grafik türüdür. Genel olarak, dört tür grafik yaygındır - nokta, çizgi, pasta ve çubuk grafikleri. Her grafik türü farklı amaçlar için uygundur ve farklı avantajlara ve dezavantajlara sahiptir. Bu el kitabı bölümü yalnızca en yaygın iki veri görselleştirme yöntemine odaklanacaktır - nokta grafikleri ve çizgi grafikleri.

Nokta grafikleri

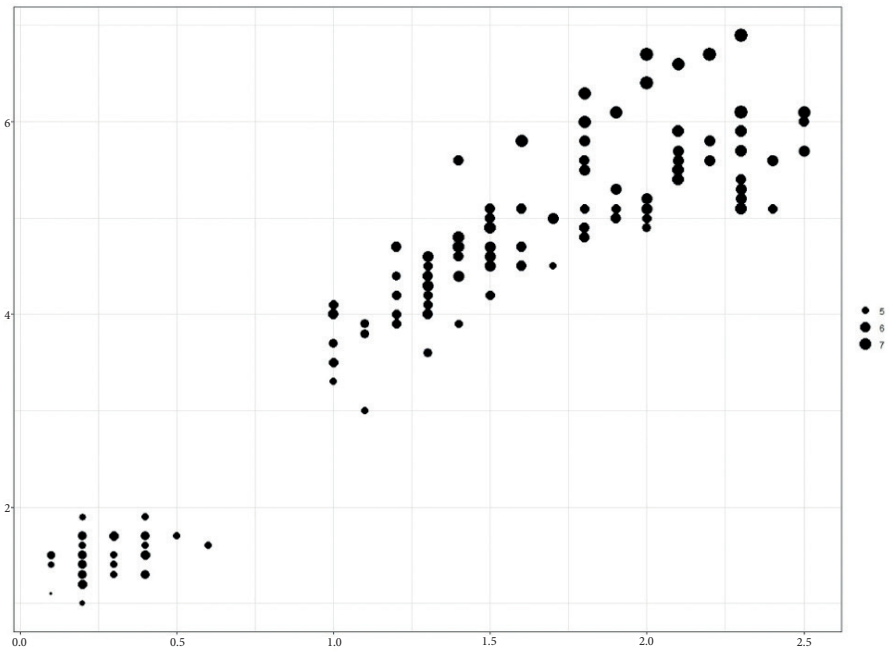
Nokta grafikleri, bir veri kümesinin iki (veya daha fazla) özelliği arasındaki ilişkiyi noktaları kullanarak görselleştirmek için kullanılır. Standart görselleştirme yöntemi, iki özelliğin değerlerini bir düzlemde karşılaştırmaktır:



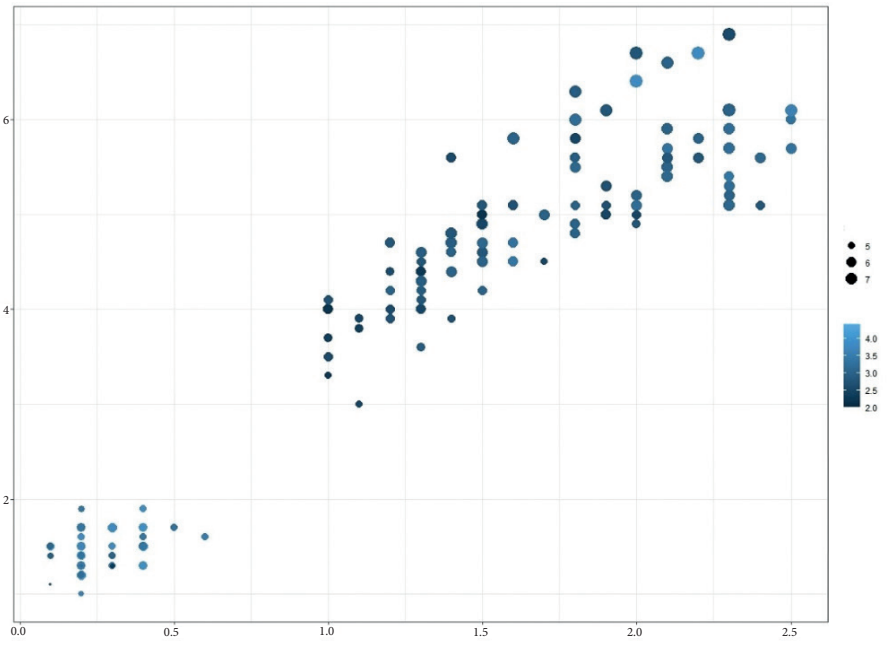
Veri görselleştirmesi için bu tür bir yaklaşım kullanıldığında, grafikteki nokta boyutuna dikkat etmemiz gerekmektedir. Aşağıdaki resim, büyük nokta boyutundan kaynaklanan grafikteki aşırı doymayı göstermektedir; benzer bir sorun, birbirine yakın birçok noktanın bulunduğu durumlarda ortaya çıkar.



Ancak, nokta boyutu, veri görselleştirmesi bağlamında ek bilgi iletmek için kullanılabilir. Noktanın boyutunu, veri kümesindeki üçüncü özelliğin boyutuyla doğru orantılı olarak ayarlarsak, veri kümesinin üç özelliği arasındaki ilişkiyi görselleştirebiliriz.

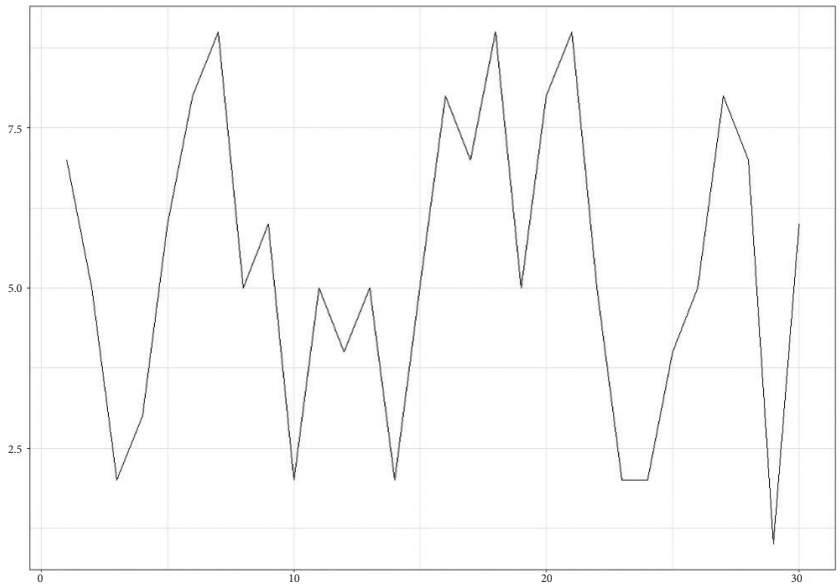


Bu kavramı noktaların diğer özellikleriyle (aşağıda listelenenler gibi) genişletebiliriz. Bu şekilde, veri kümesinin dört özelliği arasındaki ilişkiyi görselleştirebiliriz. Ancak, bu görselleştirme yöntemi büyük bir nokta veya özellik sayısıyla kullanmak zordur. Genel olarak, bir düzlemde üç veya dört özelliği görselleştirmek önerilmez.

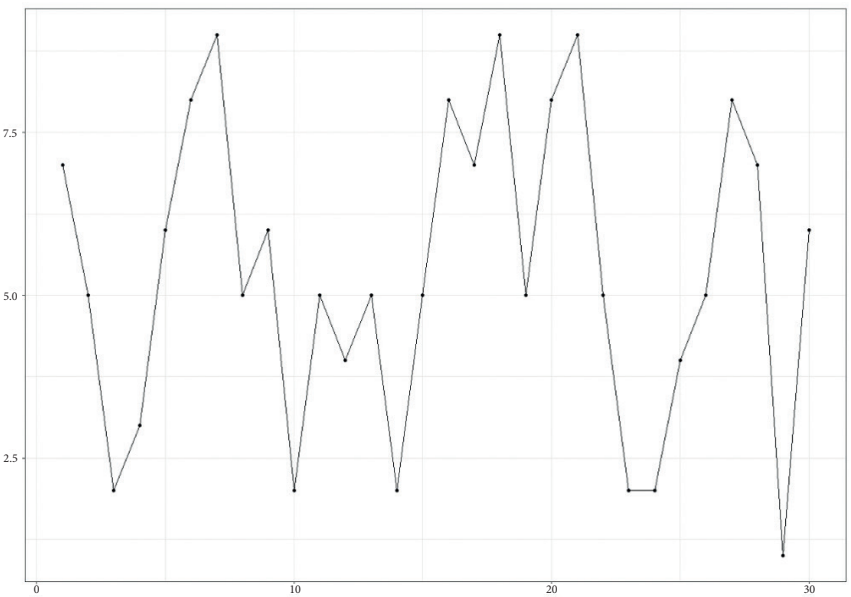


Çizgi grafikleri

Çizgi grafikler, bir özelliğin zaman içindeki değerini veya başka bir özelliğe bağlı olarak bir özelliğin değerindeki dalgalanmayı görselleştirmek için kullanılır. Çizgi grafiklerindeki çizgiler, noktaların bir yaklaşımıdır - kesikli veri noktalarının sürekli çizgilere dönüştürülmesi ve bu nedenle bazı yerlerde doğru olmayabilir. Nokta grafikleri ile karşılaştırıldığında, çizgi grafikleri kategorik (linguistik) veriler için kullanılması zor olabilir.



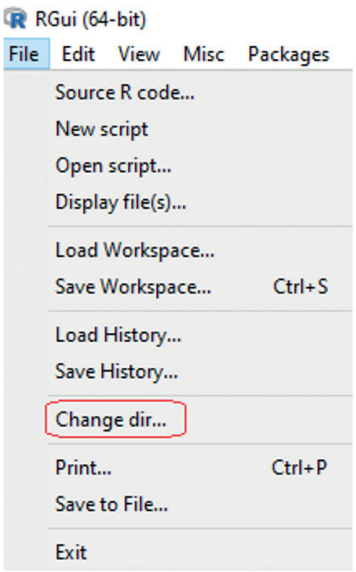
Çizgiler, veri noktalarının yaklaşımları olduğu için, çizgiyi ve çizginin dayandığı noktaları görselleştirmek önerilir. Bu, özellik değerlerinin doğruluğu konusundaki belirsizliği azaltır.



4.4 UYGULAMADA KEŞFEDİCİ VERİ ANALİZİ

Bu bölüm, R dilinde keşif veri analizi yöntemlerinin pratik uygulanmasına odaklanmıştır. Aşağıda verilen örnekler, *R 4.3.0* sürümünde uygulandı, ancak tüm veri analizi için uygun programlama araçlarının tüm sürümlerinde bu kavramlar ve komutlar bulunmaktadır.

İlk olarak, oturumun çalışma dizinini yapılandırılmış verilerin depolandığı dizine değiştirmemiz gerekiyor - bunu programın üst çubuğunu kullanarak yapabiliriz: *Dosya → Dizin değiştir → çalışma dizini ayarla*. Keşif veri analizi yöntemi sunumları için Apendiks A'da açıklanan Iris veri kümesini kullanacağız.



Çalışma dizinini değiştirdikten sonra veri kümesini R'e yüklemeye başlayabiliriz. Bu işlem birkaç farklı şekilde gerçekleştirilebilir. Biz (bizim açımızdan) en basit yolunu sunuyoruz - *read.table ve read.csv*) komutları, farklı türde giriş dosyaları için benzer şekilde çalışır. *read.csv* biçimindeki komut, veri analizi araçları için en yaygın yapılandırılmış giriş olan *.csv* dosyaları için kullanılır. *read.table* biçimi *.txt veya .data* formatındaki girişler için kullanılabilir.

```
read.table("title", header=T/F, sep="symbol")

read.csv("title", header=T/F, sep="symbol")
```

Burada, *title*, verilerimizin dosya adını ve dosya uzantısını içeren kısmı temsil eder. Komutun header kısmı, giriş veri dosyasının başlık içerip içermediğini belirtir *(T başlık içeriyorsa, F başlık içermiyorsa)*. Sep kısmı ise dosya içindeki özellik değerlerinin ayrıldığı karakteri bekler.

Veri kümesini program içinde daha fazla kullanabilmek için, onu seçtiğiniz bir başlık altında kaydetmemiz gerekir, örneğin, *title_of_data*:

```
title_of_data <- read.table("title", header=T/F, sep="symbol")
```

our_data.data' başlığı altında saklanan bir veri dosyasını yüklemenin bir örneği aşağıdaki gibi görünebilir. İkinci verilen komutta, veri setimizi 'data' başlığı altında kaydediyoruz:"

```
read.table("our_data.data", header=T, sep=",")
data <- read.table("our_data.data", header=T, sep=",")
```

Keşif Amaçlı Veri Analizi - Adım 1 - Belirli bir veri kümesine aşinalık

Dataseti tanımak için yapabileceğimiz ilk işlemlerden biri, R aracının konsolunda *title_of_data* kullanarak tüm veri kümesini listelemektir. Ancak, binlerce varlık içeren büyük veri kümeleri için bu pratik değildir. Bu nedenle, veri kümesinin başlangıcından belirtilen sayıda varlığı listeleyen head komutunun ikinci versiyonunu sunuyoruz.

```
title_of_data
head(title_of_data, number_of_entities)
```

Bu kavramın bir örneği, veri adıyla saklanan tüm veri kümesini listeleyen veya bu dosyanın ilk beş varlığını listeleyen bir örnek olabilir.

```
data
head(data, 5)
```

Bu komutun R dilindeki çıktısı, aşağıdaki biçimdeki bir sahte tablo olacaktır:

```
> data <- read.table("iris.data", header = T, sep = ",")
> head(data, 5)
  sepal_length sepal_width petal_length petal_width       class
1          5.1         3.5          1.4         0.2 Iris-setosa
2          4.9         3.0          1.4         0.2 Iris-setosa
3          4.7         3.2          1.3         0.2 Iris-setosa
4          4.6         3.1          1.5         0.2 Iris-setosa
5          5.0         3.6          1.4         0.2 Iris-setosa
```

Veri kümesi ile temel bir tanışma işleminden sonra, merkezlilik ve değişkenlik ölçümlerini hesaplamaya devam edebiliriz. Bu işlevlerin tümü, bireysel işlevlerin İngilizce versiyonundan türetilmiştir (*örneğin, standart sapma için sd*), ve bunların girişi yalnızca veri kümesinin bir özelliğidir ve şu formatta yazılır:

title_of _data$title_of _attribute.

Bu işlevlerin en çok yönlüsü, *tüm özellikler için minimum, 1. çeyrek, medyan, ortalama, 3. çeyrek ve maksimumu* ölçen summary işlevidir.

```
mean(title_of_data$attribute_title)

median(title_of_data$attribute_title)

min/max/sum(title_of_data$attribute_title)

sd(title_of_data$attribute_title)

summary(title_of_data)
```

Verilerde bulunan istatistiksel özelliklerin böyle bir analizine ilişkin bir örnek, aşağıdaki komutları kullanmaktır:

```
summary(data)

sd(data$attribute_title)
```

Iris veri kümesi üzerinde çalıştırılan bu işlevlerin çıktısı aşağıdaki değer kümesinden oluşur:

```
> summary(data)
  sepal_length    sepal_width     petal_length    petal_width          class
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   Length:150
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   Class :character
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   Mode  :character
 Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
> sd(data$sepal_length)
[1] 0.8280661
```

Keşifsel Veri Analizi - Adım 2 - Korelasyon Analizi

Bu elkitabının önceki bölümlerinde belirtildiği gibi, korelasyon analizi, keşifsel veri analizinin en önemli bileşenlerinden birisidir. Korelasyon analizinin temel komutu, iki veri kümesi özelliği arasındaki korelasyon katsayısının hesaplanmasıdır ve bunun için *cor işlevi* kullanılır. Aşağıda sunulan komut sözdiziminde, veri için hesaplamak istediğimiz korelasyon katsayısının türü, *method = korelasyon_türü* parametresini kullanarak tanımlanabilir. Bu komutta varsayılan olarak Pearson korelasyon katsayısı kullanılır.

```
cor(title_of_data$attribute_title_1, title_of_data$attribute_title_2)

cor(title_of_data$attribute_title_1, title_of_data$attribute_title_2,
method = “pearson”)

cor(title_of_data$attribute_title_1, title_of_data$attribute_title_2,
method = “spearman”)
```

Veri kümesinin tüm değişkenleri arasındaki ilişkileri incelemek için veri kümesinin tüm değişkenleri arasında bir korelasyon matrisi oluşturmak istiyoruz.

```
cor(title_of_data)
cor(title_of_data, method = “spearman”)
```

Iris veri kümesinin korelasyon analizi, aşağıdaki basit komutlar kullanılarak yapılabilir.

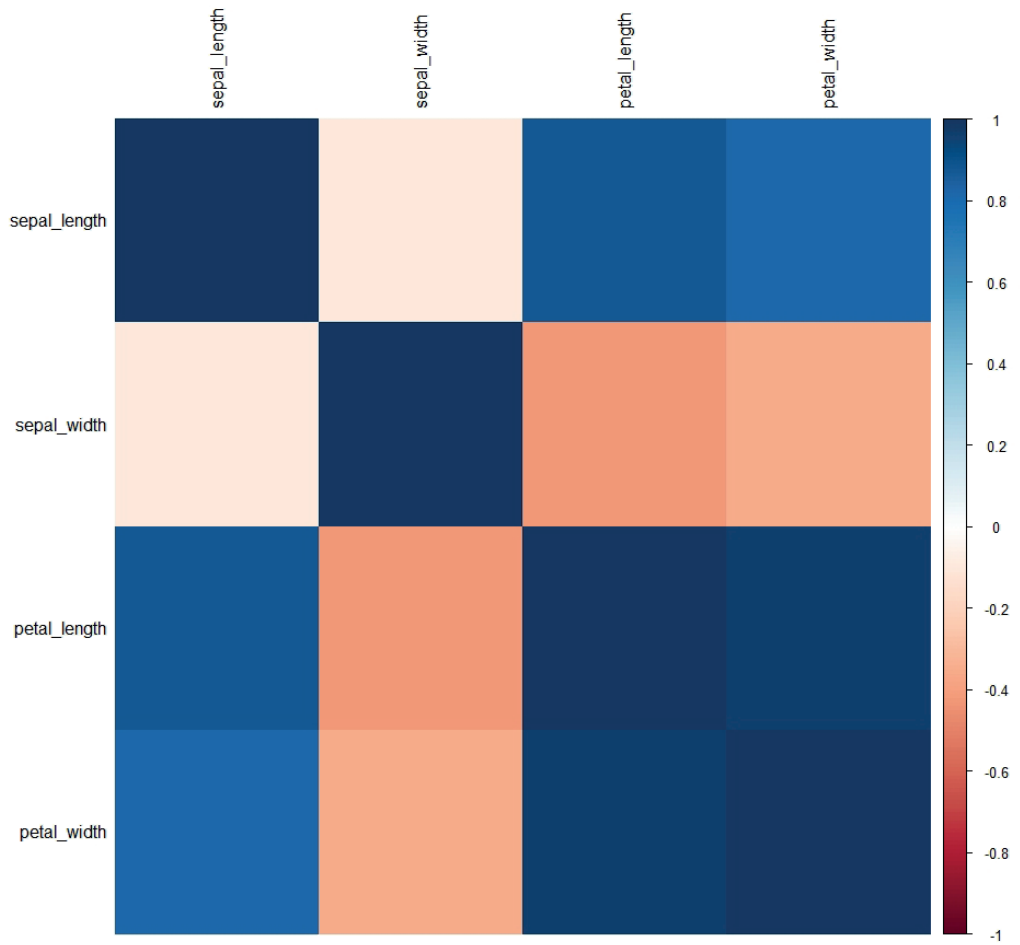
```
cor(data[, 1:4])
cor(data[, 1:4], method = “spearman”)
```

Not: Iris veri kümesi, değerleri dilbilgisel olan (attribute class) bir özelliği içerir ve korelasyon matrisi sadece sayısal değerleri içerir, bu nedenle cor işlevinin yalnızca ilk dört (sayısal) özelliği girmesi gerekmektedir. Bunu, data adlı bir veri kümesinden data[, 1:4] komutunu seçerek sağlarız.

```
> cor(data[,1:4])
             sepal_length sepal_width petal_length petal_width
sepal_length    1.0000000  -0.1093692    0.8717542   0.8179536
sepal_width    -0.1093692   1.0000000   -0.4205161  -0.3565441
petal_length    0.8717542  -0.4205161    1.0000000   0.9627571
petal_width     0.8179536  -0.3565441    0.9627571   1.0000000
> cor(data[,1:4], method = "spearman")
             sepal_length sepal_width petal_length petal_width
sepal_length    1.0000000  -0.1594565    0.8813864   0.8344207
sepal_width    -0.1594565   1.0000000   -0.3034206  -0.2775111
petal_length    0.8813864  -0.3034206    1.0000000   0.9360034
petal_width     0.8344207  -0.2775111    0.9360034   1.0000000
```

4.2 bölümünde belirtildiği gibi, büyük veri kümeleri için korelasyon ısı haritasını kullanması önerilir. Bu görselleştirme yöntemini kullanabilmek için R dili için işlevler içeren bir paket kurmamız gerekiyor. Bu paket, *corrplot* adlı bir korelasyon ısı haritası görselleştirme işlevini içerir. Bu paketi kurduktan sonra, *require* işlevini kullanarak yükleriz ve ardından bir korelasyon ısı haritası oluştururuz.

```
install.packages(“corrplot”)
require(corrplot)
corrplot(cor(data), method = “color”)
```



Iris veri kümesi için korelasyon matrisi ve ısı haritası, ileri veri analizinde kullanılabilecek ilişkileri bize anlatır. Özellikle, herhangi bir türdeki korelasyon katsayısı 0.8'den yüksek veya -0.8'den düşük olan tüm özellikler arasındaki ilişkilere ilgi duyarız.

- ρ(sepal_length, petal_length) ≈ 0.87
- ρ(sepal_length, petal_width) ≈ 0.82
- ρ(petal_length, petal_length) ≈ 0.94

Bu özellikler arasındaki ilişkiler görselleştirmeye değerdir.

Keşifsel Veri Analizi - Adım 3 - Veri Görselleştirme

Korelasyonları analiz ettikten sonra, güçlü bir korelasyon veya karşıtlık fark ettiğimiz özellik çiftlerini görselleştirmeye hazırız. Ancak görselleştirmeye başlamadan önce, görselleştirme amaçları için kullanılan bir paketi yüklememiz gerekiyor:

```
install.packages("ggplot2")
require(ggplot2)
```

ggplot2 paketi, veri görselleştirmede kullanılan en popüler paketlerden biridir ve nokta grafikleri, çizgi grafikleri, sütun grafikleri ve daha birçok grafik çizim işlevini içerir. Bu elkitabın bu bölümünde, bu işlevlerin bazı basit örneklerini seçiyoruz.

Nokta grafikleri

Verilerin en temel ve aynı zamanda en güçlü görselleştirmesi nokta grafikleridir. R dilinde ve *ggplot2* paketinde, fonksiyon girdi olarak birkaç değer beklerken herhangi bir grafik türünü oluşturmak için *ggplot* işlevini kullanırız. En basit grafikler için bu girdiler şunlardır:

- Çalıştığımız veri kümesinin başlığı (bizim durumumuzda, bu başlık ‹veri' olarak adlandırılır).
- „*Aes*“ terimi, İngilizce „*aesthetics*“ kelimesinden türetilmiş olup, genellikle bir ekseni temsil eden bilgileri iletmek için kullanılır ve genellikle „eksen_başlığı *(x veya y)* = ekseni temsil eden nitelik_başlığı“ formatında sunulur.
- Grafik türü.

Genel olarak, nokta grafikleri için komutların sözdizimi, grafik eksenlerine iki niteliğin atanmasını ve komutun bölümünü + *geom_point()* içermektedir. Bu tür komutlar için genelleştirilmiş sözdizini aşağıda sunulmuştur:

```
ggplot(title_of_data, aes(x = title_of_attribute_1, y = title_of_
attribute_2)) + geom_point()
```

Noktaların rengini değiştirmek için + geom_point() komut bölümünün uzantısını kullanarak yalnızca noktalar için geçerli olan ikinci bir aes bölümü ekleyebiliriz - yani + geom_point(aes(color = “*renk adı*”)). Bir alternatif olarak, rengi belirlemek yerine, + geom_point() komut bölümünde renk adı yerine çalıştığımız veri kümesinin nitelik adını belirterek aynı grafik içinde noktaların rengini değiştirebiliriz. Bu şekilde, seçilen niteliğin değerlerine bağlı olarak değişen noktaların rengi tarafından işaretlenen ek bir boyutta (nitelikler) iki boyutlu görselleştirme elde ederiz. Önceki alt bölümde sunulan etkili veri görselleştirme prensiplerine uygun olarak, komutun sonuna + *theme_bw*() seçeneğini ekleriz. Bu seçenek, grafik altındaki beyaz bir arka planı sağlayarak grafikte kullanılan veri ve renk arasındaki oranı maksimize eder.

```
ggplot(title_of_data, aes(x = title_of_attribute_1,

y = title_of_attribute_2))) + geom_point(aes(color = “color”))

+ theme_bw()

ggplot(title_of_data, aes(x = title_of_attribute_1, y = title_of_
attribute_2))) + geom_point(aes(color = title_of_attribute_3)) + theme_bw()
```

Bu yaklaşımın basit bir örneği aşağıda sunulduğu gibi yapılabilir. Ayrıca iki ek kavram tanıtıyoruz:

Oluşturulan grafik, seçilen bir ad altında kaydedilebilir, örneğin aşağıda graph1 olarak gösterildiği gibi.

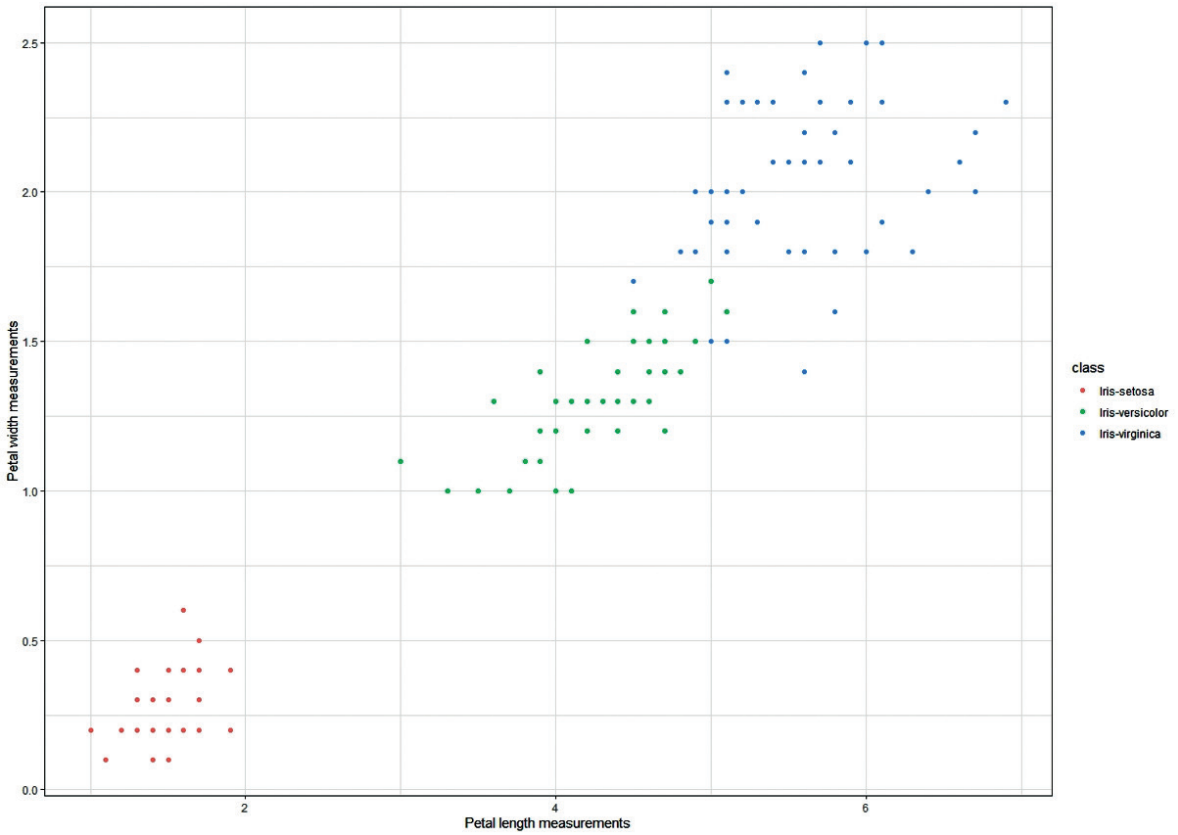
Bu şekilde kaydedilen grafiğe diğer komut bölümlerini ekleyebiliriz. Aşağıdaki örnekte, x ve y eksen etiketlerini eklemek için *+ xlab() ve + ylab()* kullanıyoruz. Grafik daha sonra adını konsolda çağırarak görselleştirilebilir.

```
graph1 <- ggplot(data, aes(x= atr_1, y = atr_2))
+ geom_point(aes(color = class) + theme_bw()
graph1 <- graph1 + xlab(“X parameter”) + ylab(“Y parameter”)
graph1
```

Iris veri kümesi için aşağıdaki kod

```
> require(ggplot2)
> graph1 <- ggplot(data, aes(x = petal_length, y = petal_width)) + geom_point(aes(color = class)) + theme_bw()
> graph1 <- graph1 + xlab("Petal length measurements") + ylab("Petal width measurements")
> graph1
```

aşağıdaki rakamı üretir:



Çizgi grafikleri

Sıkça kullanılan grafik türlerinden biri de çizgi grafiğidir ve bir özelliğin zaman içindeki değerini görselleştirmek veya bir özelliğin değer başka bir özelliğe bağımlı olarak nasıl dalgalanacağını görselleştirmek için kullanılır. *ggplot2* paketinde çizgi grafiği komutunun sözdizimi, elkitabının bu bölümünde sunulan önceki örneklerden önemli ölçüde farklı değildir. Tek fark, grafik çizilirken kullanılan geometri türüdür - çizgi grafikleri durumunda, + *geom_line()* kullanılır. *geom_line()* komutunun bölümünde *linetype* seçeneğini kullanarak, verileri çizdikten sonra kullanılan çizgi türünü değiştirebiliriz.

```
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) +
geom_line()
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line
(linetype = “dashed”)
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line
(linetype = “twodashed”)
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line
(linetype = “dotted”)
```

Nokta grafiklerine benzer şekilde, geometri - bu durumda çizgi - rengini kolayca değiştirebiliriz. Bu, grafikteki tüm çizgi türleri ile birleştirilebilen bir renk seçeneği ile yapılabilir.

```
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line(color = “color”)

ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line(linetype = “type”, color = “color”)
```

Bir çizgi grafiği her zaman veri noktalarının bir yaklaşımı olduğundan, çizgi grafiği yanı sıra noktaların kendilerini de görselleştirmek uygun olabilir. Bunun yapılması çok basit bir şekilde, hem çizgi hem de nokta geometrilerinin birleştirilmesi ile aşağıdaki gibi yapılabilir:

```
ggplot(title_of_data, aes(x=attribute_title_1, y=attribute_title_2)) + geom_
line() + geom_point()
```

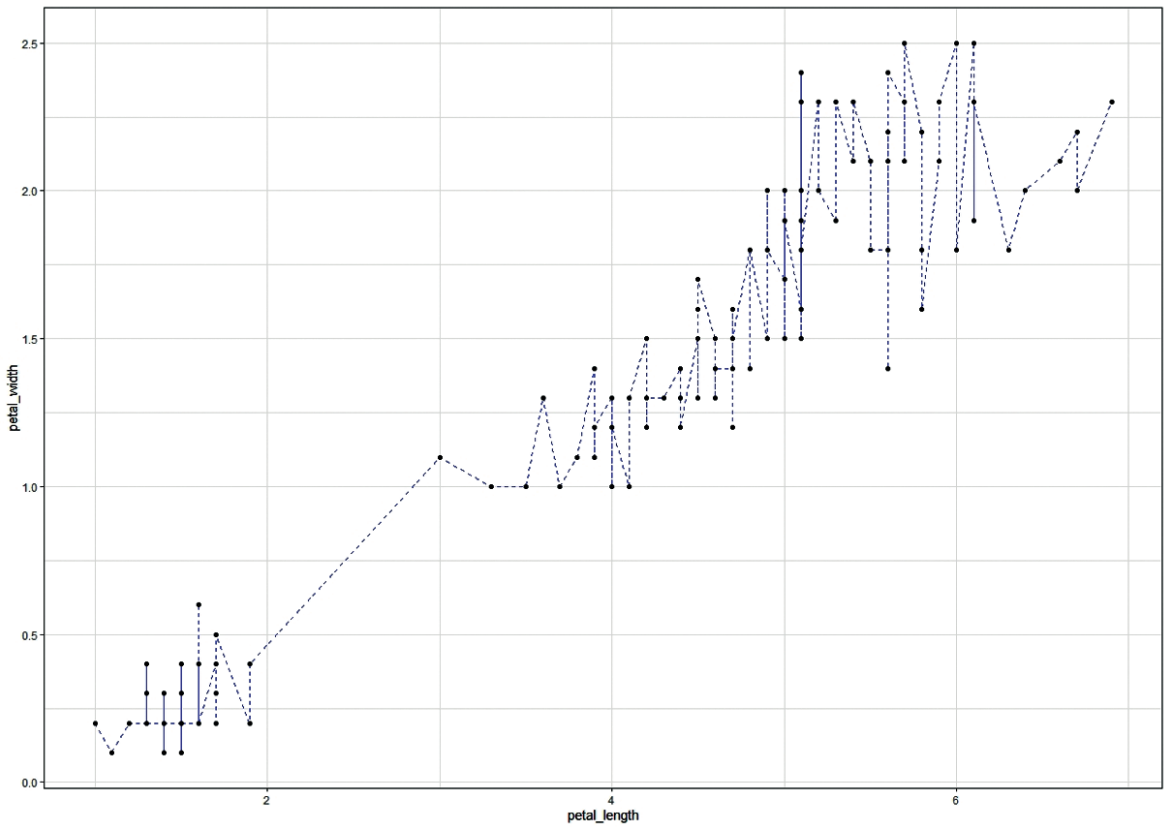
Anlaşılır bir şekilde, tüm bu seçeneklerin bir kombinasyonu mümkündür:

```
ggplot(data,   aes(x=attribute_title_1,   y=attribute_title_2))   +   geom_
line(linetype = “dashed”, color = “blue”) + geom_point()
```

Yani, Iris veri kümesi için aşağıdaki kod:

```
> ggplot(data, aes(x = petal_length, y = petal_width)) + geom_line(linetype = "dashed", color = "blue") +
+ theme_bw() + geom_point()
```

aşağıdaki rakamı oluşturur:



Çizgi grafiklerin gerçek gücü, bir niteliğin değerleri ile bir dizi niteliğin karşılaştırılmasını görselleştirme yeteneklerinde yatmaktadır - başka bir deyişle, birden fazla çizgi çizme olasılığı. Aşağıdaki sözdizimi örneğinde, *ggplot* işlevinin yalnızca bir niteliği (x-eksenine yerleştirileni) içerdiğini görebiliriz ve y-eksenini attribute2 ve attribute3'ün değerlerini görselleştirmek için kullanıyoruz. Bu görselleştirme, ayrı + *geom_line()* kod bölümleri tarafından gerçekleştirilir. Ayrıca, yukarıdaki seçeneklerin birkaç kombinasyonunu da bu yaklaşımla sunuyoruz.

```
ggplot(title_of_data, aes(x=attribute_title_1))
+ geom_line(aes(y= attribute_title_2))
+ geom_line(aes(y= attribute_title_3))
```

```
ggplot(title_of_data, aes(x=attribute_title_1))
+ geom_line(aes(y= attribute_title_2), color = “color”)
+ geom_line(aes(y= attribute_title_3), color = “color”)

ggplot(title_of_data, aes(x=attribute_title_1))
+ geom_line(aes(y= attribute_title_2), linetype = “type”, color = “color”)
+ geom_line(aes(y= attribute_title_3), linetype = “type”, color = “color”)
```

İris veri kümesine sahip olalım, burada üç özellik arasında güçlü ilişkiler ölçtük - çanak yaprağı uzunluğu, taç yaprağı uzunluğu ve taç yaprağı genişliği. Çanak yaprağı uzunluğu ve taç yaprağı genişliğinin çanak yaprağı uzunluğuna bağlı olarak dalgalanması, iki çizgi kullanılarak görselleştirilebilir; bu çizgiler türleri veya renkleriyle ayrılacaktır. Bizim durumumuzda:

Çanak yaprağı uzunluğuna bağlı olarak dalgalanmanın görselleştirilmesi için kırmızı çizgi kullanılarak yapılır,

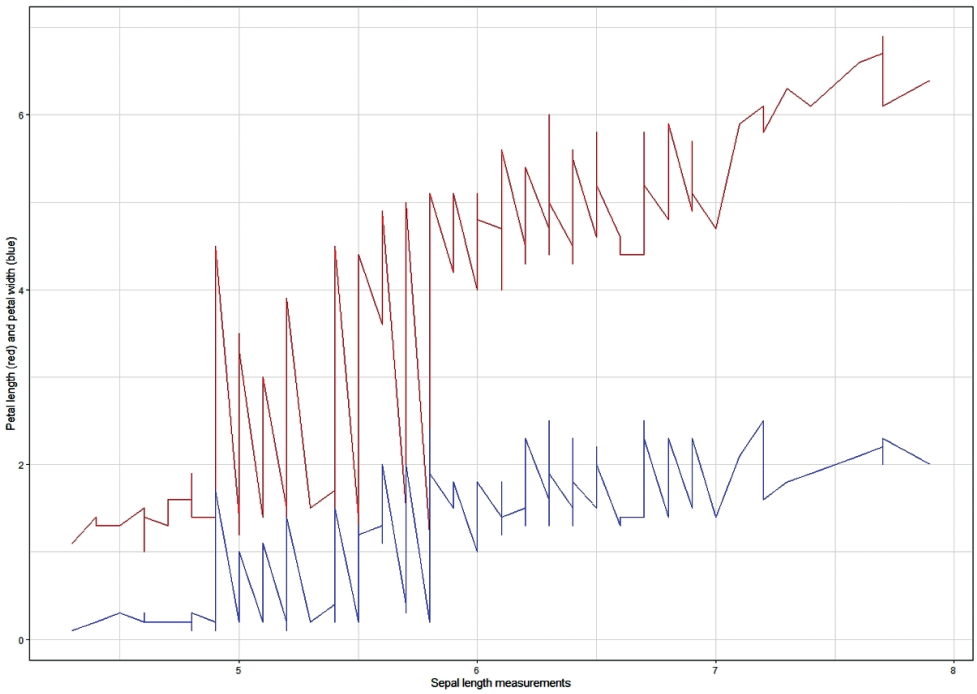
Çanak yaprağı genişliğine bağlı olarak dalgalanmanın görselleştirilmesi için mavi çizgi kullanılarak yapılır.

```
ggplot(data, aes(x=attribute_title_1)) + geom_line(aes
(y= attribute_title_2), linetype = “dotted”, color = “red”)
+ geom_line(aes(y= attribute_title_3), color = “blue”)
So, the following code for the Iris dataset
```

Yani, Iris veri kümesi için aşağıdaki kod

```
> ggplot(data, aes(x = sepal_length)) + geom_line(aes(y = petal_length), color = "red") +
+ geom_line(aes(y = petal_width), color = "blue") + theme_bw() + xlab( "Sepal length measurements") +
+ ylab("Petal length (red) and petal width (blue)")
```

aşağıdaki rakamı oluşturur.



Böyle bir veri görselleştirmesi, verilerdeki eğilimleri ve desenleri aramak için kullanılabilir, bu da daha karmaşık veri analizi yaklaşımlarında bu elkitabının diğer bölümlerinde sunulan biçimde kullanılabilir. Keşifsel veri analizinin bir eksikliği vardır - gerçekten Büyük Veri Kümeleri üzerinde kullanılması zordur, bunların düzgün bir şekilde analiz edilmesi için boyutsal azaltma ve diğer tekniklere ihtiyaç duyar.

BÖLÜM 5

BELİRSİZ KÜMELER

Bu elkitabın bu bölümü, Slovakya'nın Banská Bystrica şehrinde yer alan Matej Bel Üniversitesi Doğa Bilimleri Fakültesi Bilgisayar Bilimleri Bölümü'nden Alžbeta Michalíková tarafından yazılmıştır.

Günlük yaşamda genellikle "genç bir insan," "az tuz," "biraz sağda," "kuvvetli rüzgar," "yüksek sıcaklık," "düşük fiyat" gibi belirsiz ifadeler kullanırız. Bu ifadeler kesin sınırlara sahip değildir. Net bir şekilde tanımlanmamışlardır. Onları belirsiz veya flu olarak adlandırabiliriz. Matematiksel modelleme ile belirsiz kavramlar aracılığıyla ilk fikir, Lotfi A. Zadeh'in [1] makalesinde bulunabilir.

ZADEH, L. A.: *Fuzzy sets*. Information and control. Volume 8, pp. 338-353, 1965.

Bulanık kümeler kullanmak, **doğal dil ile çalışmak** olarak adlandırılır (sadece sayılarla değil, insan konuşma terimleriyle de çalışmak). Aynı zamanda, farklı insanların farklı kavramları farklı şekillerde algıladığını anlamak anlamına gelir. Aşağıdaki metnin ana bölümü, Slovakya'nın Banská Bystrica şehrindeki Matej Bel Üniversitesi, Uygulamalı Bilimler Fakültesi, Bilgisayar Bilimleri Bölümü'nde uygulamalı bilişim öğrencileri için tasarlanmış üniversite ders kitabına **("[2] Bilgisayar Bilimlerinde Bulanık Küme (Slovakça)")** dayanmaktadır.

Neden bulanık mantık kullanıyorsunuz?

- Bulanık mantık fikrinin anlaşılması kolaydır,
- Yanlış verilere toleranslı, esnek bir sistemdir,
- uzmanların tecrübeleri ile çalışılabilir,
- Herhangi bir karmaşıklıkta doğrusal olmayan bir sistemi modelleyebilir,
- standart teknik ekipmanlarda kullanılabilir.

Bulanık kümelerin kullanımları

- Uzman sistemler,
- nesnelerin tanınması ve sınıflandırılması,
- kontrol ve düzenleme teorisi,
- veritabanı sistemleri,

- matematiksel modelleme,
- son zamanlarda – açıklanabilir sinir ağları.

Bulanık kümelerin uygulama alanları

Bulanık küme alanları Belirsizlik hesaplamaya dahil edildiğinde kullanılır. Genellikle ekonomik açıdan pahalı cihazları temsil etmeyen cihazlarda kullanılırlar, örneğin, elektronik ev aletleri (çamaşır makineleri, mikrodalga fırınlar, elektrikli süpürgeler, traş makineleri, basınç ölçerler, ...). Ayrıca onları karmaşık ve ekonomik olarak yoğun cihazlarda da bulabiliriz,

- Japonya'da metro sürüşü - (Sendai şehri - 1988'den itibaren) [3],
- Patlama fırını kontrolü (geleneksel düzenleyicilerden daha verimli bir şekilde kontrol edilebilen sıcaklık kontrolü),
- Nükleer enerji santrallerinin yönetimi [4].

Örneğin:Şöyle bir iş pozisyonu için bir ilanımız olsun ki adayların yaşı 20-30 yaş aralığında olmalıdır. Bu küme nasıl tanımlanır?

Evren nedir?

Bu küme karakteristik fonksiyonuyla nasıl tanımlanabilir?

Yarın 31. yaşını kutlayacak bir kişi bu ilana başvurabilir mi?

Evren nedir?

Bu küme genellikle X harfi ile gösterilir. Evren, erişilebilir değerlere sahip herhangi bir aralık olmalıdır. Örneğin, görevimizde şöyle olabilir X=(0,∞).

Bu küme karakteristik fonksiyonuyla nasıl tanımlanabilir?

Karakteristik bir fonksiyon, düşünülen küme içinde bulunan öğelere sayı 1 atayan ve diğer taraftan düşünülen kümeye ait olmayan öğelere sayı 0 atayan bir fonksiyondur. Bu fonksiyon genellikle χ harfi ile gösterilir. Bizim örneğimiz için, fonksiyon aşağıdaki gösterime sahiptir.

$$\chi_A : \mathbb{X} \to \{0, 1\} \qquad \chi_A(x) = \begin{cases} 1, & \text{if } 20 \le x \le 30, \\ 0, & \text{if } 0 \le x < 20 \text{ or } x > 30. \end{cases}$$

Yarın 31. yaş gününü kutlayacak bir kişi bu ilana cevap verebilir mi?

HAYIR! - Çünkü birisi cevaplarını okuduğunda artık gerekli koşulu karşılamayacaktır.

Örnek: Benzer bir durumumuz olsun: Bir ilanda, şirket genç insanlar arıyor.

Önceki örnekle durum değişti mi?

Evren nedir?

Bu kümeyi nasıl tanımlayabiliriz?

Yarın 31. yaş gününü kutlayacak bir kişi bu ilana cevap verebilir mi?

Önceki örnekte durum değişti mi?

EVET! – gençlerin kümesi **bulanık kümeyi** temsil eder. Bu sete ait insanlar için keskin bir sınır yoktur!

Evren nedir?

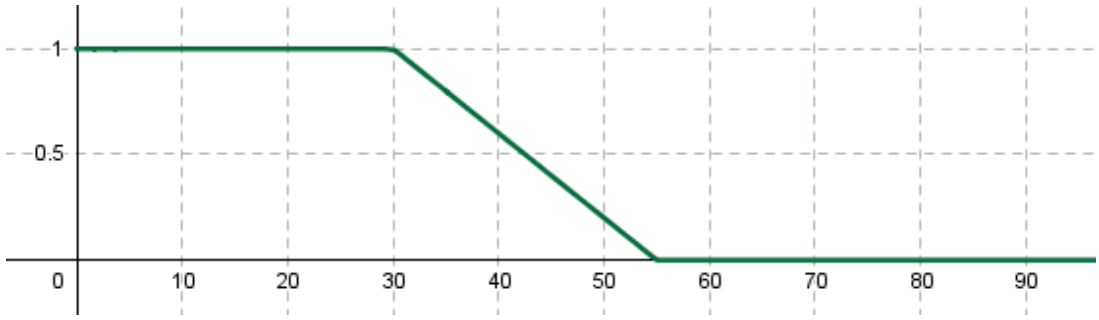
Bulanık kümenin evreni, ulaşılabilir değerlere sahip herhangi bir aralık olmalıdır. Bu, klasik küme gibi aynı küme olabilir, örneğin X=(0, ∞).

Bu seti nasıl tanımlayabiliriz?

Örneğin, 20 yaşındaki bir kişi kesinlikle gençtir. Bu nedenle, gençlik derecesi 1 olarak atanır. Benzer şekilde, 30 yaşındaki bir kişi kesinlikle genç olarak kabul edilebilir. Bu nedenle, gençlik derecesi 1 olarak atanır, ancak 35 yaşındaki bir kişiye gençlik derecesi 0.8 atanabilir... Bulanık kümeleri tanımlamak için, bu tür üyelik fonksiyonları adı verilen fonksiyonları kullanırız. Bunlar μ harfi ile gösterilir. Bu fonksiyonu kullanarak, her evren öğesine birim aralıktan (örneğin (0,1) aralığından) bazı değerler atamamız gerekmektedir. İlk olarak, kesin olarak genç bir kişi yaşı olarak kabul ettiğimiz yaş değerlerine bakalım. Bir örnek olarak, bu değerler (0,30) aralığından olabilir. Üyelik fonksiyonu bu değerlere 1'e eşit bir değer atar. Şimdi, kesin olarak genç bir kişi yaşı olarak kabul ettiğimiz yaş değerlerine bakalım. Bu tür değerlere bir örnek, 55'ten büyük değerler olabilir. Bu değerlere, üyelik fonksiyonu 0'a eşit bir değer atar. (30,55) aralığındaki değerler için, genç kişiler kümesinin üyelik değerlerinin 1'den 0'a sıralı olarak azalmasını bekleriz (Bkz. Şekil 1).

"Genç insanların bulanık kümesini B ile gösterelim. Tanımlanan fonksiyonun tarifi şöyledir:"

$$\mu_B : \mathbb{X} \to \langle 0, 1 \rangle \qquad \mu_B(x) = \begin{cases} 1, & \text{if } x \in \langle 0, 30 \rangle, \\ \frac{1}{25}(55 - x), & \text{if } x \in (30, 55\rangle, \\ 0, & \text{if } x > 55. \end{cases}$$



Şekil 1. Bulanık genç kümesinin üyelik fonksiyonu

Notlar:

Bulanık küme ve üyelik fonksiyonu terimleri sıklıkla eşdeğer kabul edilir. Belirli bir giriş değerine atanan değere **üyeliğin derecesi ve ya üyelik derecesi** denir.

Formül (1) tarafından belirlenen bulanık küme, genç insanların kümesini tanımlasın. 20, 35 ve 45 yaşındaki insanlara üyelik derecesi atayabilir miyiz? Formül (1) kullanarak şunları elde ederiz:

(20) = 1, yani 20 yaşındaki bir kişi kesinlikle gençtir,

(35) = 0.8, yani 35 yaşındaki bir kişi %0.8 derecesiyle gençtir,

μB (40) = 0.6, yani 40 yaşındaki bir kişi %0.6 derecesiyle gençtir.

Yarın 31. yaşını dolduracak biri bu ilana cevap verebilir mi?

EVET! – çünkü μB bulanık kümesine ait üyelik fonksiyonu derecesi 0,96'ya eşittir (çünkü μB(31) =0,96). Bu değer, bulanık genç kümesindeki yüksek derecede üyeliği temsil etmektedir.

Bulanık kümelerin üyelik fonksiyonlarının birçok türü bulunmaktadır. İlerleyen örnekte bunlardan bazılarını göstereceğiz.

Örnek: "Yaklaşık 7"yi temsil eden gerçek sayılar bulanık C kümesini modelleyelim.

Evren nedir?

Bu kümenin özelliklerini nasıl tanımlayabiliriz?

Bu terimin kullanımıyla "yaklaşık 7" gibi cümleler hayal edebiliriz.

Dışarısı yaklaşık 7 santigrat derece.

veya

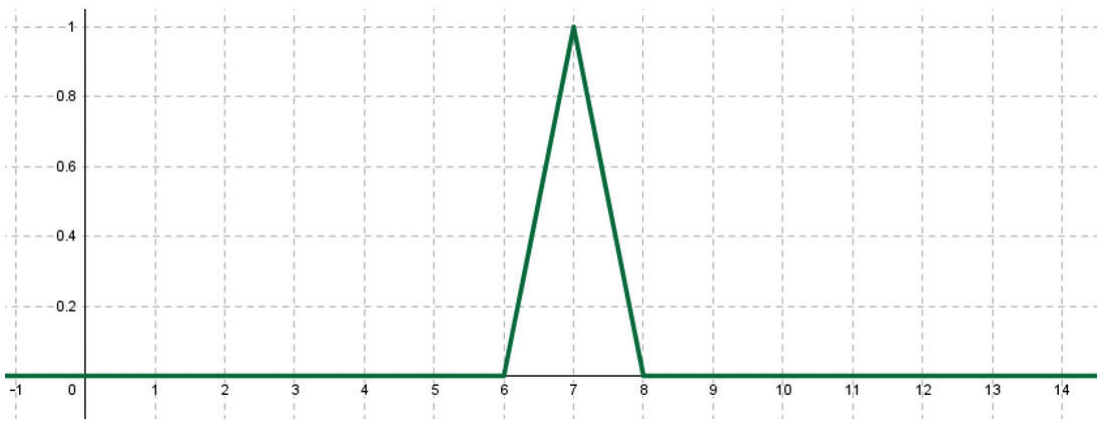
Mağazada yaklaşık 7€ harcadım.

Evren nedir?

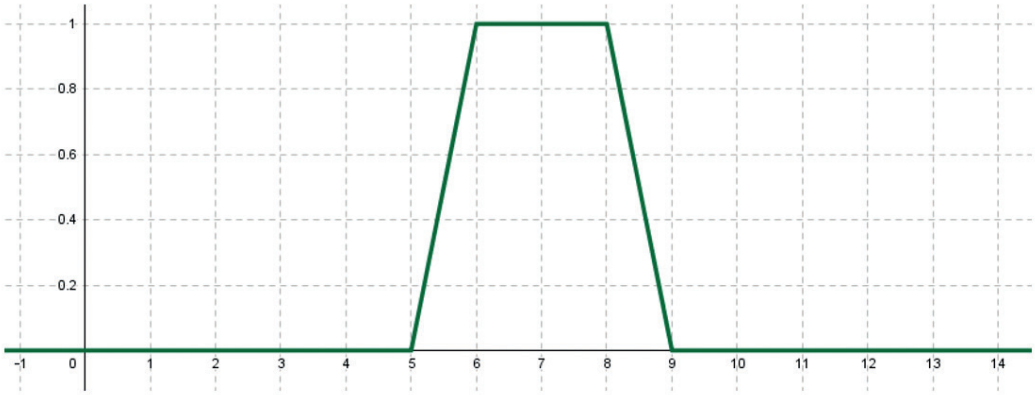
Bir evren olarak genellikle mümkün olan en büyük kümeyi düşünürüz. Bu örnekte, tüm gerçek sayılar set olabilir.

Bu kümenin özelliklerini nasıl tanımlayabiliriz?

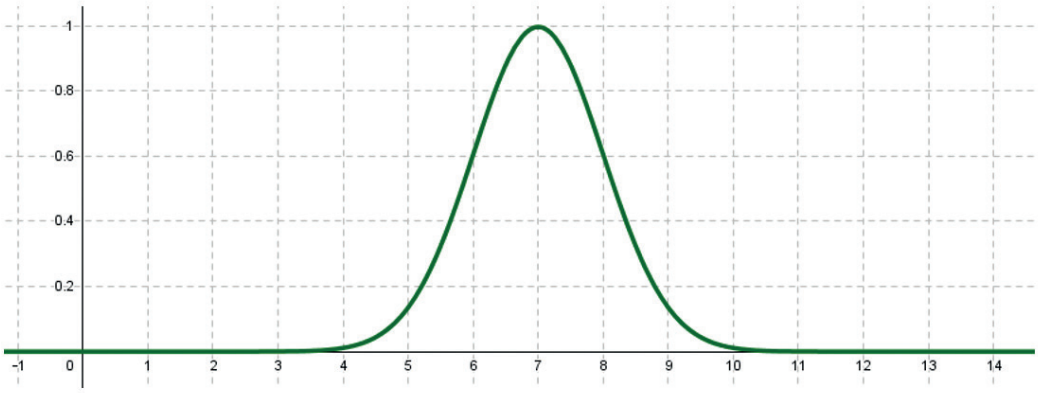
Bu kümeyi C kümesi olarak adlandıralım. Bu kümenin üyelik fonksiyonu için, μ_C iki koşulu sağlayabilir: μ_C (7) = 1, |x-7| artan bir farkla değerleri sıfıra düşmelidir. Ardından, bu olasılıkların bazıları Şekil 2, Şekil 3 ve Şekil 4'te gösterilmiştir.



Şekil 2. "Yaklaşık 7" değerini temsil eden üçgen üyelik fonksiyonu



Şekil 3. "Yaklaşık 7" değerini temsil eden yamuk üyelik fonksiyonu



Şekil 4. Yaklaşık 7 değerini temsil eden bir diğer üyelik fonksiyonu

Üyelik fonksiyonlarının türleri

Şekil 2-4'te üyelik fonksiyonunun birçok farklı formda olabileceğini gördük. Bunlardan bazılarının MATLAB yazılım aracında tanımlanmış olduğunu göstereceğiz.

Doğrusal üyelik fonksiyonları

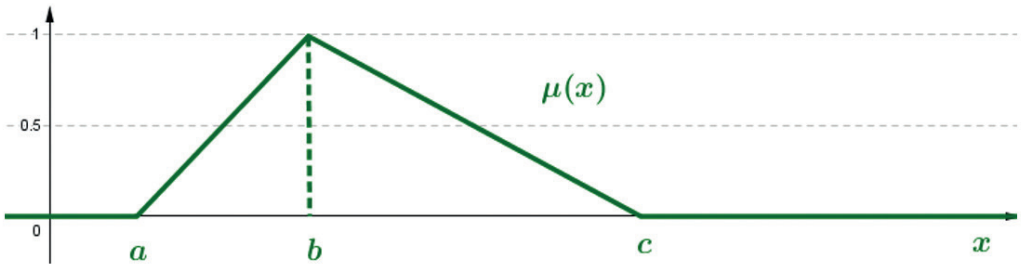
Doğrusal üyelik fonksiyonları, en basit üyelik fonksiyonları türünü temsil eder. Bunlar düz çizgilerin parçaları kullanılarak oluşturulur. İki temel gruba ayrılırlar:

- üçgen,
- yamuk.

Üçgen üyelik fonksiyonu

Üçgen üyelik fonksiyonu dört bölümden oluşur (Şekil 5'e bakın). İlk bölüm, giriş değerlerine eşit olan bir çıkış değeri atar (Şekil 5'teki *(-∞ a)* aralığı). İkinci bölüm, değeri 0'dan değeri 1'e doğrusal olarak artar (Şekil 5'teki *(a; b)* aralığı). Üçüncü bölüm, değeri 1'den değeri 0'a doğrusal olarak azalır (Şekil 5'teki *(b, c)* aralığı). Son bölüm yine giriş değerlerine eşit bir çıkış değeri atar (Şekil 5'teki *(c, ∞)* aralığı). Bu üyelik fonksiyonu genellikle *a, b, c* olmak üzere üç parametre ile tanımlanır. MATLAB yazılımında **trimf** olarak gösterilir ve parametreler için ***[a b c]*** gösterimini kullanırız. Üçgen üyelik fonksiyonunun

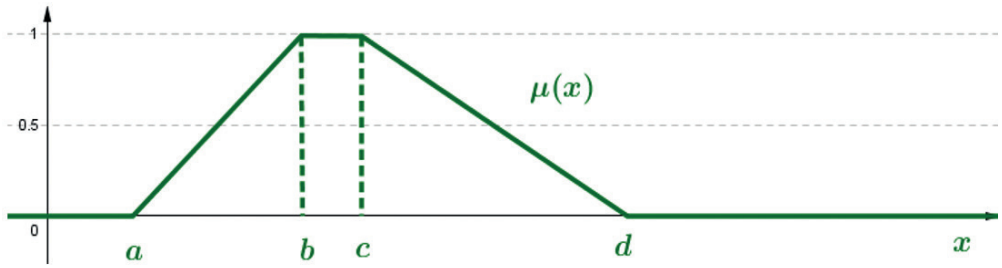
dikkat çekici bir özelliği, yalnızca bir giriş için (belirli olarak giriş değeri b için) çıkış değerinin *1*'e eşit olduğudur.



Şekil 5. Genel üçgen üyelik fonksiyonu Trapez üyelik fonksiyonu

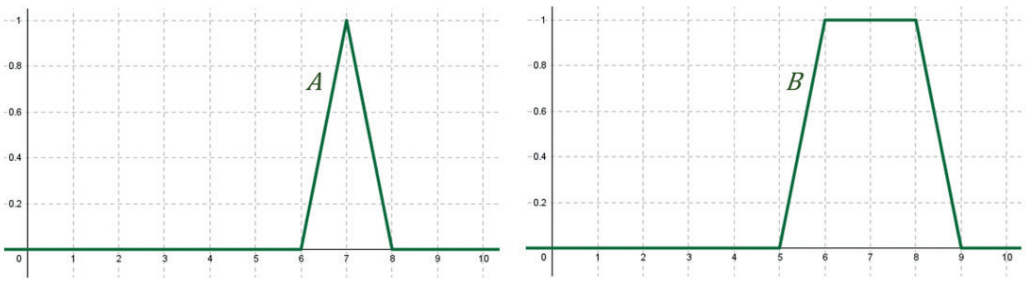
Trapez üyelik fonksiyonu

Dörtgen üyelik fonksiyonu beş bölümden oluşur (Şekil 6'ya bakın). Üçgen üyelik fonksiyonunun aksine, bu fonksiyon çıkış değeri 1'e ulaşan giriş değerlerinin aralığından oluşur. Bu üyelik fonksiyonu genellikle **a, b, c, d** olmak üzere dört parametre ile tanımlanır. MATLAB yazılımında **trapmf** olarak gösterilir ve parametreler için [a b c d] gösterimini kullanırız.



Şekil 6. Genel yamuk üyelik fonksiyonu

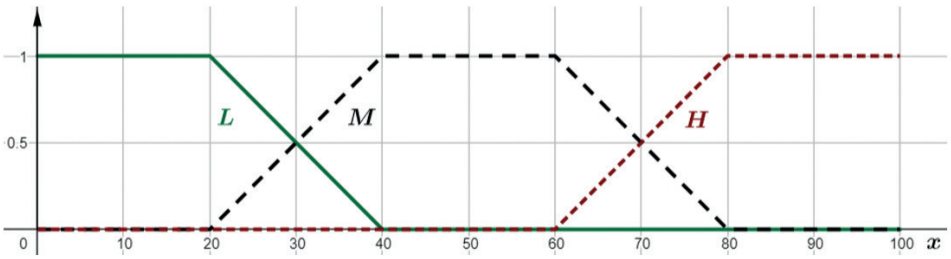
Örnek: Şekil 7'de gösterilen A, B bulanık kümelerinin MATLAB notasyonunu yazın:



Şekil 7. Örnekteki bulanık A ve B kümeleri

Bulanık küme A, üçgen üyelik fonksiyonu kullanılarak temsil edilir. MATLAB programındaki gösterimi A [6 7 8] şeklindedir. Bulanık küme B ise, dörtgen üyelik fonksiyonu kullanılarak temsil edilir ve MATLAB programındaki gösterimi B [5 6 8 9] şeklindedir.

Örnek: Gerçek hayatta bazı gözlemlenen fenomenler için sıkça "düşük," "orta" ve "yüksek" terimlerini kullanırız. Su sıcaklığı için, düşük sıcaklık, orta sıcaklık ve yüksek sıcaklık gibi terimler kullanabiliriz (Şekil 8'e bakınız). Orta sıcaklık" (fonksiyon M) terimi, önceki metinde belirtildiği gibi bir dörtgen üyelik fonksiyonu ile tanımlanabilirken, "düşük sıcaklık" (fonksiyon L) ve "yüksek sıcaklık" (fonksiyon H) değerleri, asimetrik üyelik fonksiyonları kullanılarak tanımlanması gereken özgün terimlerdir. Bu fonksiyonların reçetesini MATLAB programında nasıl yazabiliriz?



Şekil 8. Örnekteki bulanık L, M ve H kümeleri

MATLAB yazılımındaki bazı bulanık kümeleri tanımlamak için, aynı zamanda aşağıdaki gibi incelenen değişkenin evrenine ait olmayan parametreleri de kullanabiliriz. Yani ilk adımda "su sıcaklığı" teriminin evrenini belirlememiz gerekiyor. bu 'dür. Böyle yazabiliriz:

$$L\ [-20\ -\ 10\ 20\ 40],\quad M\ [20\ 40\ 60\ 80],\quad H\ [60\ 80\ 110\ 120].$$

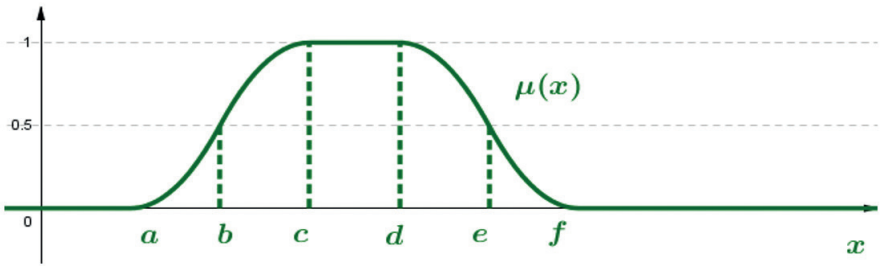
Polinom tabanlı üyelik fonksiyonları

Bu tür fonksiyonlar fonksiyon polinom (ikinci dereceden) kullanılarak oluşturulur. Üç temel gruba ayrılırlar:

- Pi eğrisi,
- S eğrisi,
- Z eğrisi.

Pi tipinin üyelik fonksiyonu

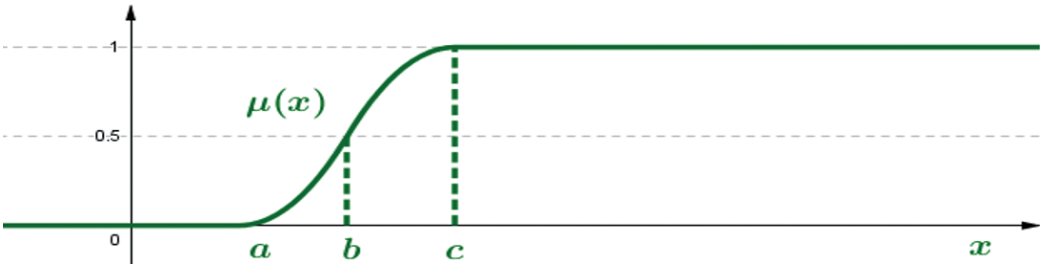
Pi tipi üyelik fonksiyonu altı parametre *a,b,c,d,e,f* ile tanımlanır (bkz. Şekil 9). İkinci dereceden fonksiyonlarla tanımlanan dört parçası vardır (aralıkları ⟨a, b⟩ ; ⟨b, c⟩; ⟨d, e⟩; ⟨e, f⟩), 0 değerli iki kısım (aralıklar (- ∞, a⟩; (f, ∞)) her giriş değerine atanır ve 1 değerli kısım da (aralık ⟨c, d⟩) her giriş değerine atanır. MATLAB yazılımında **pimf** olarak ifade edilir.



Şekil 9. Pi tipi üyelik fonksiyonu

S Tipi Apartman İşlevleri

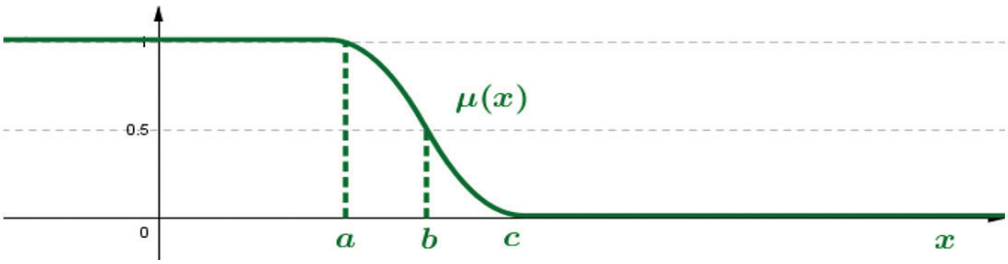
S tipinin üyelik fonksiyonu *a,b,c* olmak üzere üç parametre ile tanımlanır (bkz. Şekil 10). İkinci dereceden fonksiyonlarla tanımlanan iki kısmı vardır (aralıklar ⟨a,b⟩; ⟨b,c⟩), bir kısmı 0 değerindedir (aralıklar (-∞, a⟩) her giriş değerine atanır ve her bir giriş değerine 1 değerinin (aralık ⟨c, ∞) atandığı kısımdır. MATLAB yazılımında smf olarak belirtilmektedir.



Şekil 10. Türün üyelik işlevi

Z Tipi Apartman İşlevleri

Tip Ayırma İşlevi Tanımlı 3 parametre *a,b,c*'dir (şekil 44'te). Aynı zamanda, belirli bir işlemsel işleve (aralık ⟨a,b⟩; ⟨b,c⟩) veya değer alanı 1'e (aralık (-∞, a⟩) ait olan) intrarei değeri atfedilir. ve parçanın değeri 0'dır (aralık ⟨c, ∞)) bu iç değere atfedilir. MATLAB yazılımında **zmf** notu bulunmaktadır.



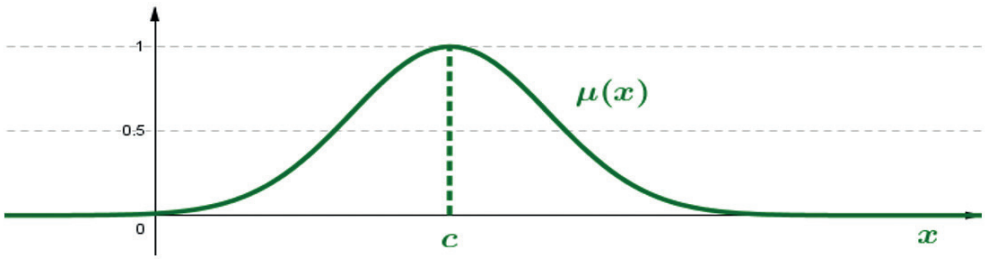
Şekil 11. Z tipi üyelik fonksiyonu

Açıklama:

S ve Z tipi üyelik fonksiyonları asimetrik üyelik fonksiyonlarını temsil eder. Onlar değişkenlerin düşük ve yüksek değerlerinin modellenmesinde kullanılabilir.

İstatistiksel temele dayalı fonksiyon bazlı

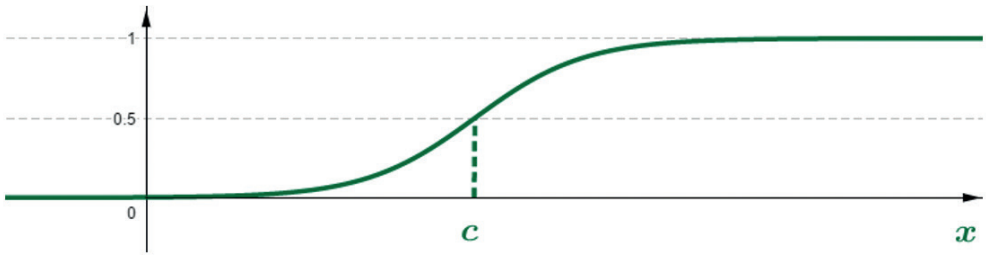
Gauss üyelik fonksiyonları (gaussmf), iki parametresi c,σ olan (bkz. Şekil 12) klasik Gauss dağılım eğrisinden türetilir; burada c, ortalamayı ve σ, verilerin standart sapmasını temsil eder.



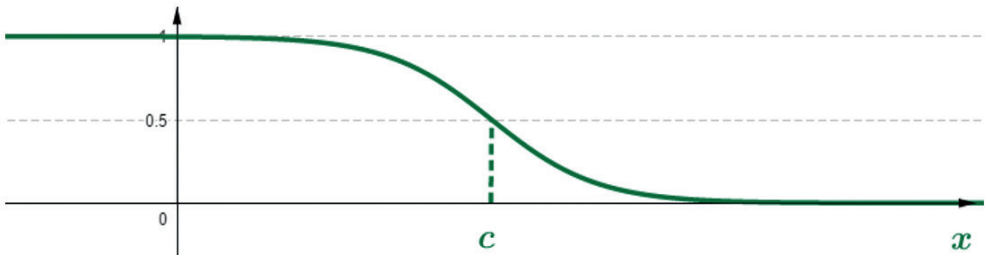
Şekil 12. Gauss üyelik fonksiyonları

Sigmoidal üyelik fonksiyonları

Gauss üyelik fonksiyonları **asimetrik üyelik fonksiyonlarını belirleyemez.** Bu nedenle a, c olmak üzere iki parametreli **sigmoidal üyelik fonksiyonları** (sigmf) kullanılır (bkz. Şekil 13 ve Şekil 14). Daha sonra a, c parametreleri yine istatistiksel bir yaklaşım kullanılarak elde edilir.



Şekil 13. a > 0 olduğunda sigmoidal üyelik fonksiyonları



Şekil 14. a < 0 olduğunda sigmoidal üyelik fonksiyonları

Notlar:

Üçgen ve yamuk üyelik fonksiyonlarını kullanacağız. Gerçek hayattaki uygulamalarda, Gauss ve sigmoidal üyelik fonksiyonları da sıklıkla kullanılır ve bunların parametreleri istatistiksel veri analizi kullanılarak seçilir.

Gerçek hayatta genellikle öncelikle **uzmandan sorunu uygun işlevlerle** tanımlamasını isteriz. Daha sonra, **ikinci adımda**, **genellikle büyük veri grubunun (istatistiksel) işlenmesini** kullanarak fonksiyonların parametrelerini belirleriz.

Bulanık kümelerle çalışmak için bulanık kümelerdeki temel işlemleri (**kesişim, birleşim ve tümleyen)** tanımlamamız gerekir. Benzer şekilde, çok sayıda üyelik fonksiyonu mevcut olduğundan, bulanık kümeler üzerinde **çeşitli işlem türleri de tanımlanmıştır**. Profesör Zadeh'in önerdiği **bulanık kümeler üzerinde standart işlemler** olarak adlandırılan işlemlerden bahsedeceğiz.

Tanım (standart kesişim)

Evren $\mathbb{XXX}$ ve A, B bulanık kümeler olsun. **İki noktanın standart kesişimi bulanık kümeler** A, B, üyelik fonksiyonuna sahip A ∩ B bulanık kümesidir;

$$\mu_{A\cap B}(x) = \min\big(\mu_A(x), \mu_B(x)\big).$$

Tanım (standart birleşim)

Evren $\mathbb{XXX}$ ve A, B bulanık kümeler olsun. **İki bulanık kümenin standart birleşimi** A, B, üyelik fonksiyonuna sahip A ∪ B bulanık kümesidir

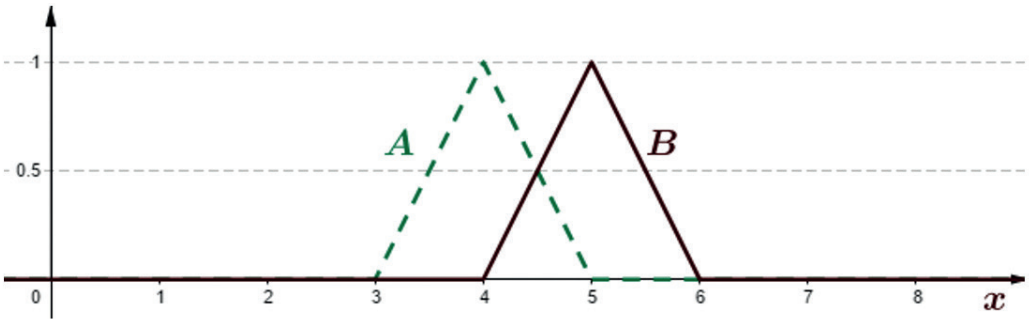
$$\mu_{A\cup B}(x) = \max\big(\mu_A(x), \mu_B(x)\big).$$

Tanım (standart tamamlayıcı)

Evren $\mathbb{XXX}$ ve $\overline{A}$ bulanık kümeler olsun. **Bulanık kümenin standart tamamlayıcısı** *A* üyelik fonksiyonuna sahip A'nın standart tamamlayıcısı bulanık kümesidir.

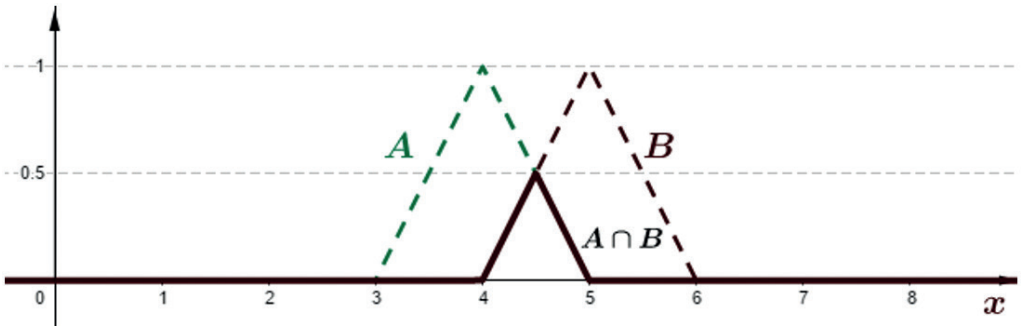
$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

Örnek: Şekil 15'te gösterilen iki A, B bulanık kümesi vardır. Önceki tanımları kullanarak, A, B bulanık kümelerinin kesişimini ve birleşimini ve aynı zamanda A bulanık kümesinin tamamlayıcısını grafiksel olarak belirleyin.



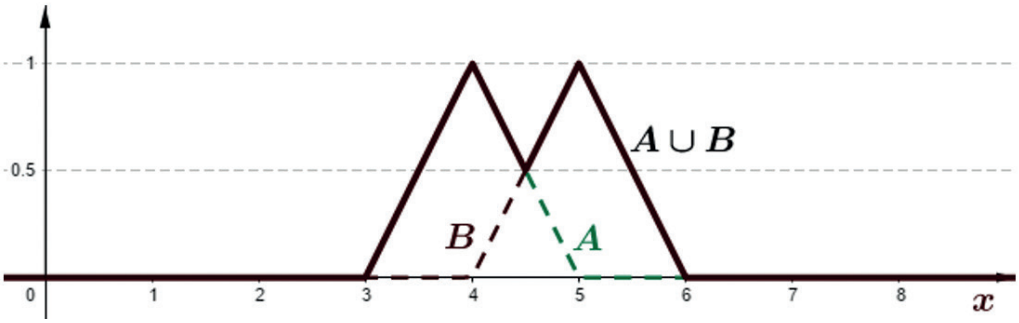
Şekil 15. Örnekteki bulanık A, B kümeleri

İki bulanık küme *A, B*'nin standart kesişimi, üyeliği olan $A \cap B$ bulanık kümesidir. İşlev $\mu_{A\cap B}(x)=\min(\mu_A(x),\mu_B(x))$. Çözüm Şekil 16'da gösterilmektedir.



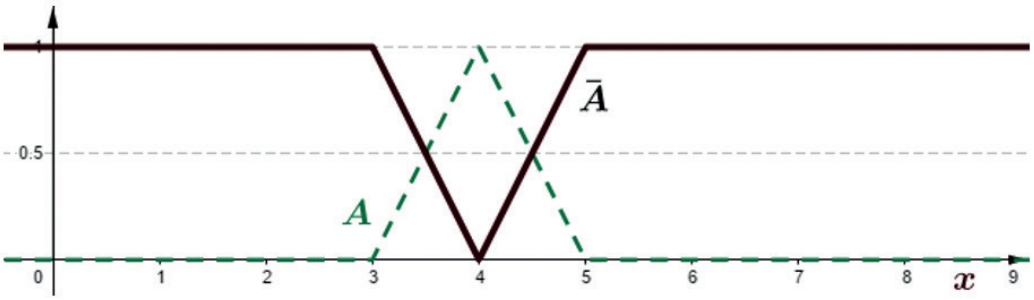
Şekil 16. Örnekteki A, B bulanık kümelerinin standart kesişimi

İki bulanık küme A, B'nin standart birleşimi, üyeliği olan A ∪ B bulanık kümesidir. İşlev $\mu_{A\cup B}(x)=\min(\mu_A(x),\mu_B(x))$. . Çözüm Şekil 17'de gösterilmektedir.



Şekil 17. Örnekteki A, B bulanık kümelerinin standart birleşimi

A bulanık kümesinin standart tamamlayıcısı üyelik fonksiyonu $\mu A(x)=1-\mu A(x)$ olan bulanık kümedir. Çözüm Şekil 18'de gösterilmektedir.



Şekil 18. Örnekte A bulanık kümesinin standart tamamlayıcısı.

BÖLÜM 6

BULANIK AKIL YÜRÜTME

El kitabının bu bölümü Slovakya'nın Banská Bystrica şehrindeki Matej Bel Üniversitesi Doğa Bilimleri Fakültesi Bilgisayar Bilimleri Bölümü'nden Alžbeta Michalíková tarafından yazılmıştır.

Bulanık akıl yürütme, belirsiz terimlerden oluşan bilgilere dayanarak sonuçları çıkardığımız bir süreçtir. Örneğin, gerçek hayatta sıklıkla aşağıdaki gibi kuralları kullanırız:

"Dışarısı soğuksa sıcak tutan kıyafetler giyeceğim"

Bu kuralları gözlemleyerek, öğrenerek, akıl yürüterek vb. yoluyla elde ederiz. Bulanık akıl yürütmede, aşağıdaki forma sahip olan EĞER - O ZAMAN bulanık kurallarını kullanırız:

IF ⟨...⟩	*THEN* ⟨...⟩
= antecedent (premise)	*= cosequent (conclusion)*

İhtiyaçlarımıza göre

"Dışarısı soğuksa sıcak tutacak giysiler giyeceğim."

Kuralını

"Dışarıdaki sıcaklık düşükse, O ZAMAN Elbise sıcaktır."

Formuna dönüştüreceğiz.

"**Sıcaklık**" ve "**Kıyafet**" kelimelerine **dilsel değişkenler** denir. Bu nedenle büyük harfle yazıyoruz. "**Düşük**" ve "**sıcak**" değerlerine **dilsel değişkenlerin değerleri** denir. Önceki kısımda yer alan dilsel değişkenlere **girdi dilsel değişkenler** adı verilmektedir. Sonuç olan dilsel değişkenlere **çıktı dilsel**

değişkenler adı verilir. Bağlaç (VE), ayırma (YA DA) ve olumsuzlama (DEĞİL) işlemlerini kullanarak daha karmaşık kurallar oluşturabiliriz, örneğin:

"Dışarıdaki sıcaklık düşükse VE Bulutluluk yüksekse, O ZAMAN Elbise sıcaktır."

Gerekli tüm kuralları tanımlarsak, **kural tabanı** adı verilen kurallar kümesini elde ederiz. Kural tabanının kurallarıyla çalışmaya yönelik çeşitli yaklaşımlar vardır. Bunlardan biri olan **Sugeno yöntemini** tartışacağız ve kullanacağız.

Sugeno yöntemi

Bu yöntemin yazarları T. Takagi, M. Sugeno ve G. Kang [5]'dir. Bu yöntemi 1985 yılında önerdiler. Bu yöntem, giriş ve çıkış değişkenleri arasındaki bağımlılığı tanımlamanın mümkün olduğu problemlerin modellemesi için tasarlanmıştır. Bu bağımlılık fonksiyonu, non-lineer olan ancak bazı bölümleri lineer olan durumlar için uygundur.

Otopark modelleme probleminde ilk kez Sugeno yöntemi kullanıldı. Günümüzde **sınıflandırma**, **düzenleme** ve **kontrol**, **bulanık karar verme**, uzman sistemlerde **doğrusal olmayan fonksiyonlarla verilere** yaklaşmak için kullanılmaktadır.

Sugeno yönteminde, **giriş değişkenlerinin** değerleri **üyelik fonksiyonları** tarafından tanımlanır. Bu fonksiyonları uzmanlar tasarlar. **Çıkış değişkenleri**, sabit, lineer veya herhangi bir dereceye sahip **polinom fonksiyonları olabilen fonksiyonlarla** tanımlanır.

Sabit çıkış fonksiyonlarına sahip Sugeno kuralları - sabit bir fonksiyon- her kuralın çıkış değişkenini açıklar. Genel olarak kural şu şekildedir:

$$R_j:\ IF\ X_1\ is\ A_{1j}\ \ AND\ \ X_2\ is\ A_{2j}\ \ AND\ \ \dots\ AND\ \ X_n\ is\ A_{nj}, THEN\ Y\ is\ b_j.$$

Sugeno kuralları doğrusal çıktı fonksiyonlarıyla - her kuralın çıktı değişkeni doğrusal fonksiyonla tanımlanır. Genel olarak kural şu şekildedir:

$a_{1j},\dots, a_{nj}, b_j$ doğal sayılarının olduğu yerde,

$$R_j:\ IF\ X_1\ is\ A_{1j}\ \ AND\ \ \dots\ AND\ \ X_n\ is\ A_{nj}, THEN\ Y\ is\ a_{1j}{x_1}^{m_1} + \cdots + a_{nj}{x_n}^{m_n} + b_j,$$

Sugeno, polinom çıkış fonksiyonlarına sahip kurallar - her kuralın çıkış değişkeni, herhangi bir dereceden bir polinom fonksiyonu ile tanımlanır.

$a_{1j},\dots,a_{nj}, b_j$ gerçek sayılarının ve $m_1,\dots,m_n$ doğal sayılarının olduğu yerde

$$R_j:\ IF\ X_1\ is\ A_{1j}\ \ AND\ \ \dots\ AND\ \ X_n\ is\ A_{nj}, THEN\ Y\ is\ a_{1j}{x_1}^{m_1} + \cdots + a_{nj}{x_n}^{m_n} + b_j,$$

Örnek: Sabit çıktı fonksiyonuna sahip Sugeno kuralı

Öğrencileri Sugeno yöntemi kullanarak değerlendirmemiz gerekiyordu. Bu, aşağıdaki türde kurallarla tanımlanabilirdi:

R: EĞER Sunum yüksek VE Test puanı değeri yüksekse,
O ZAMAN Değerlendirme 1'e eşittir (=A).

Örnek: Doğrusal çıktı fonksiyonlarıyla Sugeno kuralları

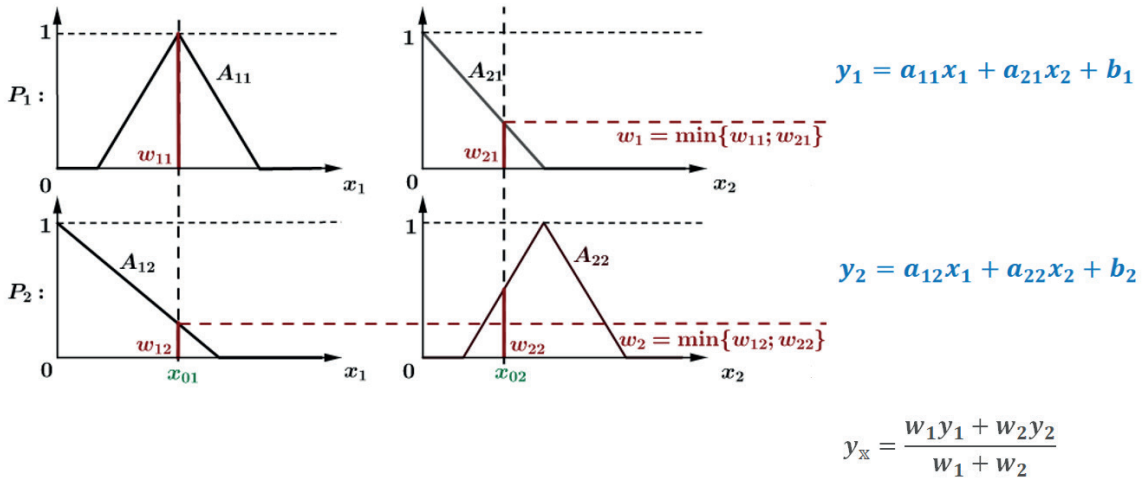
Bazı değerler için (örneğin, küçük değerler için) arabanın pozisyonunun, kolayca belirlenebilen düz bir çizgi reçetesini takip edeceğini biliyoruz. Kuralın aşağıdaki formu vardır:

R: EĞER Pozisyon değeri düşükse, O ZAMAN Çizgi reçetesi 3,25x+2,5'tir.

K kurallı kural tabanına sahip olalım. $\mathbb{X}=(x_1, x_2,\ldots,x_n)$ olsun. Her kuralın şu formda bir çıktı fonksiyonu olsun: $y_j = a_{1j}{x_1}^{m_1} + a_{2j}{x_2}^{m_2} + \cdots + a_{nj}{x_n}^{m_n} + b_j$. Daha sonra nihai çıktı $y_{\mathbb{X}}$ aşağıdaki formülle hesaplanır:

$$y_{\mathbb{X}} = \frac{\sum_{j=1}^{k} w_j y_j}{\sum_{j=1}^{k} w_j}$$

burada, w_j j'inci kuralın ağırlığıdır (bkz. Şekil 18).



Şekil 19. Sugeno yönteminde iki giriş değişkeni ve iki kuralla son çıktı

Ağırlık almanın yolu hakkında açıklama:

η giriş değişkenli R_j kuralımız olsun ($x_1, x_2,\ldots, x_n$). İlk olarak, ölçülen giriş x_i'nin değeri ile ilgili üyelik fonksiyonu A_{1j} arasındaki kesişim olarak wij ağırlıklarını hesaplıyoruz. İkinci olarak, aşağıdaki formülü kullanarak w_j ağırlığını hesaplıyoruz.

$$w_j = \min_i w_{ij} .$$

Bulanık kurallar uygun şekilde nasıl tasarlanır?

- Uzmanlardan bilgilerini uygun işlevleri kullanarak açıklamalarını isteyebiliriz.
- Bilinen çok sayıda veriyi işleyerek fonksiyonların parametre değerlerini belirleyebiliriz.

Bu yöntemin çeşitli isimleri vardır, örneğin **Sugeno yöntemi, Takagi-Sugeno bulanık çıkarım sistemi**, **Takagi-Sugeno regülatörü**,... Bu isimler aynı yöntemi temsil etmektedir. Kullanılan ad, yöntemin kullanıldığı alanla ilgilidir.

Sugeno yönteminin iki farklı alanda kullanımını göstereceğiz

- veri sınıflandırmasında,
- veri yaklaşımında.

İki durumda da, verilen sistemin kurallarını tasarlayacak uzmanlar olacağız [2], [6], [7]. Giriş dil değişkenlerinin değerlerini oluşturmak için en basit üyelik fonksiyonlarını kullanacağız.

Giriş dilsel değişkenlerinin değerlerini oluştururken, en basit üyelik fonksiyonu türlerini kullanacağız. - **Doğrusal üyelik fonksiyonları**: **Çıkış dil değişkenlerinin** değerlerini oluşturmak için **sınıflandırma için sabit fonksiyonlar** ve **yaklaşım için lineer fonksiyonlar** kullanacağız.

BÖLÜM 7

VERİ İÇİN SUGENO YÖNTEMİNİN KULLANILMASI SINIFLANDIRMA

El kitabının bu bölümü Slovakya'nın Banská Bystrica şehrindeki Matej Bel Üniversitesi Doğa Bilimleri Fakültesi Bilgisayar Bilimleri Bölümü'nden Alžbeta Michalíková tarafından yazılmıştır.

El kitabının bu bölümünde **Iris** veri seti ile çalışacağız (bkz. Ek A). **İris veri kümesinin** 150 İris çiçeği örneğinden oluştuğunu hatırlayın. Her çiçek için **dört temel özelliğimiz** vardır: çanak yaprakların uzunluğu ve genişliği ve yaprakların santimetre (veya milimetre) cinsinden uzunluğu ve genişliği. Öte yandan çiçekleri, İris'in üç türüne (**Iris Setosa, Iris Virginica ve Iris Versicolor)** karşılık gelen üç sınıfa ayırabiliriz.

Bu bölümde Excel ve MATLAB yazılımlarındaki veri işlemeyi birleştireceğiz.

Örnek: Sugeno yöntemini kullanarak Iris veri kümesindeki verileri uygun sayıda sınıfa ayırın. (Bu örneğin çözümü Ek B'de bulunabilir.)

Öncelikle şu sorulara cevap vermeye çalışalım:

1. İris veri setinde kaç tane giriş değişkeni var?
2. Giriş değişkenlerini tanımlamak için ne kullanacağız?
3. Ne tür bulanık üyelik fonksiyonlarını kullanacağız?
4. Çıktı ne olacak?
5. Çıkış değişkenlerini tanımlamak için ne kullanacağız?
6. Hangi tür kuralları kullanacağız?
7. Bir kurala örnek yazın!

İkinci olarak, Iris veri kümesini bir web sayfasından indirin ve Excel dosyasına kopyalayın. İlk 50 varlığı **kırmızı** renkle, sonraki 50 varlığı **mavi** renkle ve geri kalanları da **yeşil** renkle **işaretleyelim**. Excel'de dört bağımsız sayfa oluşturun ve renkli tabloyu her birine kopyalayın. İlk sayfada değerleri ilk sütuna göre (en küçükten en büyüğe) sıralayın. Benzer şekilde, her sayfadaki değerleri sütunlardan birine göre sıralayın. Giriş değişkenlerini modellemek için **yamuk fonksiyonlarını** kullanacağız. Bu verilerden giriş değişkeni parametrelerinin değerlerini belirleyin ve bunları aşağıdaki tablolara doldurun.

Tablo1. Giriş değişkenlerinin parametreleri

Input1:

Nume	**Parametrii**
Univers	
Roşu	
Albastru	
Verde	

Input2:

Nume	**Parametrii**
Univers	
Roşu	
Albastru	
Verde	

Input3:

Nume	**Parametrii**
Univers	
Roşu	
Albastru	
Verde	

Input4:

Nume	**Parametrii**
Univers	
Roşu	
Albastru	
Verde	

Üçüncü olarak, çıkış parametrelerinin değerlerini belirleyin. **Çıkış dilsel değişkeni** için **sabit işlevler** kullanıyorsak aşağıdaki tabloyu doğru değerlerle doldurun.

Tablo 2. Çıkış değişkeninin parametreleri

REZULTAT:

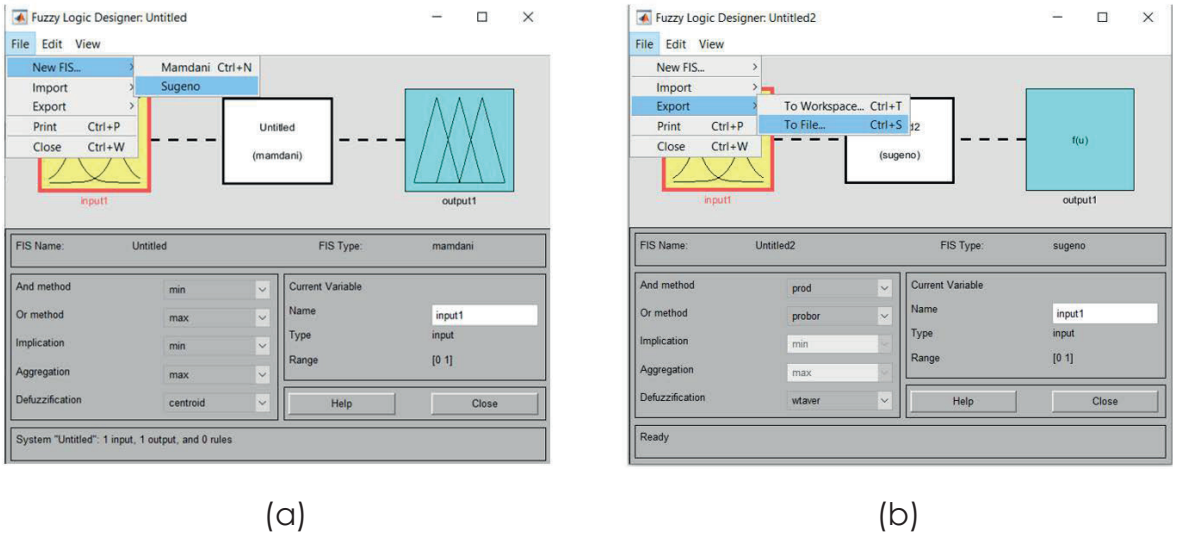
Nume	**Parametrii**
Univers	
Roşu	
Albastru	
Verde	

Dördüncü olarak, kural sayısını önerin ve bunları doğru yazın.

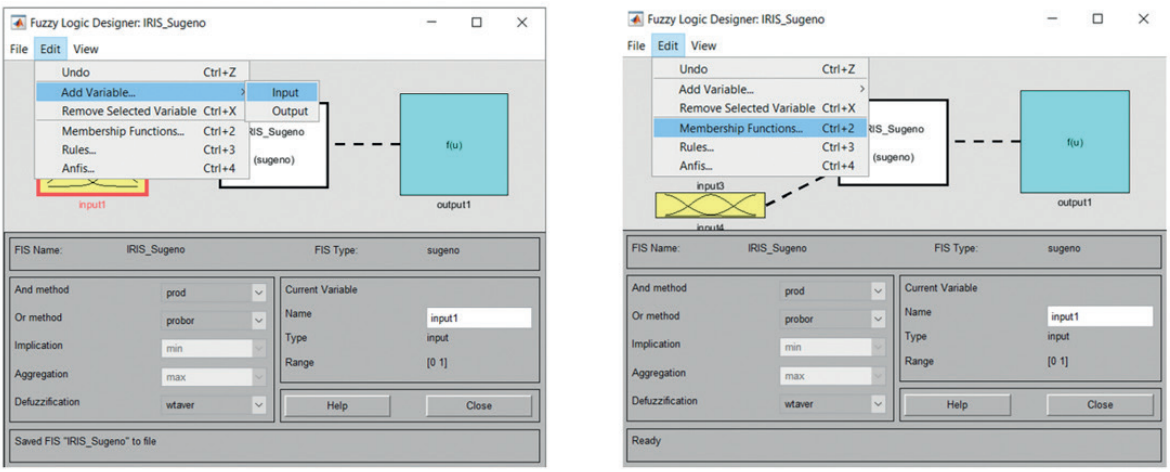
Kurallar:

Şimdi elde ettiğimiz değerleri MATLAB yazılımında işleyeceğiz. MATLAB yazılımını açınız ve **Fuzzy** komutunu Komut Penceresine yazınız. Bu komut Bulanık Mantık Tasarımcısını açar. Sugeno yöntemini kullanacağız (Sugeno bulanık çıkarım sistemi = Sugeno FIS). Bu nedenle bu tip FIS'i açmamız gerekiyor (bkz. Şekil 19a). Bu FIS'i, **IRIS_Sugeno** dosyası olarak yeniden adlandırıp kaydedebiliriz (bkz. Şekil

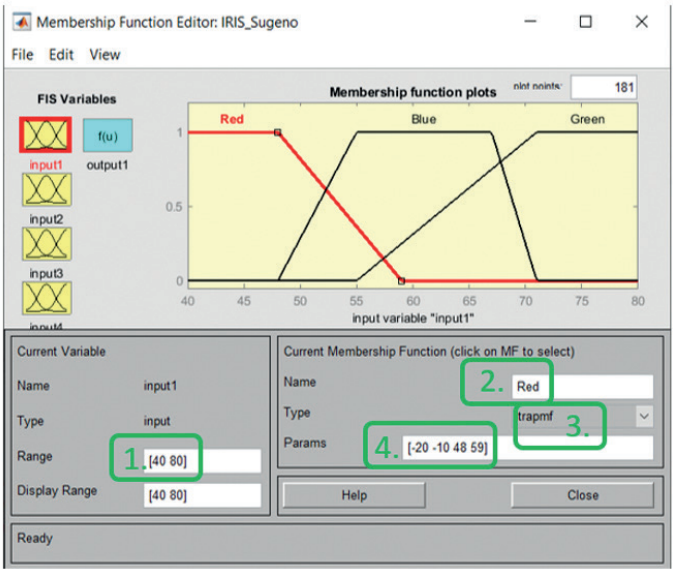
19b). Şimdi dört giriş dilsel değişkenine ihtiyacımız var; üç yeni giriş değişkeni ekliyoruz (bkz. Şekil 20a) ve ardından üyelik fonksiyonlarının parametrelerini düzenlememiz gerekiyor (bkz. Şekil 20b). Şimdi her giriş değişkeni için, değişkenin aralığını adım adım değiştireceğiz, üyelik fonksiyonlarının adını ekleyeceğiz, üyelik fonksiyonlarının türünü değiştireceğiz ve parametreleri her üyelik fonksiyonuna ekleyeceğiz (Tablo 1'i kullanın). Bu adımlar Şekil 21'de gösterilmektedir.



Şekil 20. Yeni Sugeno FIS'in (a) açılması ve FIS'in (b) MATLAB yazılımında yeniden adlandırılması/kaydedilmesi

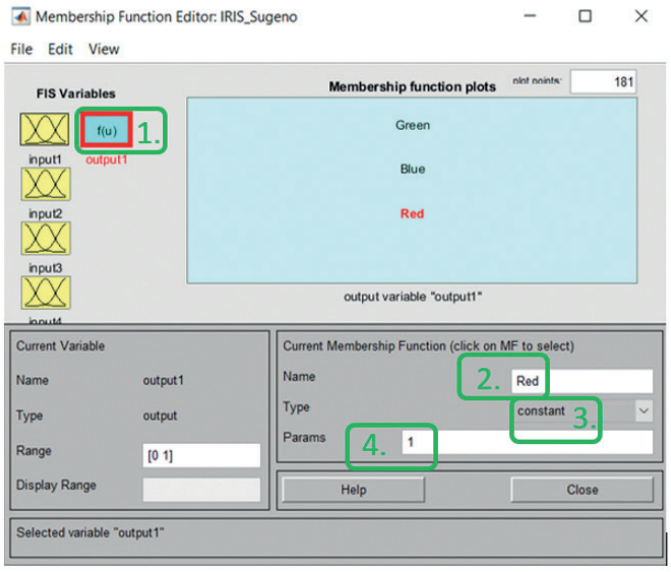


Şekil 21. MATLAB yazılımında yeni değişkenlerin eklenmesi (a) ve üyelik fonksiyonlarının düzenlenmesi (b)



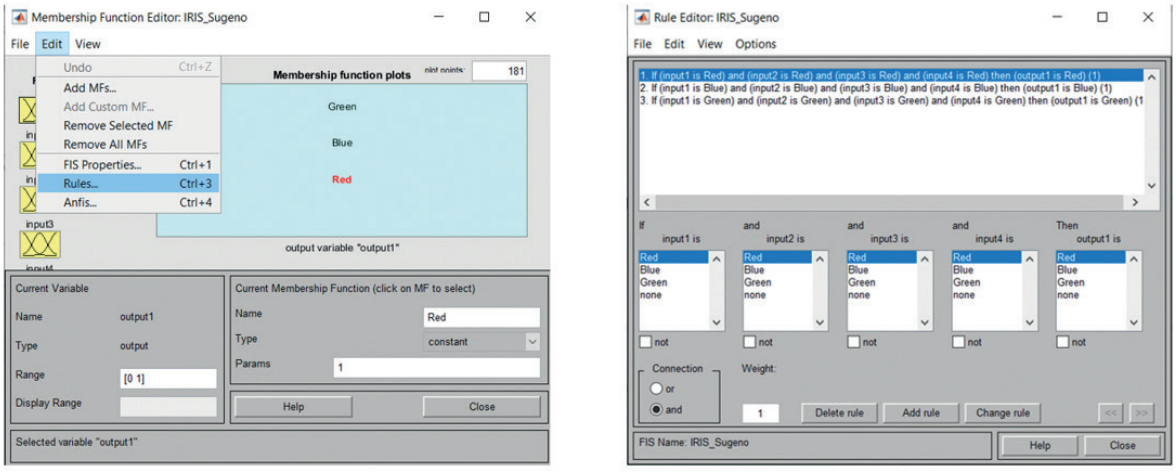
Şekil 22. MATLAB yazılımında giriş üyelik fonksiyonu parametrelerinin değiştirilmesi

Şimdi çıktı değişkeninin değerlerini değiştirmemiz gerekiyor. Çıkış değişkenini düzenlemek için, çıkış1 adlı mavi dikdörtgene çift tıklamayı kullanırız. Şekil 22'de gösterildiği gibi çıkış değişkeni için yeni menüyü elde edeceğiz. Bu menüye Tablo 2'deki değerleri dolduracağız.



Şekil 23. MATLAB yazılımında çıkış değerleri parametrelerinin değiştirilmesi

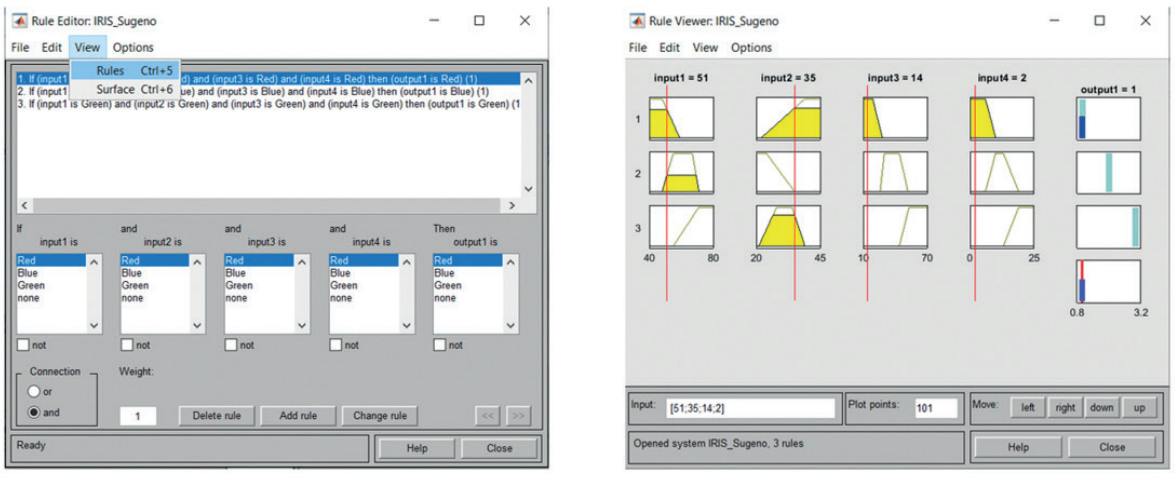
Son adımımız sistemimizin kurallarını oluşturmaktır. Kural menüsünü açıyoruz (bkz. Şekil 23a) ve üç basit kural kullanıyoruz (bkz. Şekil 23b).



(a) (b)

Şekil 24. MATLAB yazılımında kural menüsünün açılması (a) ve kuralların (b) eklenmesi

Sistemimiz hazır. Artık sistemin bilinen girdiler için verdiği sonuçları değerlendirebiliriz. Kural görüntüleyiciyi açabiliriz (bkz. Şekil 24a) ve her giriş değişkenine belirli bir değer ekleyebiliriz (bkz. Şekil 24b). Menünün üst kısmındaki kırmızı çizgileri hareket ettirerek veya menünün alt kısmındaki sıralı dörtlü parametrelerinin değerlerini değiştirerek bu değerleri ekleyebiliriz.



(a) (b)

Şekil 25. MATLAB yazılımında kural görüntüleyiciyi açma (a) ve giriş parametrelerine (b) belirli değerler ekleme

Notlar:

Şekil 24b'de görüntülenen giriş değerleri İris veri tablosunun ilk satırına aittir. Görüldüğü gibi sistem bu giriş niteliklerine sahip nesneyi sınıf 1'e (çıkış1 = 1) sınıflandırmıştır. Sonuç olarak beklediğimiz değer bu!

İris veri kümesinden bir nesneyi nasıl sınıflandırabileceğimizi gösterdik. Bu yaklaşımı tablonun her satırı için kullanabiliriz. Elbette, komut satırından bir dizi komut kullanarak tüm tablo satırlarını tek adımda da sınıflandırabiliriz. Verilen FIS'i kullanarak sınıflandırmanın başarı oranını da hesaplayabiliriz. Ek B'de belirtilen parametreleri kullanarak %94,6667'lik bir başarı oranına ulaştık, yani 150 çiçekten 142'si doğru şekilde sınıflandırıldı.

Sınıflandırmanın başarı oranı birkaç farklı yaklaşımın kullanılmasıyla artırılabilir. Örneğin, giriş dilsel değişkenlerinin daha fazla değerini kullanabilir ve ardından daha fazla kural oluşturabiliriz. Her giriş değişkeni için üç değer kullandık (**kırmızı – mavi – yeşil**). Ayrıca her giriş değişkeninin, **çok_küçük_değer – küçük_değer – orta_değer – yüksek_değer – çok_yüksek_değer** değerlerini temsil eden beş değerini de kullanabiliriz. Daha sonra bu giriş değişkenlerinin değerlerini birleştirerek daha fazla kural oluşturabiliriz. Öte yandan, girdi ve çıktı değişkenlerinin değerlerinin parametrelerini optimize etmek için tasarlanmış diğer yöntemleri de kullanabiliriz. Bunlardan biri, bir Sinir Ağı kullanarak oluşturulan FIS'in parametrelerini optimize eden **ANFIS = Adaptif Nöro-Bulanık Çıkarım Sistemi** olarak adlandırılan sistemdir. Sinir ağlarına ilişkin temel bilgiler bu el kitabının bir sonraki bölümünde sunulmaktadır.

BÖLÜM 8

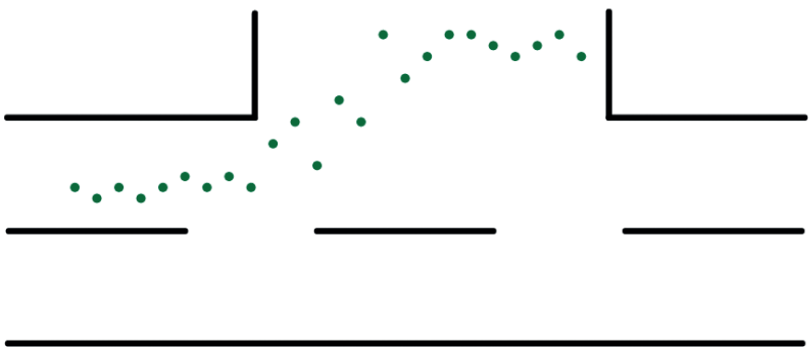
VERİ YAKLAŞTIRMA İÇİN SUGENO YÖNTEMİ KULLANMA

Bu bölüm, Slovakya, Banská Bystrica'daki Matej Bel Üniversitesi Fen Fakültesi Bilgisayar Bilimleri Bölümü'nden Alžbeta Michalíková tarafından yazılmıştır.

Bu durumda, çok miktarda veriye sahibiz ve bunları işlememiz gerekiyor. Sıkça, kesin giriş değerleri için gerçek çıktı yaklaşımını veren daha basit bir fonksiyonla bu verileri yaklaştırmak faydalı olabilir. Bu süreç, **yaklaştırma** olarak adlandırılır. Sugeno yöntemi, bu tür verileri alanın bazı bölümlerinde doğrusal bir şekilde (2D'de bir kısmı doğru parçasıyla yaklaştırılabilir) yaklaştırmak için tasarlanmıştır. Alanın geri kalan kısmında, uygun bir fonksiyonla yaklaştırılması gerekmektedir. Bu metin bölümünde, aracın yolunu temsil eden verileri yaklaştıracağız.

Bu bölümde, veri işleme işlemlerini Excel ve MATLAB yazılımlarında birleştireceğiz.

Örnek: Kendi kendine giden bir araç geliştirdiğimizi hayal edelim. Çözmemiz gereken sorunlardan biri, aracın belirli bir park yerine park etmesini tanımlayacak işlevi bulmaktır. Profesyonel bir sürücüden belirli bir park yerine birkaç kez park etmesini isteyebilir ve aracın yolunu sensörlerle yakalayabiliriz (Şekil 25'e bakınız).



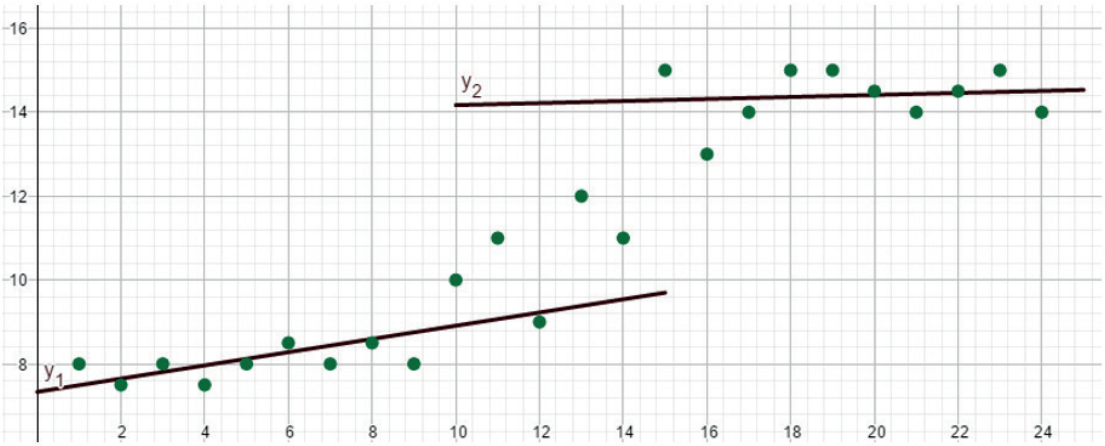
Şekil 26. Araç park etme süreci sırasındaki aracın konumu

Çözüm: Bu araç hareketi, iki kısımda çizgilerle tanımlanabilir. İlk kısım - park öncesinde düz yol üzerindeki hareket. İkinci kısım - park yerindeki hareket. Bu iki kısım arasındaki hareket, Sugeno yöntemiyle yaklaştırılacaktır.

İlk adımda, verilerimizi Kartezzyen koordinat sistemi içine yerleştireceğiz (Tablo 3 ve Şekil 26'ya bakınız). Ayrıca düz hareket çizgilerini çizebilir ve bunlara y1 ve y2 adını verebiliriz. Gördüğümüz gibi, bazı veri noktaları bir çizginin tanımına katkıda bulunacaktır (aralıkları olan x değerine sahip noktalar) ve ayrıca iki çizginin tanımına katkıda bulunacak veriler de olacaktır (aralıkları olan x değerine sahip noktalar). Kullanılan bulanık küme üyelik fonksiyonlarını tasarlarken bu bilgi önemlidir.

Tablo 3. Yaklaştırılmış verilerin koordinatları

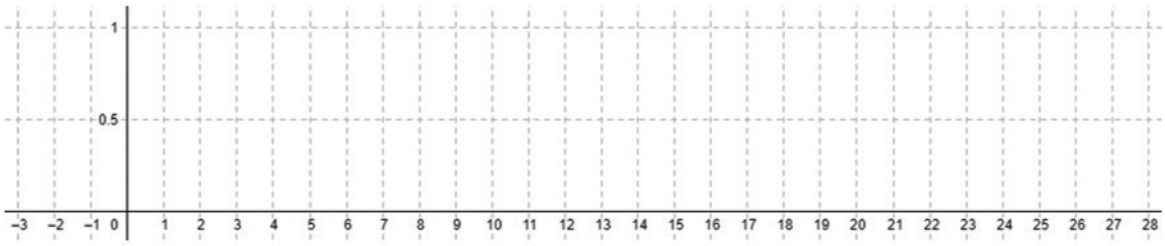
x	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
y	*8*	*7,5*	*8*	*7,5*	*8*	*8,5*	*8*	*8,5*	*8*	*10*	*11*	*9*
x	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*
y	*12*	*11*	*15*	*13*	*14*	*15*	*15*	*15*	*14*	*15*	*15*	*14*



Şekil 27. Verileri Kartezyen koordinat sistemi içine yerleştirme

Aşağıdaki soruları cevaplayalım :

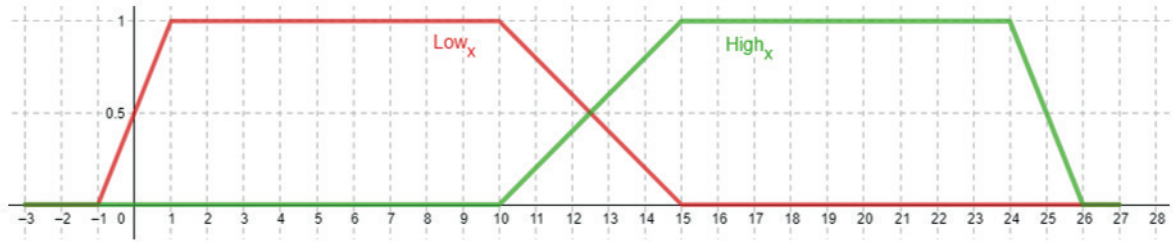
1. Kaç adet **giriş değişkenimiz** var? Bu değişkenlerin adlarını belirtin!
2. Kaç adet **giriş değişkeni değeri** kullanacağız? Bu değişkenlerin değerlerini belirtin!
3. **Giriş değişkenlerini açıklamak** için ne kullanacağız?
4. Hangi **tür bulanık üyelik fonksiyonlarını** kullanacağız?
5. Bu **üyelik fonksiyonlarını çizebilir** misiniz? Aşağıdaki ızgarayı kullanın:



6. Bu fonksiyonların **evrenini** ve **parametrelerini yazabilir** misiniz? MATLAB yazılımı için bu fonksiyonların **tanımını** yazabilir misiniz?
7. **Çıktı** ne olacak?
8. **Çıktı değişkenlerini tanımlamak** için ne kullanacağız?
9. **Kaç kural** kullanacağız?
10. Bir kuralın bir **örneğini** yazın!

Cevaplar :

Tek bir giriş değişkeni var - bu **x-eksenindeki Konum** (arabanın) olarak adlandırılabilir. Bu giriş değişkeninin iki değeri vardır - **x koordinatının Düşük değeri** ve **x koordinatının Yüksek değeri**. Bu değerleri bulanık kümelerle tanımlayacağız. **Trapezoidal üyelik fonksiyonlarını** kullanacağız. Bu fonksiyonlar, Şekil 27'de görüldüğü gibi çizilebilir.)



Şekil 28. Veri yaklaştırması için trapezoidal üyelik fonksiyonları

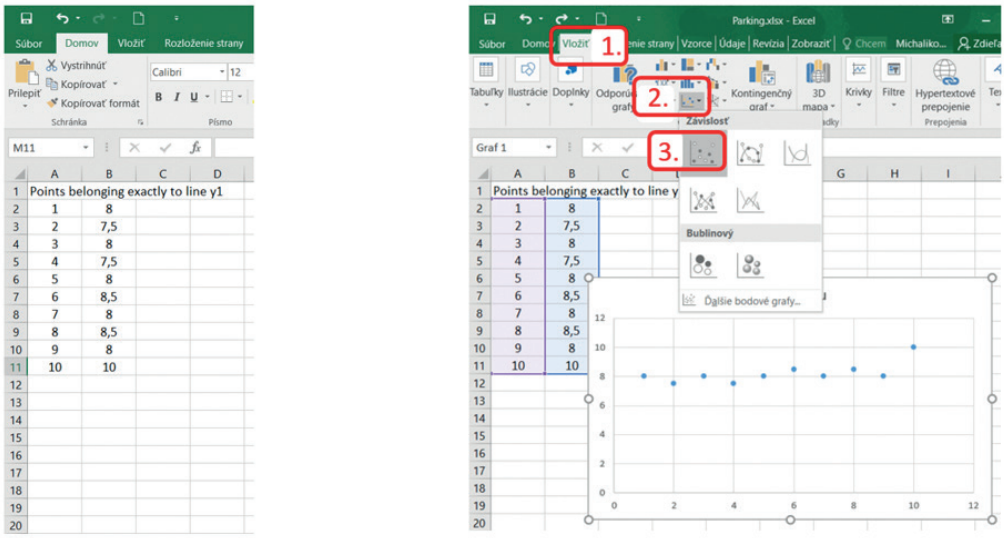
Bu bulanık fonksiyonların evreni x koordinatının Düşük değeri için elimizde parametrelerimiz var. Koordinat x'in Yüksek değeri için elimizde parametrelerimiz var.

Çıktı değişkeni, **y eksenindeki Arabanın Pozisyonunu** temsil eder. Çıktı olarak, bir doğrusal fonksiyon (çizgi) kullanacağız. İki çizgi- *y*1 ve *y*2 kullanacağız. Bu çizgilerin parametrelerini açıklamak için **Excel yazılımını** kullanacağız (aşağıda). Ardından, aşağıdaki gibi yazılan iki EĞER-O ZAMAN kuralı olacaktır:

R1: EĞER aracın x ekseni üzerindeki konumu Düşük x ise, O ZAMAN aracın y ekseni üzerindeki konumu $\mathbf{y}_1$*'dir.*

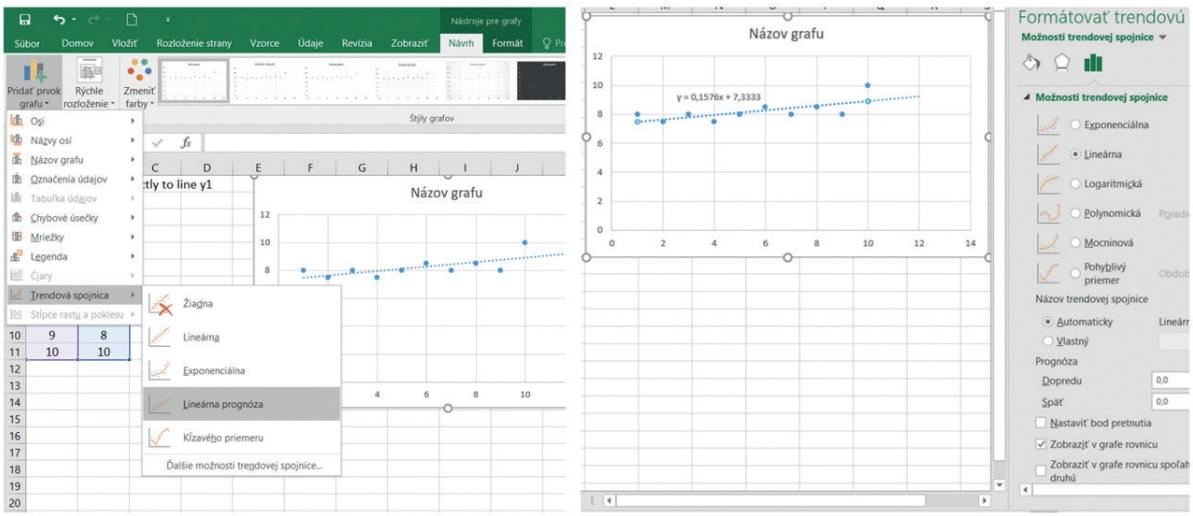
R2: EĞER aracın x ekseni üzerindeki konumu Yüksek x ise, O ZAMAN aracın y ekseni üzerindeki konumu $\mathbf{y}_2$*'dir.*

Lineer fonksiyonların, kuralların çıktısını temsil eden parametrelerini hesaplamamız gerekiyor. Bu parametreler, tam olarak bir çizginin tanımına katkıda bulunan veri değerlerinden hesaplanacaktır; yani, *y*1'in tanımına katkıda bulunan ilk 10 veri noktasının tanımına. Bu verileri Excel'e kopyalayalım - her bir noktayı bir satıra koyun (Şekil 28a'ya bakınız). Ardından bu noktaları işaretleyin ve aşağıdaki adımları kullanarak işlem yapın**: Ekle → Grafikler → Noktalar**



Şekil 29. Excel programında giriş verilerinin işlenmesi

Daha sonra Şekil 29a'da gösterildiği gibi grafiğin elemanını ekleyin ve doğrunun denklemini görüntüleyin (bkz. Şekil 29b).



Şekil 30. Excel yazılımında doğrunun denklemi

y_1 çizgisinin parametrelerini elde ettik. Aynı prosedürü kullanarak y_2 çizgisinin parametrelerini alacağız. O zaman y_1=0,1576x + 7,3333 a y_2= 0,0242x + 13,927 olur. MATLAB'da y_1 [0,1576 7,3333] ve y_2 [0,0242 13,927] kullanılır.

Tablo 4: Veri yaklaşımı için giriş ve çıkış parametrelerinin özeti

INPUT:

Name	Parameters
Universe	[1, 24]
Low x	[-1, 1, 10, 15]
High x	[10, 15, 24, 26]

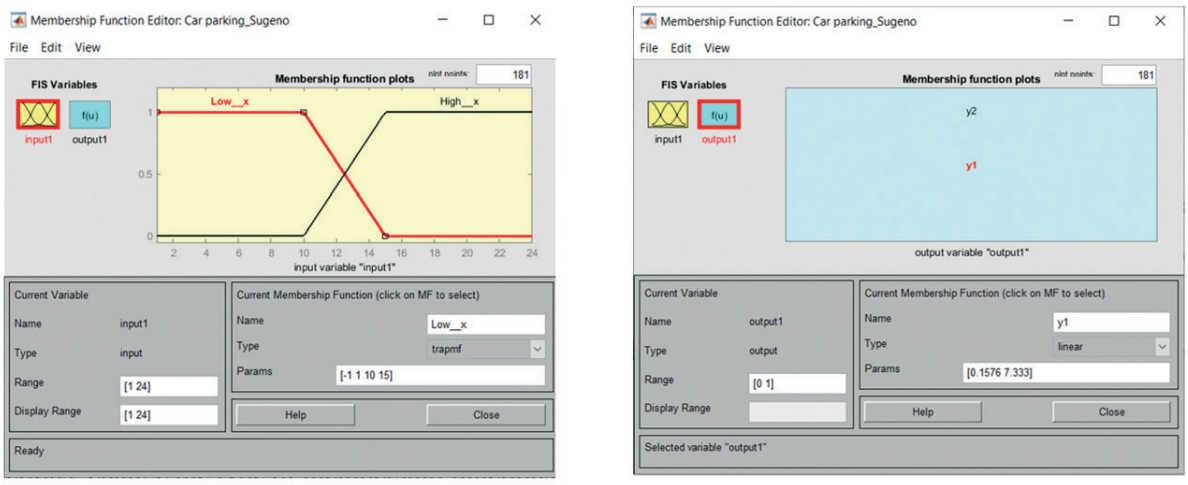
OUTPUT:

Name	Parameters
Universe	---
y_1	[0,1576 7,3333]
y_2	[0,0242 13,927]

Şimdi tüm parametrelere sahibiz ve MATLAB programında Sugeno tipi FIS'yi oluşturabiliriz. İlk adımlar, 3. Bölümde (veri sınıflandırması) belirtilenlere benzerdir. MATLAB'ı açın ve Komut Penceresine **bulanık (fuzzy)** komutunu yazın. Bu komut Bulanık Mantık Tasarımcısını açar. Sugeno yöntemini kullanacağız (Sugeno bulanık çıkarım sistemi = Sugeno FIS). Bu nedenle, bu tür bir FIS'yi açmamız gerekir (Şekil 19a'ya bakınız). Örneğin, bu FIS'i yeniden adlandırabilir ve **Parking_Sugeno** dosyası olarak kaydedebiliriz (Şekil 19b'ya bakınız).

Sadece bir **dilsel giriş değişkenimiz** var. Bu değişken için sadece **iki üyelik fonksiyonumuz** var. Bunlardan birini kaldırmak için grafikteki fonksiyonlardan birine tıklayın ve klavyede Sil'i kullanın. Ardından üyelik fonksiyonlarının parametrelerini düzenleyin (Tablo 4'ten değerleri kullanın). Giriş üyelik fonksiyonları için son yapılandırma Şekil 30a'da gösterilmektedir.

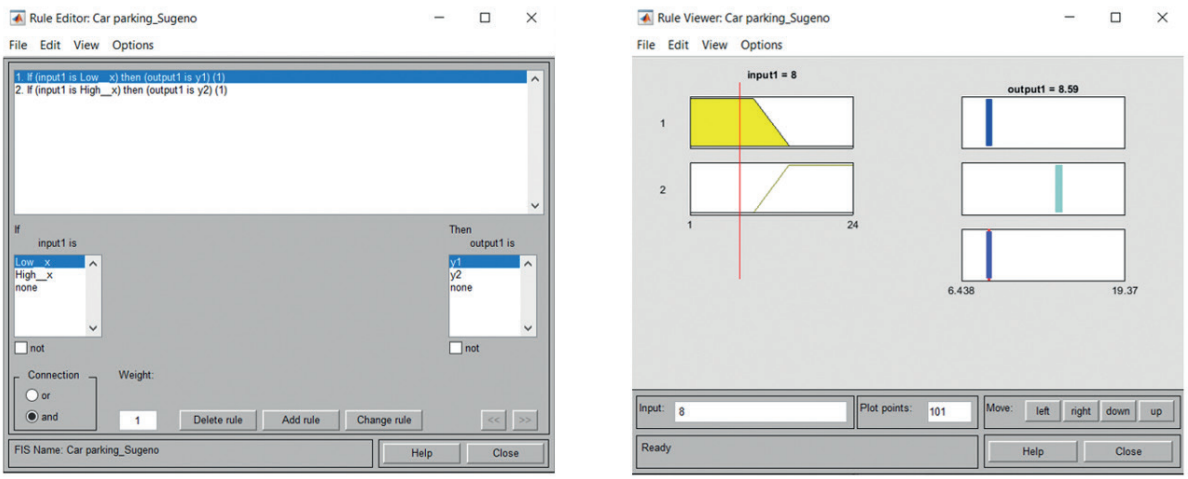
Şimdi çıktı değişkeninin değerlerini değiştirmemiz gerekiyor. Çıktı değişkenini düzenlemek için yine çıktı1 (output1) adlı mavi dikdörtgeni çift tıklayın. Çıktı değişkeni için bir menü alacaksınız. Önceki FIS oluşturulduğunda olduğu gibi tüm değerleri (Tablo 4'ten) dolduruyoruz. Unutmayın ki bu FIS'te **çıktı fonksiyonunun türü doğrusaldır** (Şekil 30b'ye bakınız).



(a) (b)

Şekil 31. MATLAB programında girdi ve çıktı değişkenlerinin yapılandırılması

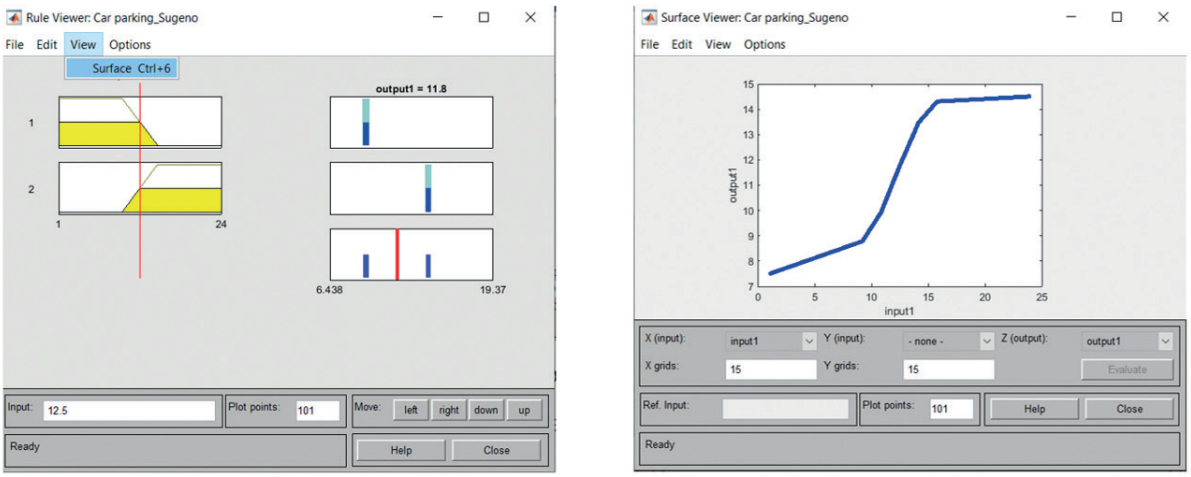
Son adımımız sistemimizin kurallarını oluşturmaktır. İki basit kural kullanacağız (Şekil 31a'ya bakınız). Sistemimiz hazır. Şimdi sistemin bilinen girişler için verdiği sonuçları değerlendirebiliriz. Kural görüntüleyiciyi açabilir ve giriş değişkenine belirli bir değer ekleyebiliriz (Şekil 31b'ye bakınız).



(a) (b)

Şekil 32. MATLAB yazılımında kuralların yapılandırılması ve sonuçların değerlendirilmesi

Bu FIS'yi kullanarak oluşturduğumuz fonksiyonu gösterme imkanı da vardır (Şekil 32a'ya bakınız).



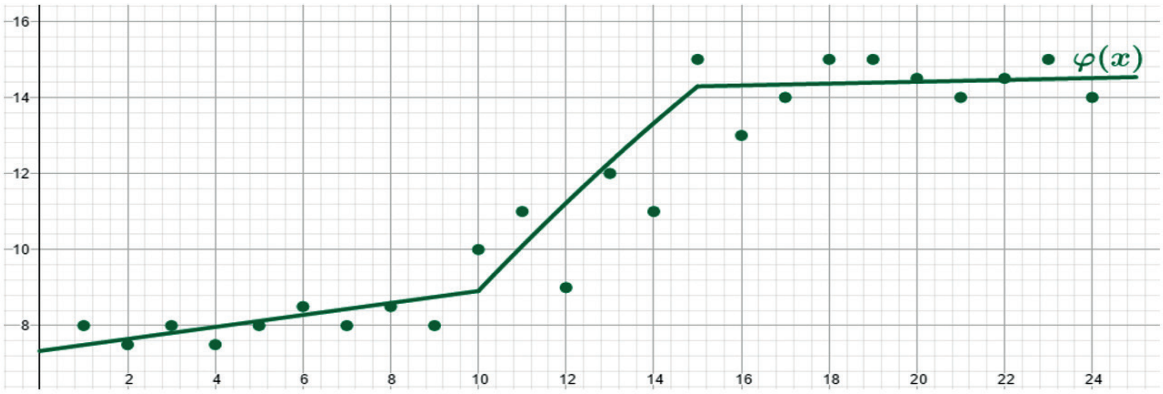
(a) (b)

Şekil 33. MATLAB yazılımında Yüzey Görüntüleyicisini açma ve oluşturulan FIS'nin son işlevi

Notlar:

Şekil 31b'de görüntülenen giriş değeri 8'dir. Gördüğümüz gibi, sistem bu giriş değerine 8.59 çıktı verdi. Gerçek değer (Tablo 3'e bakınız) 8.5 idi. Bu nedenle, elde edilen değer bu nokta için iyi bir yaklaşımı temsil eder.

Orijinal (gerçek) verileri elde edilen fonksiyonla karşılaştırabiliriz. Sonuçları karşılaştırmak (değerlendirmek) için iki temel yaklaşım vardır. Bunlardan ilki grafikseldir. İkincisi, oluşturulan sistemin hatasını hesaplamaktır. Şekil 33, gerçek veri ve elde edilen fonksiyonun **grafiksel karşılaştırmasını** göstermektedir.



Şekil 34. Gerçek verilerin ve elde edilen fonksiyonun grafik karşılaştırması

Bir spesifik giriş değerine yaklaşık çıkış değerini nasıl elde edeceğimizi gösterdik. Elbette, komut satırından bir komut dizisi kullanarak Tablo 3'ün tüm girişlerini tek adımda yaklaşık olarak elde edebiliriz. Elde edilen hatayı da hesaplayabiliriz. Daha fazla hata türü hesaplanabilir. Sözde Ortalama Kare Hatası (MSE)'dir en çok kullanılan hatadır. Bu hata şu formulle hesaplanır :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}[f(x_i) - \varphi(x_i)]^2 ,$$

n, giriş verisi sayısını temsil eder; f(xi) değerleri, gerçek çıktıları temsil eder ve φ(xi) değerleri, FIS tarafından hesaplanan çıktıları temsil eder. Bu MSE değeri, iki veya daha fazla farklı yaklaşımı karşılaştırmak istediğimizde kullanmak uygundur. O zaman en küçük değer, daha iyi yaklaşımı temsil eder. Sistemimiz için, MSE değerinin 0,7263'e ulaştığını gördük.

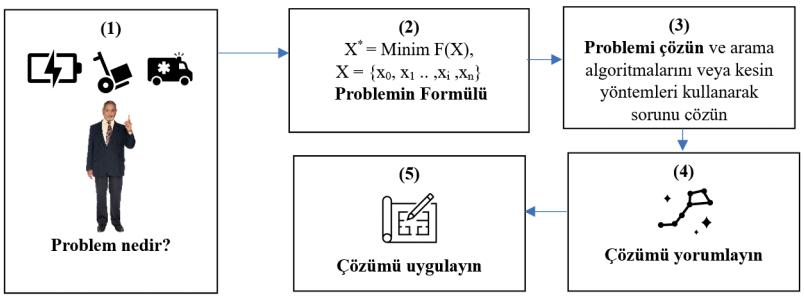
Yaklaşım kalitesi, birkaç farklı yaklaşımın kullanılmasıyla iyileştirilebilir. Örneğin, daha fazla üyelik işlevi kullanabilir ve daha fazla kural oluşturabiliriz. Sunulan örnekte, iki üyelik işlevi **(Low_x - High_x)** kullandık.**Low_x - Medium_x - High_x** değerlerini temsil eden üç giriş değişkeni değeri kullanabiliriz Ardından, 3 çizginin talimatını bulabilir ve giriş ve çıkış değerlerini birleştirerek üç kural oluşturabiliriz. Büyük bir giriş kümesi olduğunda, başka bir tür üyelik işlevi de kullanabiliriz (bu bölümde alt bölüm 2'de bahsedilmiştir). Ardından, verilerin istatistiksel dağılımı, fonksiyonların parametrelerini belirleyebilir. Öte yandan, yine giriş değerleri parametrelerini ve çıkış değişkenlerini optimize etmek için tasarlanmış başka bir yöntemi kullanabiliriz.

BÖLÜM 9

OPTİMİZASYONA GİRİŞ

Bu el kitabının bu bölümü, Türkiye'deki Adana Bilim ve Teknoloji Üniversitesi'nden Fatih Kılıç tarafından yazılmıştır.

Birçok çalışma alanında, optimizasyon problemleri matematiksel, sezgisel ve meta-sezgisel yöntemler kullanılarak arama alanının (tüm uygulanabilir çözümler) en iyi çözümünü bulmak için ele alınabilir. Mühendislik, finans, tıbbi ve üretim problemleri gibi farklı optimizasyon problemleri vardır (Mirjalili, 2016, Cui ve ark., 2017, Kilic ve ark., 2021, Aktaş ve ark., 2022). Şekil 1, optimizasyon problemlerini çözmenin ana adımlarını göstermektedir. İlk adımda, karar vericiler bir optimizasyon problemini çözmek ister, böylece mevcut sistemleri iyileştirebilir veya yeni sistemler önerebilirler. Örneğin, hastaneler için potansiyel hastaları ve talebi dikkate alarak en iyi konumları belirleyebiliriz. İkincisi, bu problem matematiksel olarak çözüm yapısı, amaç ve eşitlikler olarak formüle edilmelidir. Çözüm yapısı karar değişkenlerinden oluşur. Karar değişkenleri, bu problem için aday hastanelerin olası konumlarıdır. Amaç fonksiyonu, aday çözümler arasında değerlendirmek için bir çözümün kalitesini ölçer. Amaç fonksiyonu, örnek problem için hastaneler ile potansiyel hastalar arasındaki mesafelerin toplamı olabilir. Tüm çözümler, önceden tanımlanmış eşitlikler nedeniyle uygulanabilir veya uygulanamaz çözümler olabilir. Bu eşitlikler uzmanlar tarafından tanımlanır. Bu problem için, en az bir hastane paydaşlarca talep edilen bir alt bölgede olabilir. Üçüncü adımda, iyi çözümler bulmak için iyi bilinen yöntemler uygulanır. Bu yöntemler, optimal bir çözüm veya optimal çözüme yakın iyi çözümler üretir. Paydaşlar çözümleri yorumlar ve gerekirse çözümde herhangi bir küçük revizyon yapar. Son olarak, çözüm uyarlanır.



Şekil 1. Optimizasyon problemlerini çözmenin ana adımları

Matematiksel olarak, herhangi bir optimizasyon aşağıdaki şekilde açıklanabilir:

$$\max/\min_{x \in F \subseteq \mathbf{S}} f(x), \qquad \text{Eq. (1)}$$

burada x, karar değişkenlerinin bir kümesini gösterir, F uygun çözümleri içerir, S çözüm uzayını temsil eder ve f(x), amaç fonksiyonunu gösterir. max/min, f(x) fonksiyonunun maksimum ve minimum değerlerini bulma amacını taşır. Kısıtlamaları, veri ile değişkenlerin bir aralığını formüle edebiliriz. Bir örnek aşağıda verilmiştir:

$$\sum_{j}^{n} x_j < \mathrm{b} \qquad \text{Eq. (2)}$$

$$x_j \in \{0,1\} for\ j = 1 \ldots . n \qquad \text{Eq. (3)}$$

Burada, x_j tüm j için 0 veya 1 olabilir ve tüm x öğelerinin toplamı 2. ve 3. denklemlere göre b'den küçüktür.

Sürekli değişkenler bulmaya çalışan problemler, sürekli optimizasyon problemleri olarak sınıflandırılır. Tablo 1, iyi bilinen sürekli optimizasyon problemlerini göstermektedir. Çözüm (X), D boyutlu gerçek değerlerden oluşur. Her boyut, önceden tanımlanmış minimum ve maksimum sayılar arasındadır. Boyut, karar değişkenlerinin sayısıdır. Bunlar, optimizasyon algoritmaları tanıtıldığında performanslarını göstermek için kullanılır.

Tablo 1. Tek Tepeli Fonksiyonlar (Li ve ark., 2013, Hayyolalam, 2020, Wang ve ark., 2022).

Boyut	Alt-üst sınır	Denklem
5	[-100, 100]	$F_1(x) = \sum_{i=1}^{n} x_i^2$
	[-10, 10]	$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$
	[-100, 100]	$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$
	[-100, 100]	$F_4(x) = \max_i \{ \lvert x_i \rvert,\ 1 \le i \le n \}$
	[-30, 30]	$F_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$
	[-100, 100]	$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$
	[-1.28, 1.28]	$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \mathrm{random}\,[0, 1]$

9.1. YEREL ARAMA ALGORİTMALARI

Yerel arama algoritmaları (YAA), bilgisayar bilimi ve ilgili hesaplama bilimlerinde optimizasyon problemlerini çözmek için kullanılır (Kiliç & Gök, 2013, Hoos & Stützle, 2004). Bu algoritmalar, skalar fonksiyonların genel optimizasyonunu çözmeye çalışırlar (Rossi, 2006). Ancak, optimizasyon problemleri formüle edildikten sonra farklı optimizasyon problemleri için uygulanabilirler.

Genellikle YAA'lar, herhangi bir zamanda daha iyi bir çözüm üretmek için tek bir çözümle ilgilenir. Benzetimli Tavlama (Bertsimas & Tsitsiklis, 1993), tabu arama (Glover & Laguna, 1998), tepe tırmanma ve değişken komşuluk araması (Hansen & Mladenovic, 1999) iyi bilinen yerel arama algoritmalarıdır.

Algoritma 1: tepe tırmanma (TT) algoritmasının ana adımlarını göstermektedir.

1	**Mevcut çözüm** = Başlangıç çözümü oluştur
2	**Mevcut çözümü** değerlendir
3	**iterasyon** =0
4	**koşul** (!Durdurma koşulları)
5	**Komşu çözüm** = Hareket (mevcut **çözüm**)
6	Eğer Komşu çözüm **mevcut çözümden daha iyi ise**
7	**Mevcut çözüm =komşu çözüm**
8	**Son eğer**
9	**iterasyon = iterasyon+1**

İlk adımda, başlangıç çözümü rastgele oluşturulur ve mevcut çözüme atılır. Örneğin, Tablo 1'deki F1 fonksiyonu için *X = [60.15, -50.07, 10.08, -80.01, 17.59]* gibi bir vektör oluşturulur. Bu vektörün her bir öğesi *-100 ile +100* arasındadır ve rastgele seçilir. İkinci olarak, her iterasyonda mevcut çözümü ve küçük bir değiştirme işlevini kullanarak komşu çözüm oluşturulur. Rastgele bir indeks numarası seçilir ve X vektörü için seçilen öğe değiştirilir. Eğer komşu çözüm mevcut çözümden daha iyi ise, mevcut çözüm komşu çözümle güncellenir. İterasyonlar, mevcut çözümün belirli bir koşulu sağlayana veya maksimum iterasyona ulaşana kadar gerçekleştirilir.

9.1. EVRİMSEL HESAPLAMA

Evrimsel hesaplama (EH), biyolojik evrimi taklit eden ve üreme, rekombinasyon, mutasyon, seçim ve bireylerin hayatta kalması gibi süreçleri içeren popülasyon tabanlı bir optimizasyon algoritmasıdır (De La Iglesia, 2013, Bartz-Beielstein et al., 2014). EH'nin farklı varyasyonları biyolojik evrim süreçlerini kullanarak tanıtılmıştır. Genetik algoritma, John Holland tarafından (1962) geliştirilmiş, Evrim Stratejileri ise Ingo Rechenberg (1965) tarafından geliştirilmiştir.

Algoritma 2: Evrimsel Hesaplama	
1	**Popülasyon** = Rastgele başlangıç çözümleri oluştur
2	iterasyon =0
3	**koşul** (!Durdurma koşulları)
4	**Popülasyon** içindeki her birey için uygunluk değerleri hesaplanır
5	Uygunluk değerlerine dayalı olarak **Popülasyon** içinde ebeveyn seçilir
6	Yavruları oluşturmak için çaprazlama ve mutasyon yapılır
7	**Popülasyon**, yeni yavrular ve onların uygunluk değerlerine göre güncellenir
8	iterasyon = iterasyon+1

Tipik bir Evrimsel Hesaplama (EH) algoritmasının adımları Algoritma 2 tablosunda verilmiştir.

İlk adımlarda, bir popülasyon önceden belirlenmiş bir boyutta rastgele oluşturulur. Bir popülasyon, çözümlerden oluşur. Her birey bir çözümü temsil eder. İkinci adım, tekrarlayıcı işlemlerden oluşur. İkinci adımda, her birey için bir amaç fonksiyonu kullanarak uygunluk değeri hesaplanır. Ebeveynler, uygunluklarına veya farklı tekniklere dayalı olarak seçilir. Çaprazlama ve mutasyon, yeni çözümler oluşturmak için üreme süreçleridir. Bir sonraki nesil için, bireylerin uygunluk değerlerini kullanarak seçim süreci gerçekleştirilir. Bu adımlar, durma koşulları karşılanana kadar tekrarlanır.

Çaprazlama Operatörü

Çaprazlama operatörü, iki seçilen ebeveynin kromozomları arasında bilgi değişimini gerçekleştirerek yeni yavrular oluşturmayı amaçlar (Kilic ve ark.; 2021, Ahmed, 2010). Bu operatör, Evrimsel Hesaplama'da önemli bir sömürü operatörüdür. Farklı genel çaprazlama teknikleri bulunur, bunlar tek noktalı, çoklu noktalı, uniform çaprazlamalar ve probleme özgü çaprazlama teknikleri (kombinatoriyel optimizasyon problemleri için) olarak kenar sıralama,çoklu ebeveynli kısmi eşleştirilmiş çaprazlama ve sıra tabanlı çaprazlama gibi çeşitli teknikleri içerir. Bu operatör, çaprazlama olasılığına göre gerçekleştirilir.

Tablo 2, tek noktalı çaprazlama operatörü örneğini göstermektedir. Ebeveyn 1 ve 2, ilk ve ikinci satırlarda sırasıyla italik ve altı çizili olarak gösterilen iki çözümü temsil eden seçilmiş bireylerdir. Rastgele bir kesim noktası seçilir ve ebeveynler her bir birey için iki parçaya bölünür. Ebeveynlerin ilk kısımları sabit kalır, ikinci kısımları ise yer değiştirilerek Çocuk 1 ve 2 oluşturulur.

Tablo 2. Tek Noktalı Çaprazlama Operatörü örneği

	X_1	X_2	X_3	X_4	X_5
Ebeveyn 1	60.15	-50.07	10.08	-80.01	17.59
Ebeveyn 2	40.22	30.08	20.09	-20.05	60.85
Çocuk 1	60.15	-50.07	20.09	-20.05	60.85
Çocuk 2	40.22	30.08	10.08	-80.01	17.59

Mutasyon Operatörü

Mutasyon operatörü, popülasyon içinde çeşitliliği sağlamak için gerçekleştirilir. Mutasyon operatörü, bir ebeveyne yavruları üretmek için değişiklik yapar. Çözümün rastgele bir konumu seçilir ve mutasyon operatörünü gerçekleştirmek için ilgili gen veya bit değiştirilir. Farklı mutasyon operatörleri bulunur. Bu operatörlerden biri, bireyin çoklukonumunu aynı anda güncelleyen Büyük Ölçekli mutasyondur.

Tablo 3'te bir mutasyon örneği verilmiştir. *X*3 rastgele seçilir, flip-flop yöntemi kullanılır ve *X*3'ün yeni değeri 0 olur.

Tablo 3. Mutasyon operatörü örneği

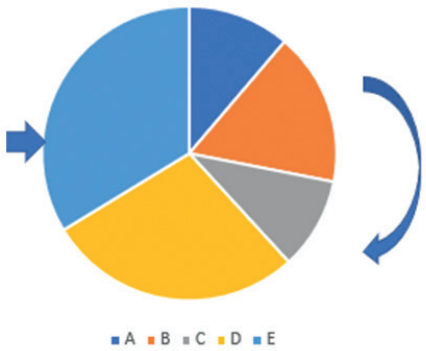
	X_1	X_2	X_3	X_4	X_5
Birey	*1*	*0*	***1***	*0*	*1*
Yeni Birey	*1*	*0*	***0***	*0*	*1*

Seçim Stratejis

Seçim stratejileri, bireylerin ve yavruların bir sonraki nesilde daha yüksek uygunlukla hayatta kalma olasılığını artırmak ve ebeveynleri seçmek için kullanılır (Kılıç ve ark, 2021). Rulet Tekerleği Seçimi ve Turnuva Seçimi popüler seçim stratejileridir.

Rulet Tekerleği Seçimi: Dairesel bir tekerlek, popülasyondaki çözüm sayısı olan n dilimden oluşur. Her çözüm, uygunluk değerine dayalı olarak bir dilim büyüklüğü alır (Sharifi & Aghdam, 2019). Tekerleğin çevresinde bir nokta seçilir ve dairesel tekerlek döndürülür.

Solutions	Fitness Values
A	10
B	15
C	9
D	25
E	30



Şekil 2. Örnek bir popülasyon

Turnuva Seçimi: Bu yaklaşımda *"turnuva, k"* seçimi, popülasyondan rastgele seçilen k bireyden oluşur ve bu bireyler arasından turnuva kullanarak en iyi uygunluğa sahip olanı seçer (Blickle, 2000).

9.3 SIRT ÇANTASI PROBLEMİ ÇÖZÜMÜ

Sırt Çantası Problemi'nde, ağırlıkları ve değerleri olan bir dizi öğe arasından en yüksek toplam değere sahip olanları sırt çantasına koymak istenir (Salkin & De Kluyver, 1975, Gaivoronski et al., 2011). Tablo 4, sırt çantası problemi için bir test kümesini göstermektedir.

Tablo 4. Sırt Çantası Problemi için test veri kümesi.

	Madde 1	Madde 2	Madde 3	Madde 4	Madde 5	Madde 6	Madde 7
Ağırlık	30	20	10	45	15	33	25
Değer	10	5	30	16	50	13	13
Örnek Çözüm	1	0	1	1	0	1	1

Aşağıdaki simgelemleri, parametreleri ve karar değişkenlerini kullanıyoruz.

Simgelem:

j: madde indexi, ∈ {1...J}, *J maddelerin sayısıdır*

Parametreler:

v_j: *j* maddesinin değeri

w_j: *j* maddesinin ağırlığı

W: sırt çantasının maksimum kapasitesi

Karar değişkenleri :

$$x_j = \begin{cases} 1 & \text{eğer j maddesi sırtçantasına seçilmişse} \\ 0, & \text{seçilmemişse} \end{cases}$$

Maksimum bulunacak fonksiyon $$F = \sum_{j=1}^{J} v_j x_j$$

sınırlamalar $$= \sum_{j=1}^{J} w_j x_j < W$$

Uygunluk Fonksiyonu Kodu :

```
function Fit = MyFitness(x)
   global wSet vSet maxCapacity;
   sumV = sum(x(1,:).* vSet);
   sumW = sum(x(1,:).* wSet);
   if sumW <= maxCapacity
     Fit= sumV;
   else
     Fit = 0;
   end
```

Genetik Algoritma Kodu

```
clc;
```

```
clear;
close all;

global nItem wSet vSet maxCapacity;
wSet = [30, 20, 10, 35, 15, 33, 25, 25, 25, 15, 25,54]; % her maddenin ağırlığı
vSet = [10, 5, 30, 16, 50, 13, 13, 23, 14, 52, 10,50]; % her maddenin değeri
maxCapacity = 120;
nItem = size(wSet,2);
FitnessFunction = @(x) MyFitness(x);
WeighFunction = @(x) MyFitnessW(x);
popSize = 20;
maxIter = 50;

muProbility = 0.2;
individual.Solution = [ ];
individual.FitnessValue = [ ];
individual.Weight = [ ];
population = repmat(individual, popSize, 1);
round(rand(1,nItem));
for i = 1:popSize
   population(i).Solution = round(rand(1,nItem));
   population(i).FitnessValue = FitnessFunction(population(i).Solution);
   population(i).Weight = WeighFunction(population(i).Solution);
end

% Popülasyonu sırala
FitnessValues = [population.FitnessValue];
[FitnessValues, SortOrder] = sort(FitnessValues,'descend');
population = population(SortOrder);

BestSol = population(1);
BestFitness = zeros(maxIter, 1);
TournamentSize=3;

for t = 1:maxIter
   % Çaprazlama Operatörü
   populationCrossover = repmat(individual, popSize/2, 2);
   for j = 1:popSize/2
      i1 = TournamentSelection(population, TournamentSize);
      i2 = TournamentSelection(population, TournamentSize);
```

```
      p1 = population(i1);
      p2 = population(i2);
      % Çaprazlama işlemi
      [populationCrossover(j, 1).Solution, populationCrossover(j, 2).Solution] =
      Crossover(p1.Solution, p2.Solution);
   % Yavruları değerlendir
   populationCrossover(j, 1).FitnessValue = FitnessFunction(populationCrossover(j,
   1).Solution);
   populationCrossover(j, 2).FitnessValue = FitnessFunction(populationCrossover(j,
   2).Solution);
   populationCrossover(j, 1).Weight = WeighFunction(populationCrossover(j,
   1).Solution);
   populationCrossover(j, 2).Weight = WeighFunction(populationCrossover(j,
   2).Solution);
end
   populationCrossover = populationCrossover(:);

   % Mutasyon Operatörü
   mutPop =0;
   populationMutation = repmat(individual, 0,1);
   for j = 1:popSize
      p = population(i);
   if (rand < muProbility)
   mutPop=mutPop+1;
   k= randi(nItem);
   p.Solution(k) = 1- p.Solution(k);
   p.FitnessValue = FitnessFunction(p.Solution);
   p.Weight = WeighFunction(p.Solution);
   populationMutation(mutPop) = p;
   end
end

populationMutation = populationMutation(:);
   population = [population
   populationCrossover
   populationMutation];
FitnessValues = [population.FitnessValue];
[FitnessValues, SortOrder] = sort(FitnessValues,'descend');
```

```
population = population(SortOrder);
population = population(1:popSize);
FitnessValues = FitnessValues(1:popSize);

BestSol = population(1);

BestFitness(t) = BestSol.FitnessValue;
disp(['Generation : ' num2str(t) ': Best Fitness value = ' num2str(BestFitness(t))]);
end
  plot(1:maxIter,BestFitness);
```

BÖLÜM 10

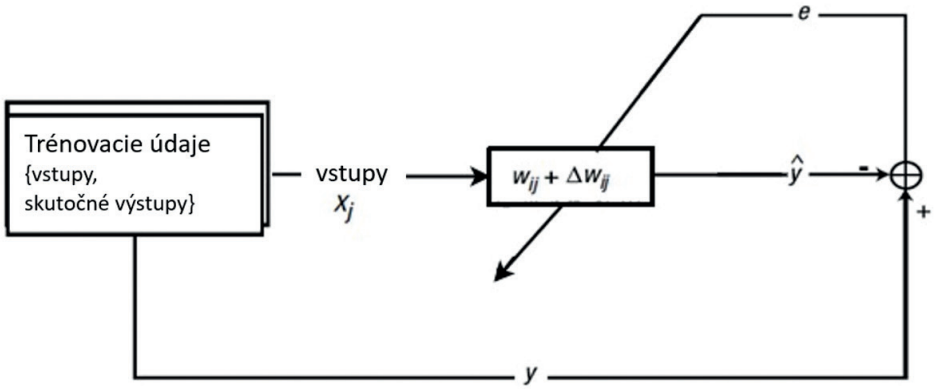
TEK KATMANLI SİNİR AĞI (PERSEPTRON)

Bu bölüm, Adana Alparslan Türkeş Bilim Bilim ve Teknoloji Üniversitesi Mühendislik Fakültesi'nden Önder Tutsoy tarafından yazılmıştır.

Sinir ağları (SA'lar), bilgiyi gözetimli (desen tanıma) veya gözetimsiz (fonksiyon yaklaşımı) öğrenme perspektiflerinden öğrenilen ağırlıklar biçiminde depolarlar. SA'lar, esasen gerçek sistemlerin yaklaşık temsilinde kullanılan parametrik olmayan modelleme yaklaşımlarıdır. Bu nedenle, analitik (derinlemesine ve kesin matematiksel) analizi zordur. SA'ları eğitmek için ağırlıkların girişler aracılığıyla sağlanan bilgiye dayalı olarak güncellenmesi gerekmektedir. Ağırlıkların güncellenmesi için kullanılan sistemli yaklaşım, öğrenme kuralı olarak adlandırılır ve sağlanan giriş bilgisini kullanır. Temel olarak giriş bilgisini çıkış bilgisine eşler. SA'ların bilgiyi sistematik olarak saklaması ve hatırlaması için eğitim tek yol olduğundan, öğrenme kuralı öğrenme sürecinin vazgeçilmez bir bileşeni olarak kabul edilir ve aşağıda tartışılmaktadır.

10.1 DELTA KURALI

Delta Kuralı, tek katmanlı SA'nın temsili bir öğrenme kuralıdır. Tek katmanlı bir SA'nın eğitim süreci aşağıdaki şekilde gösterilebilir.



Şekil 1. Tek katmanlı bir SA'nın eğitim sürecinin blok diyagramı

Önemli bir nokta, tek katmanlı SA'nın tek girişli tek çıkışlı (SISO), tek girişli çoklu çıkışlı (SIMO), çoklu girişli tek çıkışlı (MISO) veya çoklu girişli ve çoklu çıkışlı (MIMO) olabilmesidir. Giriş ve çıkış sayısı öğrenme probleminin karakterine göre değişir. Ayrıca, bağlantılı dinamikler yalnızca birden fazla giriş veya çıkışa sahip SA'lar tarafından öğrenilebilir.

Ancak, öğrenme problemi eşleştirilmiş değilse, ancak SA'nın öğrenme problemi eşleştirilmiş olarak inşa edilirse, SA'ların verimliliği azalacaktır. Bu nedenle, başlangıçta giriş verilerinin karakteri analiz edilmeli ve giriş verileri hakkında elde edilen içgörülere dayanarak, SA'lar inşa edilmelidir.

Sinir ağında m girişleri ve n çıkışları için delta öğrenme kuralının sözde kodu:

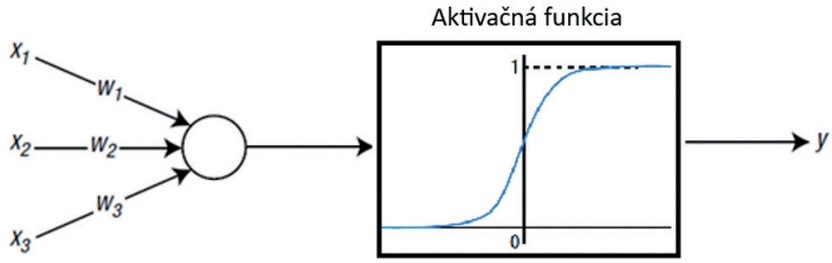
1. Giriş : Eğitim girdisi olarak $x \in \mathbb{R}^{mxl}$, burada l, m sayısı giriş verilerinin her birinin uzunluğudur, etiketli çıkış olarak $y \in \mathbb{R}^{lxn}$, burada n çıkış sayısıdır, rastgele başlatılmış bilinmeyen parametre matrisi/vektörü olarak $w \in \mathbb{R}^{mxn}$, tahmin edilen sonuç olarak $\hat{y} \in \mathbb{R}^{nxl}$, doygun ve tahmini sonuç $e \in \mathbb{R}^{lxn}$, eğitim hatası olarak $\hat{y}_o \in \mathbb{R}^{nxl}$, öğrenme hızı parametresi olarak $0<\eta\leq 2/x(:,1)^T x(:,1)$, çoklu simülasyon sayısı simMultiple, matris depolama w_s, hata depolama e_s, hata durdurma eşiği e_t, tahmini çıktıyı depolama $\hat{y}_o$.
2. **Çıkış:** Eğitimli parametrelerin son değeri w, öğrenme hatası depolama e_s, çıkışı depola $\hat{y}_s$
3. for i to simMultiple
4. for j to l
5. 1. Tahmini güncel çıktıyı hesapla $\hat{y}_o$
6. $\hat{y}_o(:,j) = w^T x(:,j)$
7. 2. Çıktıya(gerekirse) bir eşik değeri σ uygulayın.
8. $\hat{y}(:,j) = \sigma\left(\hat{y}_o(:,j)\right)$
9. 3. Hatayı belirle
10. $e(:,j) = y - \hat{y}(:,j)$
11. 4. Bilinmeyen parametreleri güncelle ve depola
12. $w \rightarrow w + \eta e(:,j) x(:,j)^T$
13. $w_s = \left[w_s; reshape\left(w_s, 1, []\right)\right]$
14. end j
15. Hatayı ve doygun çıktıyı depola
16. $e_s = \left[e_s; reshape\left(e, 1, []\right)\right]$
17. $\hat{y}_s = \left[\hat{y}_s; reshape\left(\hat{y}, 1, []\right)\right]$
18. If $e(:,j) < e_t$ then
19. break
20. end if
21. end i

Delta kuralı, bilinmeyen parametreleri bir seferde çözmek yerine yinelemeli olarak günceller. Gradyan inişi kullanan bir tür sayısal yinelemeli yöntemdir.. Gradyan inişi, başlangıç değerinden başlar ve çözüme doğru ilerler. Adını, bir topun en dik yol boyunca yokuş aşağı yuvarlanırken çözümü aradığı

davranışından alır. Bu benzetmede topun pozisyonu, modelden gelen ara sıra çıkan çıktıdır ve taban çözümdür. Yinelemeli gradyan iniş yönteminin topu tek bir atmayla düşüremeyeceği dikkate değerdir. Tüm süreç tekrarlanır, çünkü aynı verilerle modelin yeniden eğitilmesi modeli iyileştirebilir.

Örnek: Delta Kuralı

Aşağıdaki şekilde gösterilen üç giriş düğümü ve bir çıkış düğümü içeren bir SA düşünün.



Şekil 2. Üç giriş düğümü ve bir çıkış düğümünden oluşan SA

Şekil 2'den de görülebileceği gibi, çıkış düğümünün aktivasyon işlevi olarak sigmoid fonksiyonu kullanılır. Aşağıdaki tabloda gösterilen dört eğitim veri noktasına sahibiz.

Tablo 1 - Etiketli VEYA(OR) Kapısı eğitim veri noktaları

{0,0,1,**0**}
{0,1,1,**1**}
{1,0,1,**1**}
{1,1,1,**1**}

Denetimli öğrenme için kullanıldıklarından, her veri noktası bir giriş-doğru çıkış çiftinden oluşur. Her veri setinin son koyu renkli numarası doğru çıkıştır. Bu 1'inin yanılması olarak girişin son değerine sahip bir VEYA Kapısı sorunudur.

Tek katmanlı olduğu ve basit eğitim verileri içerdiği için kod karmaşık değildir. Kodu takip ettiğinizde SA'nın öğrenme davranışını açıkça anlayacaksınız. İlgili kod aşağıdaki gibi ilerler:

Başlangıçta, eğitim parametreleri aşağıdaki "trainPar" işleviyle tanımlanır:

```
function trainPar = trainParameters()
% Eğitim giriş verisi, son değer 1'in yanlılığı temsil ettiği yer
trainPar.x = [0 0 1; 0 1 1; 1 0 1; 1 1 1]';
% Etiketlenmiş çıkış verileri
trainPar.y = [0 1 1 1]';
```

```
% Rastgele başlatılmış bilinmeyen parametreler
trainPar.w = rand(size(trainPar.x,1),size(trainPar.y,2));
% Tahmin edilen sonucu başlatma
trainPar.yo_hat = zeros(size(trainPar.x,2),size(trainPar.y,2));
% Tahmin edilen sonucu başlatma
trainPar.y_hat = zeros(size(trainPar.x,2),size(trainPar.y,2));
% Hatayı başlatma
trainPar.e = zeros(size(trainPar.x,2),size(trainPar.y,2));
% Öğrenme hızının başlatılması
trainPar.mu = zeros(size(trainPar.y));
% Öğrenme hızı üst sınıra çıkarma
trainPar.mur = 2;
% Hata eşiğini durdurma
trainPar.et = 0.001;
% Çoklu eğitimlerin sayısı
trainPar.simMultiple = 1000;
% Çıkış doygunluk işlevi üst sınırı (sigmoid)
trainPar.satUppper = 1;
end
```

İlgili eğitim parametrelerini tanımladıktan sonra, öğrenme süreci için aşağıdaki fonksiyon kullanılır.

```
% Bu m-dosyası VEYA problemi için tek katmanlı bir SA çalıştırır.
% Eğitim parametrelerini yükleyin
trainPar = trainParameters();
% Tahsis edilen hatayı yükleyin
e = trainPar.e;
% Tahsis edilen hesaplanmış hatayı yükleyin
yo_hat = trainPar.yo_hat;
Eşikle tahsis edilen çıktıyı yükleyin
y_hat = trainPar.y_hat;
Tahsis edilen bilinmeyen parametreyi yükleyin
w = trainPar.w;
Tahsis edilen öğrenme hızını yükleyin
mu = trainPar.mu;
```

```
% Bilinmeyen parametre için depolama matrisini tanıtın
ws = [];
% Hata için depolama matrisini tanıtın
es = [];
% Tahmini çıkış için depolama matrisini tanıtın
ys_hat = [];
for i=1:trainPar.simMultiple
   for j=1:size(trainPar.x,2)
      % Tahmin edilen güncel çıktı hesapla
      yo_hat(j,:) = w'*trainPar.x(:,j);
      % Tahmin edilen çıktı için bir eşik uygula
      y_hat(j,:) = satOutput(yo_hat(j,:),trainPar);
      % Anlık hatayı belirle
      e(j,:) = trainPar.y(j,:) - y_hat(j,:);
      % Öğrenme hızını güncelle
      mu(i,j) =trainPar.mur/(trainPar.x(:,j)'*trainPar.x(:,j));
      % Bilinmeyen parametre vektörünü/matrisini güncelle
      w = w + mu(i,j)*e(j,:)*trainPar.x(:,j);
      % Bilinmeyen parametre vektörünü/matrisini depola
      ws = [ws;reshape(w,1,[])];
   end
   % Hata geçmişini depola
   es = [es;reshape(e,1,[])];
   % Tahmin edilen çıktıyı depola
   ys_hat = [ys_hat;reshape(y_hat,1,[])];
end
```

Sigmoid aktivasyon fonksiyonu için oluşturulan satOutput fonksiyonu aşağıda olduğu gibi oluşturulmuştur.

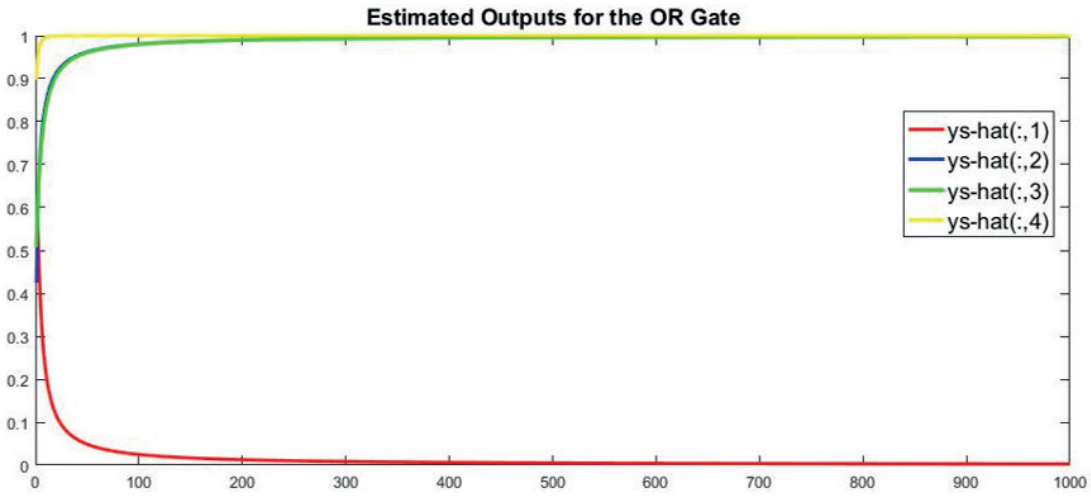
```
function y_sat = satOutput(y_unsat, trainPar)
y_sat = trainPar.satUpppper / (1 + exp(-y_unsat));
end
```

*Daha sonra her bir giriş kümesi için tahmini çıktılar **ys_hat** aşağıdaki kod bloğu kullanılarak çizilir:*

```
figure(1),
```

```
plot(1:length(ys_hat),ys_hat(:,1),'r','LineWidth',2),
hold on,
plot(1:length(ys_hat),ys_hat(:,2),'b','LineWidth',2),
plot(1:length(ys_hat),ys_hat(:,3),'g','LineWidth',2),
plot(1:length(ys_hat),ys_hat(:,4),'y','LineWidth',2),
hold off
title('Estimated Outputs for the OR Gate')
```

Bu kodun uygulanması aşağıdaki şekli ortaya çıkarır:



Şekil 3. VEYA Kapısı için tahmin edilen çıktı sonuçları

Bu kodu çalıştırmak, aşağıdaki değerleri üretir. Bu çıkış değerleri, hedef y değerlerindeki doğru çıkışlara çok yakındır. Bu nedenle, SA'nın VEYA Kapısını öğrenmek için uygun bir şekilde eğitildiği sonucuna varabiliriz.

$$\begin{bmatrix} 0.0025 \\ 0.9980 \\ 0.9980 \\ 1.0000 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

```
Eğitim hatasını çizmek için ,
figure(),
plot(1:length(es),es(:,1),'r','LineWidth',2),
hold on,
plot(1:length(es),es(:,2),'b','LineWidth',2),
plot(1:length(es),es(:,3),'g','LineWidth',2),
```

```
plot(1:length(es),es(:,4),'y','LineWidth',2),
hold off
title('Training Error for the OR Gate')
```

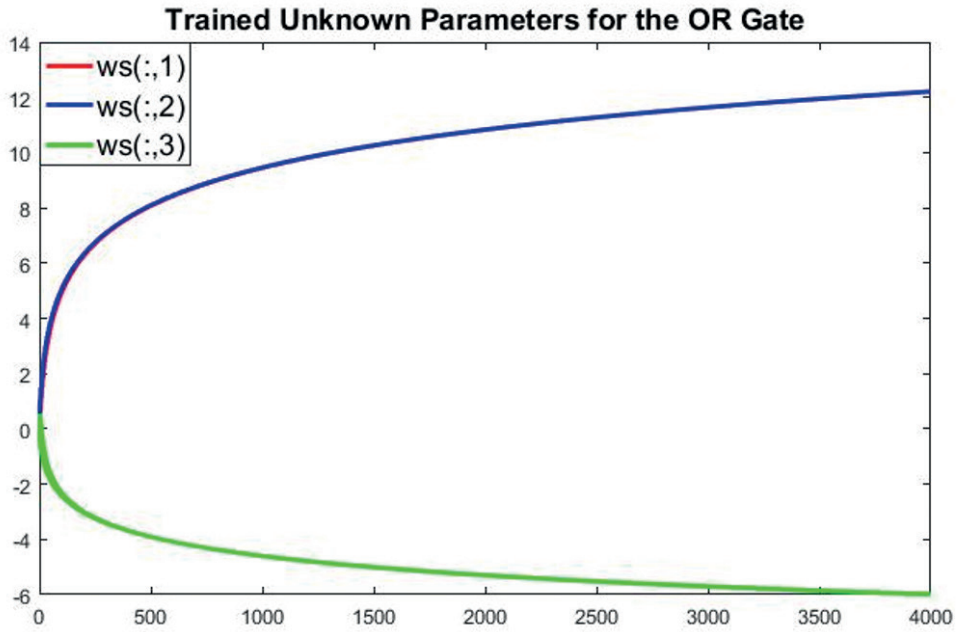
kod bloğu kullanılır ve elde edilen sonuç şu şekilde temsil edilir:

Şekil 4. VEYA Kapısı için Eğitim Hatası sonuçları

Şekil 4'ten görülebileceği gibi, hata, karşılık gelen VEYA Kapısı veri noktaları için sırasıyla sıfıra yaklaşır.

Son olarak, eğitilmiş bilinmeyen parametreler çizilir ve aşağıdaki kod bloğu kullanılarak gösterilir:

```
figure(),
plot(1:length(ws),ws(:,1),'r','LineWidth',2),
hold on,
plot(1:length(ws),ws(:,2),'b','LineWidth',2),
plot(1:length(ws),ws(:,3),'g','LineWidth',2),
plot(1:length(ws),ws(:,4),'y','LineWidth',2),
hold off
title('Trained Unknown Parameters for the OR Gate')
```



Şekil 5. VEYA Kapısı için Eğitilmiş Bilinmeyen Parametreler

Şekil 5'te açıkça görüldüğü gibi, yalnızca üç eğitilmiş bilinmeyen parametre çizildi. Bu, aşağıdaki kod bloğunda matrislerin birbirleriyle çarpılmasından kaynaklandı.

```
trainPar.w = rand(size(trainPar.x,1),size(trainPar.y,2));
```

trainPar.x'in boyutu 3x4 ve trainPar.y'ın boyutu 4x1 olduğuna göre, bir 3x1 vektörü belirlemek mümkün olacaktır. Aslında, burada tüm eğitilmiş bilinmeyen parametreleri çizmek oldukça basittir. Tüm bu açıklamalardan sonra, lütfen kodda gereken güncellemeyi yapın.

10.2 TEK KATMANLI SA'LARIN SINIRLILIKLARI

Bu bölüm, tek katmanlı SA'nın neden çok katmanlı bir SA'ya evrilmesi gerektiğini açıklar Bununla ilgili belirli bir durumu göstermeye çalışacağız. Önceki bölümde tartışılan aynı SA'yı düşünün. Bu, üç giriş düğümü ve bir çıkış düğümünden oluşur ve çıkış düğümünün aktivasyon işlevi bir sigmoid işlevidir. Aşağıda gösterilen gibi dört eğitim veri noktasına sahip olduğumuzu varsayalım.

Tablo 2. Etiketli Özel VEYA kapısı (XOR) eğitim veri noktaları

{0,0,1,**0**}
{0,1,1,**1**}
{1,0,1,**1**}
{1,1,1,**0**}

Tablo 2'de verildiği gibi, bu 1'in yanılması olarak girişin son değerine sahip olan bir Özel VEYA Kapısı (XOR) problemidir. Bu, 'Delta Kuralı' bölümünden farklı olarak, ikinci ve dördüncü doğru çıktılar değiştirilirken girişler aynı kalır. Fark neredeyse anlaşılmaz.

Aynı SA'yı düşündüğümüzden, daha önce de belirttiğimiz gibi y için farklı değerlere sahip olması dışında, "Örnek:Delta Kuralı" bölümünden "trainPar" fonksiyonunu kullanarak onu eğitebiliriz. Kodu çalıştırmadan önce "trainPar" fonksiyonundaki etiketli çıkış veri kod bloğu aşağıdaki gibi güncellenir.

```
% Etiketli çıkış verisi
trainPar.y = [0 1 1 0]';
```

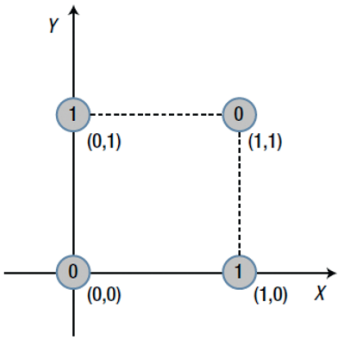
Bu kodun çalıştırılması aşağıdaki değerleri üretir. Bu çıkış değerleri hedef y değerindeki doğru çıkışlara çok yakındır. Bu nedenle, SA'nın VEYA Kapısını öğrenmek için uygun şekilde eğitildiği sonucuna varabiliriz.

$$\begin{bmatrix} 0.5297 \\ 0.5000 \\ 0.4703 \\ 0.4409 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Belirlenen denklemden de görülebileceği gibi birbirinden tamamen farklı iki setimiz var. SA'nIn daha uzun süre eğitilmesi bir fark yaratmaz.

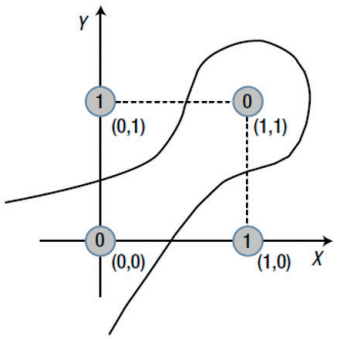
Gerçekten ne oldu?

Eğitim verilerinin gösterilmesi bu sorunun aydınlatılmasına yardımcı olabilir. Giriş verilerinin üç değerini sırasıyla X, Y ve Z koordinatları olarak yorumlayalım. Üçüncü değer (Z koordinatı) 1 olarak sabitlendiğinden eğitim verileri aşağıdaki şekilde gösterildiği gibi bir düzlem üzerinde görselleştirilebilmektedir.



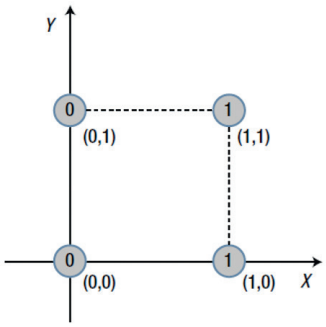
Şekil 6. Giriş verilerinin üç değerinin X, Y ve Z koordinatları olarak yorumlanması

Dairelerdeki 0 ve 1 değerleri, her noktaya atanan doğru çıktılardır. Bu şekilde dikkat edilmesi gereken nokta, 0 ve 1 bölgelerini düz bir çizgiyle bölemeyeceğimizdir. Ancak bunu aşağıdaki şekilde gösterildiği gibi karmaşık bir eğri ile bölebiliriz. Bu tür problemlerin doğrusal olarak ayrılamaz olduğu söylenir.



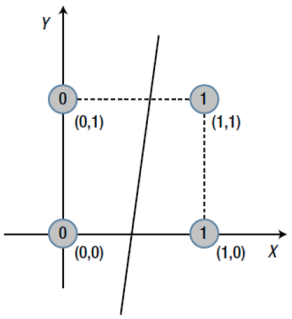
Şekil 7. 0 ve 1'i karmaşık bir eğriyle ayırma (doğrusal olarak ayrılamaz)

Aynı süreçte X-Y düzlemindeki "Örnek: Delta Kuralı" bölümündeki eğitim verileri şu şekilde karşımıza çıkıyor:



Şekil 8. The delta rule training data

Bu durumda 0 ve 1 bölgelerini ayıran düz bir sınır çizgisi kolaylıkla bulunabilir. Bu, aşağıdaki şekilde gösterildiği gibi doğrusal olarak ayrılabilir bir sorundur:



Şekil 9. Doğrusal olarak ayrılabilir problem

Basitçe söylemek gerekirse, tek katmanlı SA yalnızca doğrusal olarak ayrılabilir problemleri çözebilir. Bunun nedeni, tek katmanlı SA'nın giriş veri alanını doğrusal olarak bölen bir model olmasıdır. Tek katmanlı SA'nın bu sınırlamasını aşmak için ağda daha fazla katmana ihtiyacımız var. Bu ihtiyaç, tek katmanlı NN'nin başaramadığını başarabilen çok katmanlı SA'nın ortaya çıkmasına neden olmuştur. Tek katmanlı SA'nın belirli problem türleri için geçerli olduğunu unutmayın. Çok katmanlı SA'nın böyle bir sınırlaması yoktur. Lütfen daha fazla ayrıntı için aşağıdaki referanslara bakın.

BÖLÜM 11

SİNİR AĞ UYGULAMASI

El kitabının bu bölümü Slovakya'nın Banská Bystrica şehrindeki Matej Bel Üniversitesi Doğa Bilimleri Fakültesi Bilgisayar Bilimleri Bölümü'nden Jarmila Škrinárová tarafından yazılmıştır.

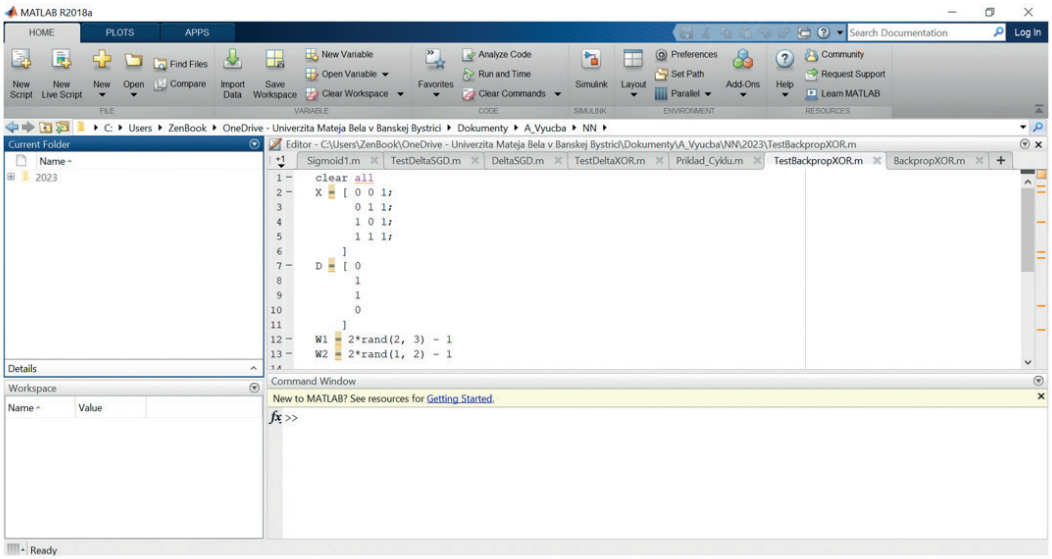
Matlab, sayısal hesaplama, görselleştirme ve programlamaya yönelik üst düzey bir dil ve etkileşimli bir ortamdır ve şu alanlarda kullanılır :

- Veri analizi,
- Algoritma geliştirme,
- Model ve uygulama oluşturma

Bu bölümün amacı Matlab ile nasıl çalışılacağını ve basit sinir ağlarının nasıl oluşturulacağını öğrenmektir. Grafiksel Matlab ortamında bir metodoloji ve üç sinir ağı örneği sunuyoruz. Örnekler, uyarlanabilir fonksiyon oluşturma ve sınıflandırma görevlerine yöneliktir.

11.1 MATLAB'A KISA GİRİŞ – MATRİX LABORATORY

Öncelikle Matlab ortamını tanıtıyoruz. Pencerenin üst tarafında bir araç çubuğu bulunmaktadır. Araç çubuğunun altında alan, gezinme (dizin yapısı arasında gezinme) yürütülebilir komut dosyalarını düzenleme, çalışma alanını ve bir komut penceresini görüntüleme amaçlı dört pencereye bölünmüştür (bkz. Şekil 1).



Şekil 1. Düzenleyici, Komut penceresi, Çalışma alanı, Gezgin

Öncelikle ">>" işaretinden sonra komutları yazdığımız komut penceresinde çalışmayı öğreneceğiz. (bkz. Şekil 2).

Şekil 2. Komut penceresi

Basit hesaplamalar, değişkenlerle çalışma, vektörler ve matrislerle işlemler için örnekler

Değişkenlerle hesaplamalar - 1'den 4 kadarki örnekler doğrudan komut penceresinde aşama aşama uygulanabilir:

Örnek 1	Örnek 2	Örnek 3	Örnek 4
>> 12+34 ans = 46	**>> a =5, b = a^2** a = 5 b = 25	**>> (101+79) /(47 -17)** ans = 6	**>> 15*12** ans = 180

Vektör, tek boyutlu bir öğe dizisidir. Vektörlerin bireysel öğelerini genellikle köşeli parantez içinde yazarız, ve virgül veya boşlukla ayırırız. Notu % işareti sonrasında yazdığımıza dikkat edin. Bu bölümün ilerleyen kısımlarında sinir ağlarıyla çalışacağız ve sinir ağı öğrenimi için giriş ve hedef verilere ihtiyacımız olacak. Eğer ağın bir girişi varsa, giriş verisi tek boyutlu bir öğe dizisi (vektör) şeklinde olur. Eğer ağın bir çıkışı varsa, hedef verisi de tek boyutlu bir öğe dizisi şeklinde olur [3].

Örnek 5:	6:	Örnek 7:	Örnek 8:
>> **u1=[1 2 3 4]** %row vector u1 =1 2 3 4	>> Örnek **u2=[1 2 1 2]** %row vector u2 = 1 2 1 2	**>> u1.*u2** %scalar product of two vectors ans = 1 4 3 8	>> **v=[-1; -7; -3]** %column vector v = -1 -7 -3

Örnek 9:	Örnek 10:	Örnek 11:	Örnek 12:
>> w=[1 7 -2]' %transposed vector w = 1 7 -2	**>> 6:2:12** % To generate a regular vector, we define the first and last elements of the vector and the step. ans = 6 8 10 12	**>> m=15:-3:0** m = 15 12 9 6 3 0	**>> x=12** x =12 **>> z=[x, 2*x, 3*x]** z =12 24 36

Örnek 13:	Örnek 14:
>> W=2*rand(1,3)-1 W = 0.9298 -0.6848 0.9412	**>> x2=linspace(-1, 4, 8)** % -1 to 4 is interval and 8 is number of elements x2 =-1.0000 -0.2857 0.4286 1.1429 1.8571 2.5714 3.2857 4.0000

Sinir ağlarında birden fazla giriş veya hedef kullanmak için verileri iki boyutlu alanlar şeklinde hazırlamamız gereklidir. Matlab'da iki boyutlu diziler matrislerle temsil edilir. Bu nedenle, matrislerle çalışmayı uygulayacağız:

Örnek 15:	Örnek 16:
>> A=[1 -1 2 -3; 3 0 4 5; 3.2, 5 -6 12] %matrix A = 1.0000 -1.0000 2.0000 -3.0000 3.0000 0 4.0000 5.0000 3.2000 5.0000 -6.0000 12.0000	**>> O=[]** %empty matrix O = []

Örnek 17:	Örnek 18:
>> **B=[A; u1]** %Matrix expansion by 1 row (vector u1). B = 1.0000 -1.0000 2.0000 -3.0000 3.0000 0 4.0000 5.0000 3.2000 5.0000 -6.0000 12.0000 1.0000 2.0000 3.0000 4.0000	**>> C=[A, v]** %Extending the matrix by 1 column (vector v). C =1.0000 -1.0000 2.0000 -3.0000 -1.0000 3.0000 0 4.0000 5.0000 -7.0000 3.2000 5.0000 -6.0000 12.0000 -3.0000

Örnek 19:	Örnek 20:
>> Z=zeros(2,5) %Creating a null matrix of size 2 rows by 5 columns. Z = 0 0 0 0 0 0 0 0 0 0	**>> O1=ones(3,4)** %Creating a unit matrix with dimension 3 rows by 4 columns. O1 = 1 1 1 1 1 1 1 1 1 1 1 1

Örnek 21:	Örnek 22:
>> A=[1 -1 2 -3; 3 0 4 5; 3.2, 5 -6 12] A = 1.0000 -1.0000 2.0000 -3.0000 3.0000 0 4.0000 5.0000 3.2000 5.0000 -6.0000 12.0000 **>> A(2, :)** % Listing of the 2nd row of matrix A ans = 3 0 4 5 **>> A(:, 3)** % Listing of the 3rd column A matrix ans = 2 4 -6	**>> I=eye(5,8)** %Creating a diagonal matrix of size 5 rows by 8 columns. I =1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0
Örnekl 23:	Örnek 24:
>> R1=rand(3,5) %Create a random matrix of size 3 rows by 5 columns, with values in the range 0 to 1 R1 = 0.1419 0.7922 0.0357 0.6787 0.3922 0.4218 0.9595 0.8491 0.7577 0.6555 0.9157 0.6557 0.9340 0.7431 0.1712	**>> R2=randn(4)** %matrix with random elements - standard distribution R2 =0.8884 -2.9443 1.3703 0.3192 -1.1471 1.4384 -1.7115 0.3129 -1.0689 0.3252 -0.1022 -0.8649 -0.8095 -0.7549 -0.2414 -0.0301

Exemplul 25:	Exemplul 26:
>> A=[1 5 0; -1 2 3; 1 2 1] A = 1 5 0 -1 2 3 1 2 1 **>> c=2** c = 2 %matrix multiplication by vector **>> D=A*c** D =2 10 0 -2 4 6 2 4 2	**>> B=[1 2 3; 1 2 3; 1 2 3]** B =1 2 3 1 2 3 1 2 3 **>> E=B*D** %matrix multiplication E = 4 30 18 4 30 18 4 30 18
Örnek 27:	Örnek 28:
>> A =[2 3; 0 10] B =[1 0; -3 5] %matrix multiplication A = 2 3 0 10 B =1 0 -3 5 **>> C=A*B** C = -7 15 -30 50	%matrix scalar multiplication, matrices A and B are from previous example **>> C=A.*B** C =2 0 0 50

Çoğu zaman birçok öğe içeren verileri kullanırız. Bu nedenle bu verileri daha sonra kullanabilmek için bir **dosyaya** yazmamız pratik olur. Matlab'da çalıştığımız tüm veriler çalışma alanında depolanır ve sol alt pencerede görülebilir. Örnek 29 ve 30'u kullanarak çalışma alanının içeriği hakkındaki bilgileri görüntüleyebiliriz.

Örnek 29:	Örnek 30:
>> **who** % workspace listing only with names of variables, vectors and matrices Your variables are: A B C O ans u v w x z	>> **whos** % workspace Name Size Bytes Class Attributes A 3x4 96 double B 4x4 128 double C 3x5 120 double O 0x0 0 double ans 1x1 8 double u1 1x4 32 double u2 1x4 32 double v 3x1 24 double w 3x1 24 double x 1x1 8 double z 1x3 24 double

Çalışma alanındaki tüm değişkenleri, vektörleri ve matrisleri dosyaya (bkz. örnek 31) veya yalnızca seçilen değişkenleri, vektörleri ve matrisleri (bkz. örnek 32) **kaydedebiliriz**.

Örnek 31:	Örnek 32:
>> **save data** % save all data to file data.dat	>> **save data1 u1 u2 v** % save variables u1, u2 and v into data1.mat

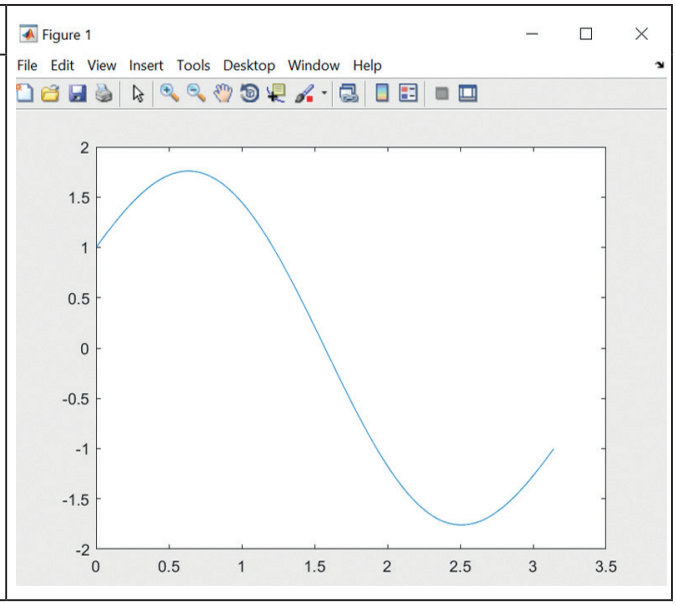
Dosyalarda saklanan veriler, onunla çalışmak üzere herhangi bir zamanda Matlab aracının çalışma alanına yüklenebilir. 33. ve 34. örneklere bakınız.

Örnek 33:	Örnek 34:
>> **load data1** % read all variables saved in data1.mat	>> **load data.dat -MAT** % read all variables saved in data.dat

Belirli bir fonksiyonun grafiğini **çizerken**, x eksenindeki eleman sayısı y eksenindeki eleman sayısıyla aynı olmalıdır. Fonksiyonu 35. örnekteki resimde görebiliriz. Fonksiyon, düzenlenebilen- eksen adları, başlıklar vb. eklenebilen yeni bir pencerede çizilir.

Örnek 35:

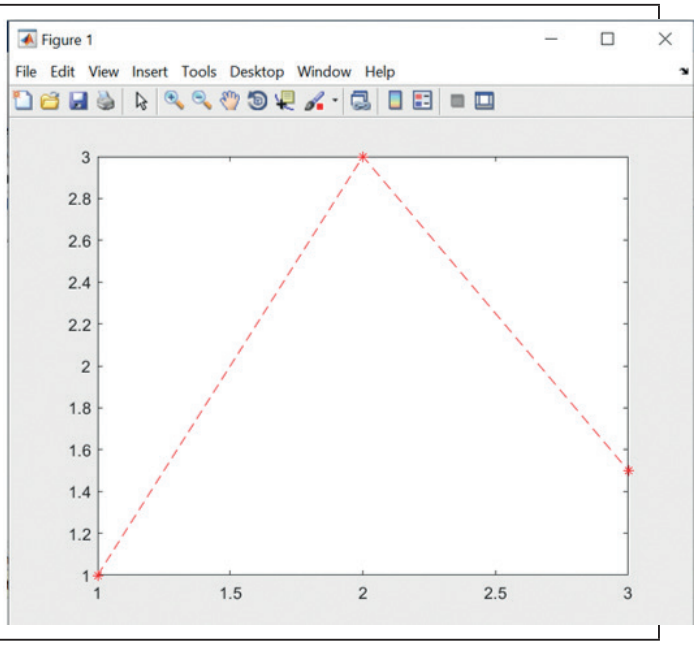
```
>> x=linspace(0, pi);
%plot the function
>> y=sin(2*x)+cos(x) ;
>> plot(x,y)
```



Grafik komutunda hem **renk** hem de **çizgi türü** değiştirilebilir. Örnek 36'ye bakınız.

Örnek 36:

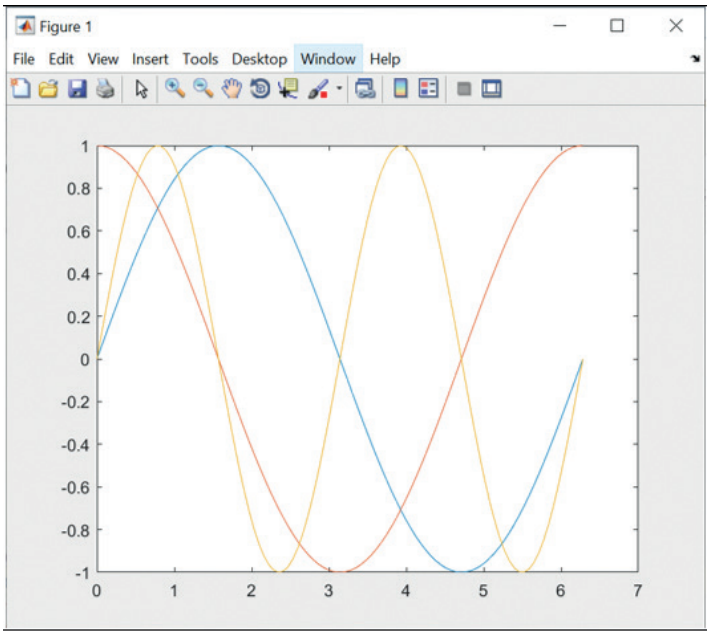
```
>> u=[1 2 3] ;
>> v=[1 3 1.5] ;
>> plot(u,v, 'r--*')
```



Tek bir görüntüde **birden fazla fonksiyonun** grafiğini çizmek de mümkündür. Örnek 37'ye bakınız

Örnek 37:

```
>> x=linspace(0, 2*pi);
>> y1=sin(x);
>> y2=cos(x);
>> y3=sin(2*x);
>> plot(x, y1, x, y2, x, y3)
```



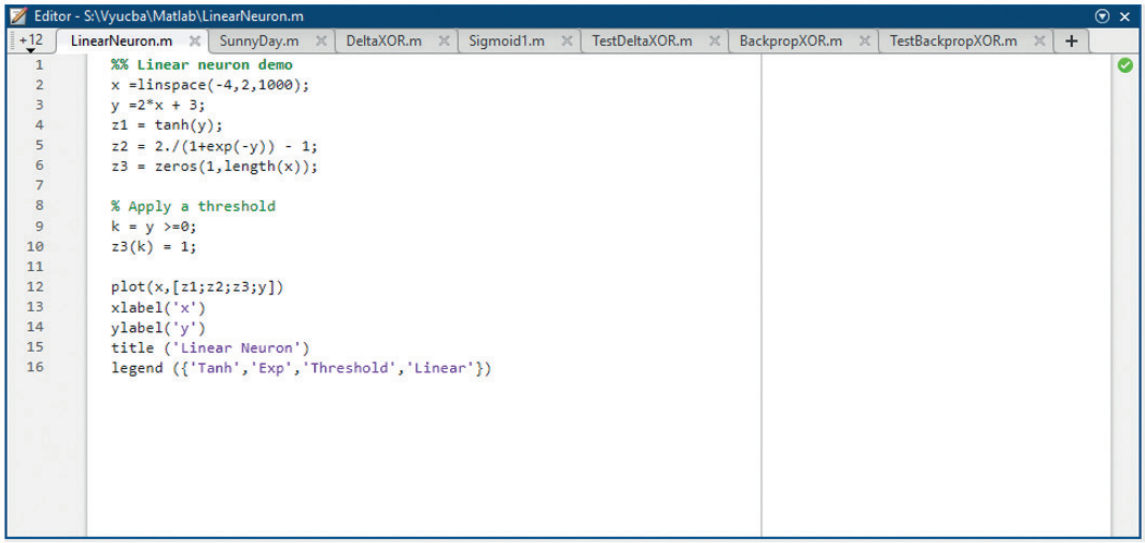
Örnek olarak, X matrisinin tek tek satırlarının bir döngü içinde sırayla yazıldığı basit bir program örneğini (örnek 38) gösteriyoruz. Programın çıktısı sağ sütundadır.

Örnek 38:

```
>> clear
all X = [ 0
     0 1;
     0 1 1;
     1 0 1;
     1 1 1;
   ] ;

N = 4; % print always one row of matrix X
for k = 1:N
    x = X(k, :)
end
```

```
x = 0 0 1

x = 0 1 1

x = 1 0 1

x = 1 1 1
```

Önceki bölümlerde komut satırında komutların nasıl yazılacağını gösterdik. Çok sayıda komut yazmamız gerekiyorsa bu pratik değildir. Bu tür ardışık komutları bir metin düzenleyicide yazıp "m" uzantılı bir dosyaya kaydediyoruz. Matlab ortamında açılıp çalıştırılacak bir **metni** bu şekilde oluşturuyoruz. Metinde yazdığımız komutlarda hata oluşmaması için öncelikle Matlab komut satırında test edilmesi gerekmektedir.



Şekil 3. Matlab düzenleyici penceresindeki M dosyası örneği

11.2 MATLAB'DA SİNİR AĞLARININ UYGULANMASI

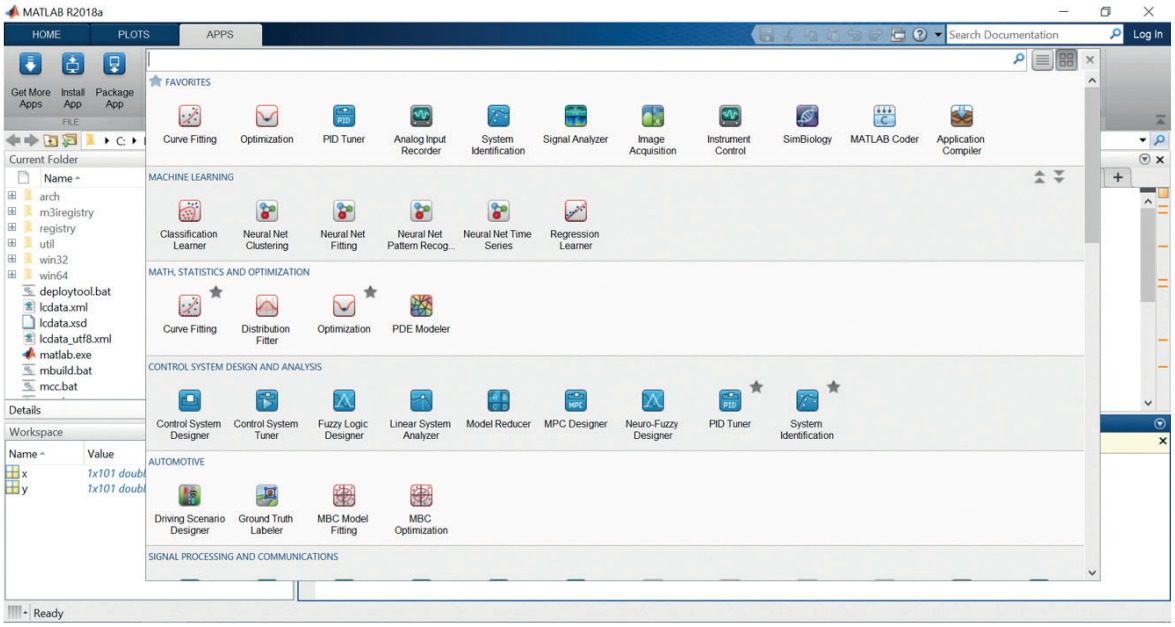
Gerçek dünyada sıklıkla çeşitli ölçümler yapabiliyoruz ancak belirli bir sistemin davranışını basit bir matematiksel modelle tanımlayamıyoruz. Bu, sisteme giren girdilerin ve bunlara karşılık gelen çıktıların ölçülen değerlerine sahip olduğumuz ancak girdilere göre çıktıları hesaplayamadığımız anlamına gelir. Bu amaçla girdiler (belirli bir değer aralığından) ile çıktılar arasındaki ilişkiyi öğrenmek veya girdileri belirli gruplara sınıflandırmak için sinir ağlarını kullanırız. İyi eğitimli sinir ağı farklı giriş değerleri için (aynı aralıktan) doğru çıktılar üretebilmektedir.

Bu alt bölüm, Matlab ortamında sinir ağları oluşturmaya yönelik **metodolojiyi** belirtmeyi amaçlamaktadır. Daha sonra sinir ağı oluşturma sürecini anlamamıza yardımcı olacak **uygulamalı örnekleri** çözeceğiz.

Matlab grafik ortamında sinir ağı oluşturma metodolojisi

Metodolojiyi adım adım açıklayacağız:

1. **Aşama:** Veri Hazırlama. Genellikle iki veri kümesine ihtiyacımız vardır (giriş kümesi ve bunlara karşılık gelen çıktı kümesi). Bir girdinin beklenen bir çıktıya (hedefe) karşılık geldiği bir veri örneğimiz varsa, o zaman her iki veri kümesi de aynı boyuta sahiptir, yani 1 satır x sütun sayısı (örnekler).
2. **Aşama:** Matlab ortamındaki APPS sekmesinden uygun bir uygulamayı seçin. Örneğin, Makine Öğrenimi kategorisinden Nötr Ağ Bağlantısı uygulamasını seçin (bkz. Şekil 4).



Şekil 4. Matlab ortamının parçası olan uygulamalar

Adım 3: Giriş verilerini veri kümesinden ve hedef verileri veri kümesinden (Adım 1'de hazırladığımız) yüklüyoruz.

Adım 4: Verilerin eğitim, doğrulama ve test için üç gruba bölünmesi gereken oranı giriyoruz; örneğin %70, %15 ve %15.

Adım 5: Ağ mimarisini tasarlayacağız. Ağ giriş ve çıkışlarının sayısı, giriş ve hedef verilerine göre otomatik olarak ayarlanır. Gizli katmanlardaki nöron sayısını ayarlamak gerekir. Örneğin, 2 gizli katmanı olan çok katmanlı bir algılayıcı ağı tasarladığımızı varsayalım. Bu durumda birinci ve ikinci gizli katmanlar için nöron sayısını belirtmemiz gerekir.

Adım 6: Öğrenme algoritmasının seçimi. Hazırlanan öğrenme algoritmalarından birini seçiyoruz; örneğin Levenberg - Marquardt, Bayesian Regularization veya ölçekli eşlenik gradyan.

Adım 7: Ağ öğrenme sürecine başlıyoruz. Uygulamada belirli değerlerin önceden ayarlandığı unutulmamalıdır. Örneğin, öğrenme dönemi sayısı 1000'e ayarlanabilir ve öğrenme doğruluğu MSE ve R kullanılarak ifade edilir. Ortalama karesel hata (MSE), eğitim sürecinden sonraki ağ çıkışları ile eğitim sürecinden önceki hedefler arasındaki ortalama karesel farktır. Amacımız hataların en küçük değerlerini elde etmektir. Sıfır değeri hata olmadığı anlamına gelir. Regresyon (R), çıktılar ve hedefler arasında ölçülen korelasyonu ifade eder. R değerinin 1 olması yakın bir korelasyonu, 0 ise korelasyonun olmadığını, başka bir deyişle rastgele bir ilişkinin olduğunu gösterir.

Adım 8: Elde edilen öğrenme doğruluğu bizim için yeterli ise ağ öğrenme süreci sona erer. Aksi takdirde, ağ mimarisinin değiştirilmesi (gizli katmanların sayısı ve bunların içindeki nöronların sayısı) veya öğrenme algoritmasının değiştirilmesi veya mümkünse ağın öğrenme dönemlerinin sayısının değiştirilmesi gerekir. Bu, 5. adımdan itibaren prosedürü tekrarlayacağımız anlamına gelir. Çok sayıda öğrenme döneminin ağ yeniden öğrenmeye yol açabileceği dikkate alınmalıdır.

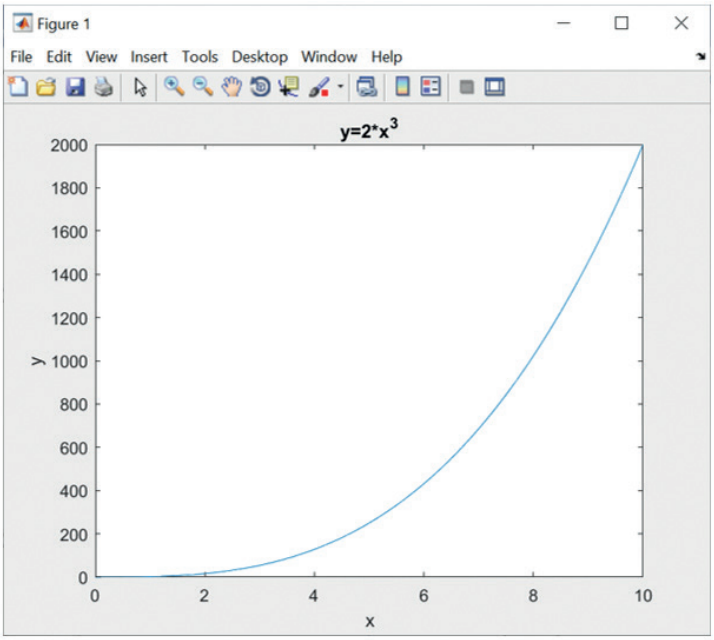
Basit sinir ağı işlevi oluşturma örneği

Bu örnekte bir sinir ağının bir fonksiyonun değerini nasıl öğrendiğini göstereceğiz. Bu bölümün 2.1 kısmında sunulan metodolojiyi takip ediyoruz.

Adım 1: Ölçülen verileri basitlik açısından kullanmayacağız, ancak giriş ve çıkış verilerini Matlab'da oluşturacağız. Matlab'daki komutları kullanarak iki veri kümesi oluşturacağız (aşağıdaki koda bakın). İlk veri kümesi veri1.mat adlı girişleri içerir, girişlerin değerlerini içerir, ikinci veri kümesi ise veri2 adını alır. mat, hedeflerin değerlerini, yani ağı öğrendikten sonra beklenen çıktıları içerir. Girişlerin ve hedeflerin elemanları, bireysel giriş elemanları ilgili hedef elemanlara karşılık gelecek şekilde düzenlenir [2].

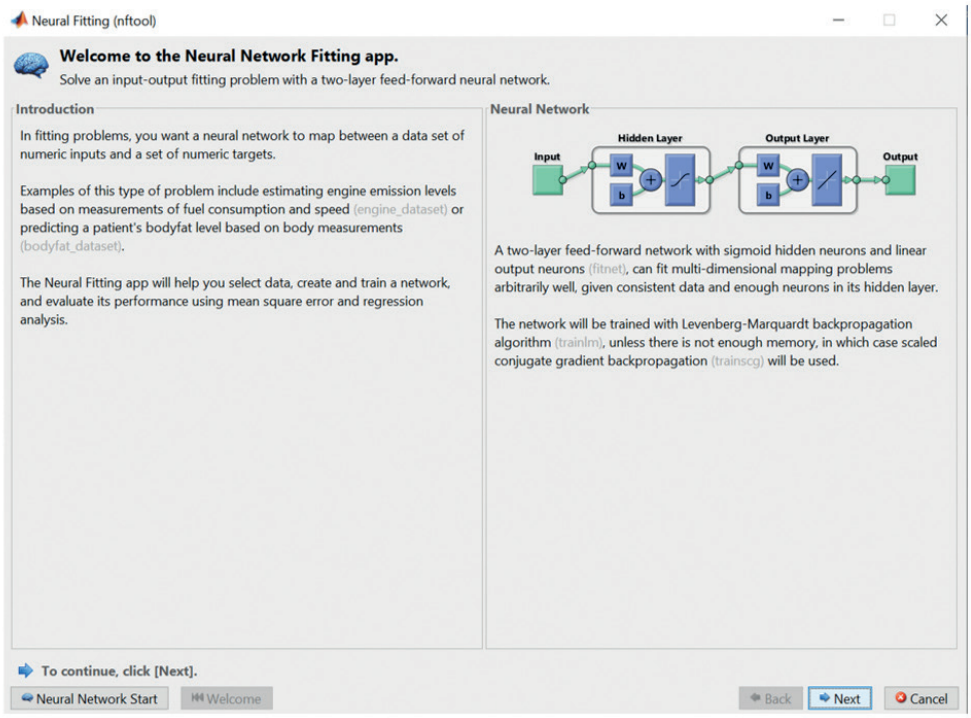
```
>> x=0:0.1:10
>> y=2*x.^3
>> plot(x,y)
>> save data1 x
>> save data2 y
```

Liste 1'deki komutlar çalıştırıldıktan sonra x ve y vektörlerinin değerleri yazılır, fonksiyon grafiği çizilir (bkz. Şekil 5) ve veri kümeleri dosyalara kaydedilir.



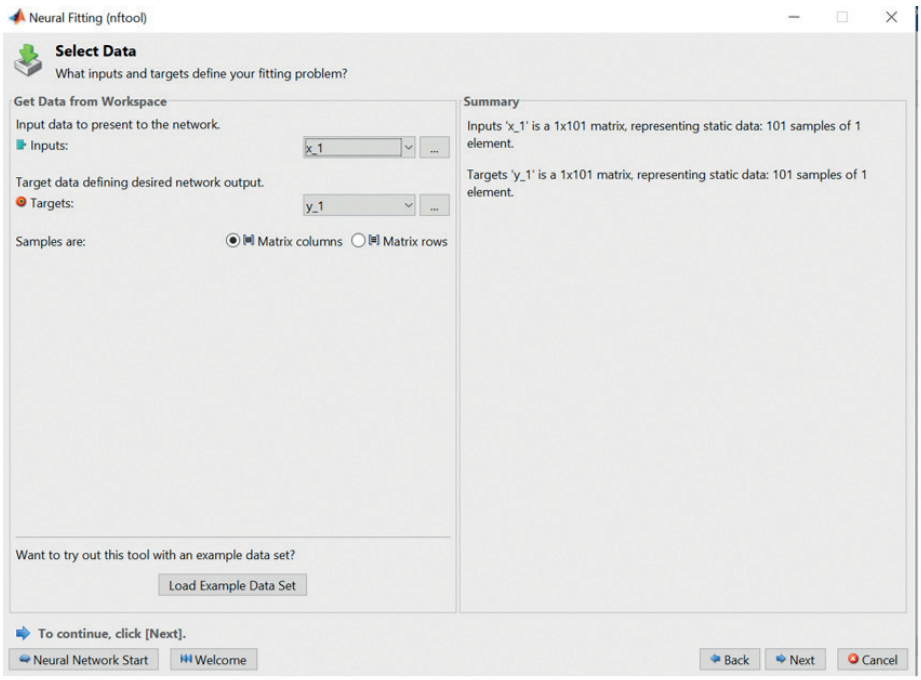
Şekil 5. $y= 2x^3$ fonksiyonunun grafiği

Adım 2: Matlab ortamında APPS sekmesinden uygun bir uygulamayı seçiyoruz. Machine Learning kategorisinden Neutral Net Fitting uygulamasını seçiyoruz. Nötr Ağ Montaj uygulamasına başlayalım (bkz. Şekil 6). Sonraki butonu kullanarak uygulamada geziniyoruz.



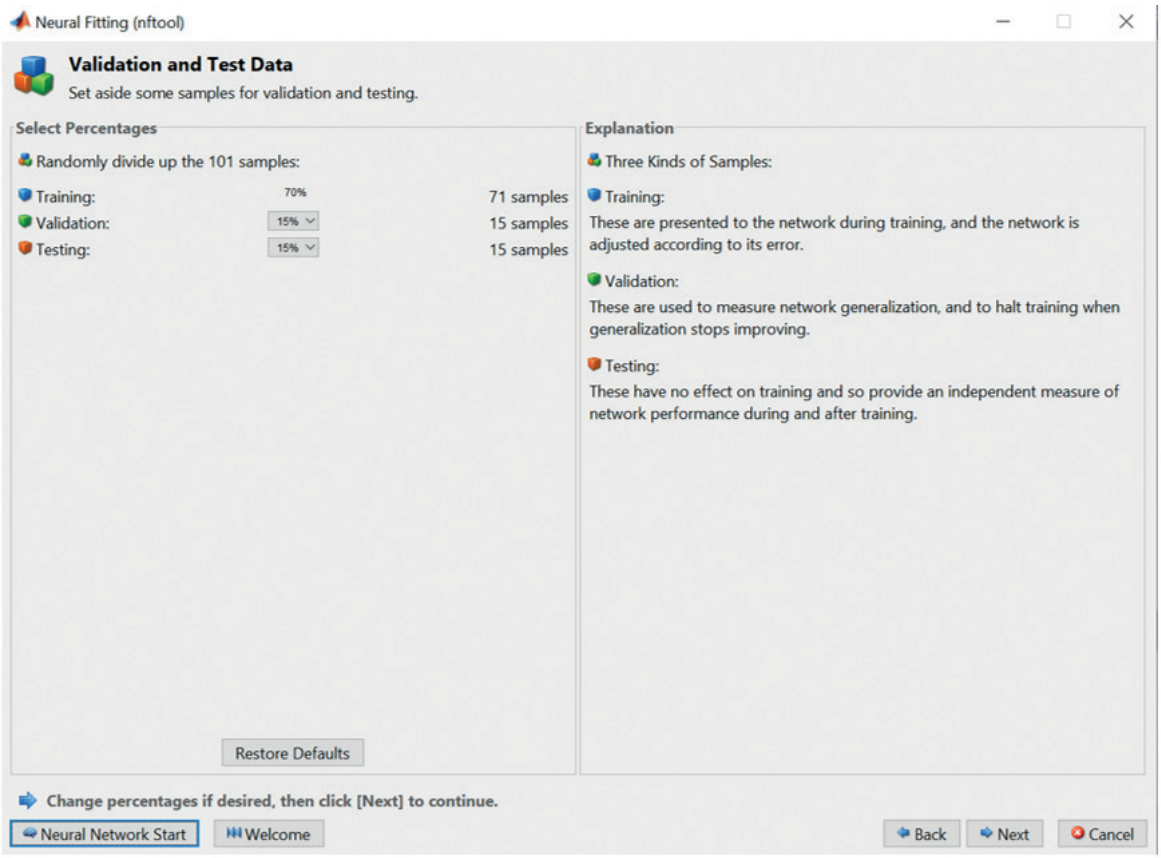
Şekil 6. Matlab'da Sinir Ağı Uydurumu

Adım 3: Hazırlanan dosyalardan girdi ve hedef verinin veri setlerini yüklüyoruz (bkz. Şekil 7, sol). Aynı sayıda giriş ve hedefe sahip olduğumuz için dosyaların aynı boyutta olmasını sağlıyoruz (bkz. Şekil 7, sağ).



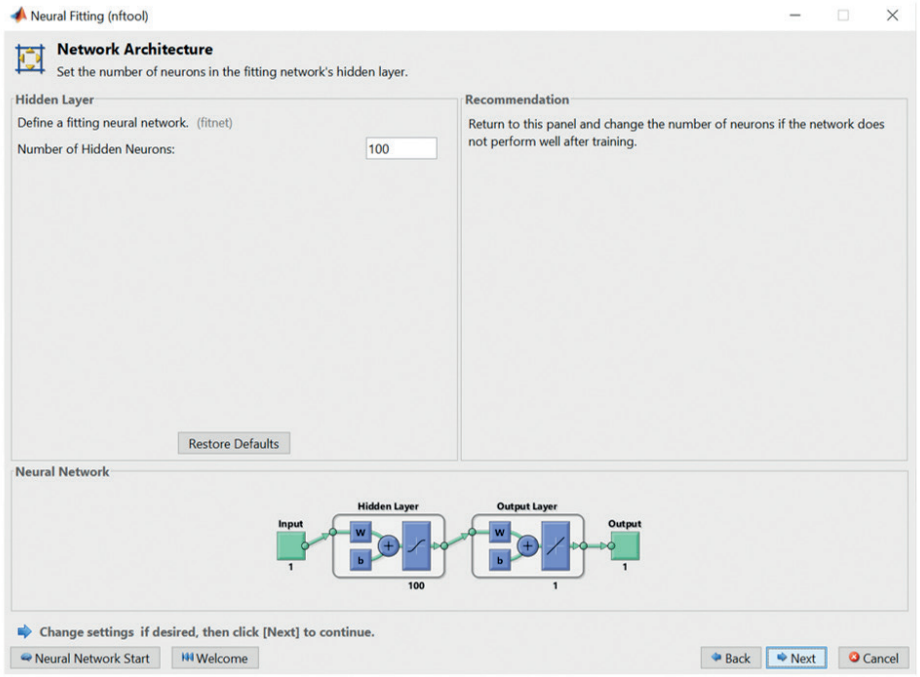
Şekil 7. Giriş ve hedef değerleri yükleme yöntemi

Adım 4: Verilerin eğitim, doğrulama ve test için üç gruba bölünmesi gereken oranı giriyoruz. Bizim durumumuzda, verilerin %70'i ağın eğitimi için, %15'i ağın öğrenme sürecinde doğrulama için ve %15'i test için kullanılır (bkz. Şekil 8).



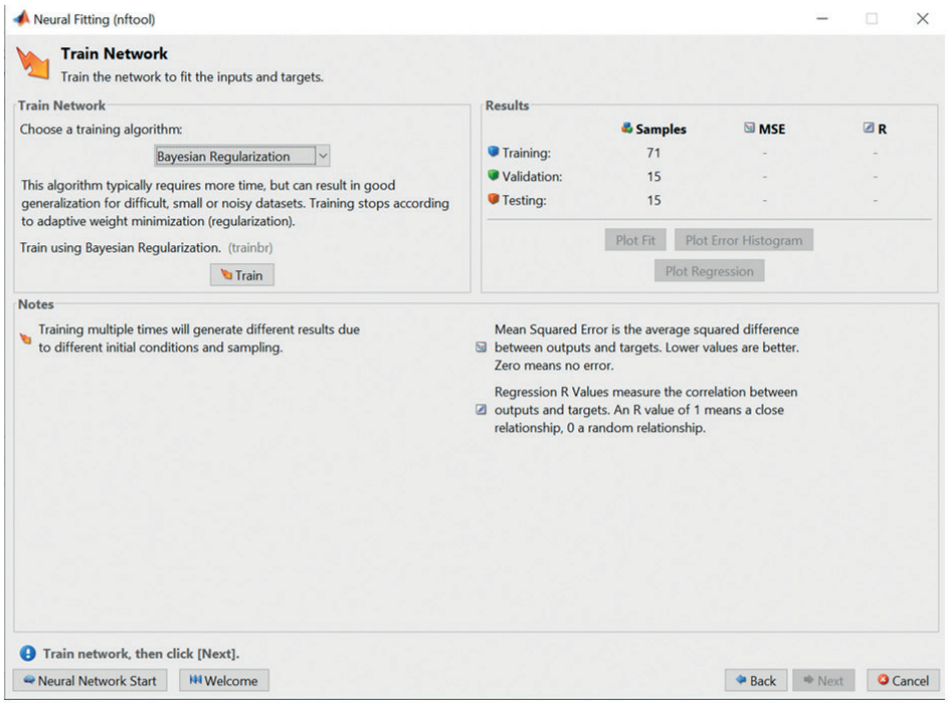
Şekil 8. Çözülen görevde veri örneklerinin dağıtım yöntemi

Adım 5: Ağ mimarisini tasarlayacağız. Ağımızın bir girişi ve bir hedefi olacak şekilde ağın giriş ve çıkış sayısı, giriş ve çıkış verilerine göre otomatik olarak ayarlandı (bkz. Şekil 9). Çıkış katmanında yalnızca bir nöron bulunur çünkü ağdan tek bir çıkışımız vardır. Çıkış katmanındaki nöron sayısı da otomatik olarak ayarlanır. Bizim durumumuzda Neutral Net Fitting uygulamasını kullandığımız için tek bir gizli katmanımız var. Gizli katmandaki nöron sayısını 100 olarak ayarladık. Eğer ağ, girdiler ve hedefler arasındaki ilişkiyi yeterince doğru öğrenemezse ağ mimarisini ayarlamaya geri dönebilir ve gizli nöronların sayısını eğiştirebiliriz.



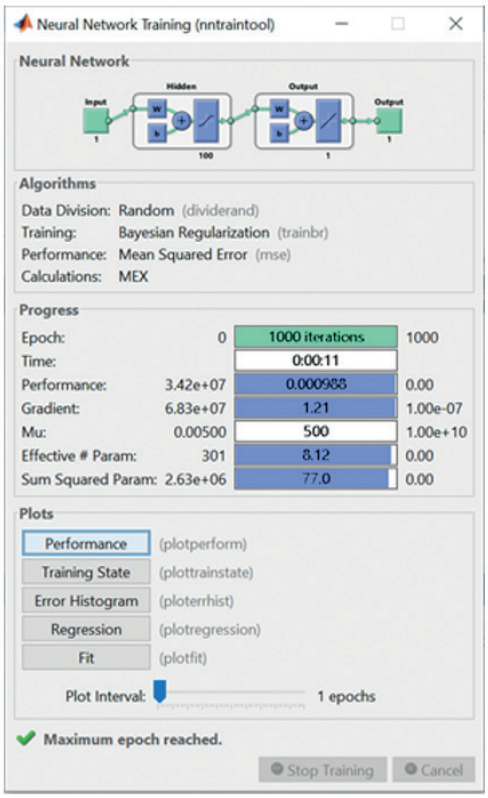
Şekil 9. Görevimiz için ağ mimari tasarımı

Adım 6: Öğrenme algoritmasının seçimi. Üç öğrenme algoritmasından birini seçebiliriz: Levenberg-Marquardt, Bayesian Regularization veya Scaled Conjugate Gradient. Bayesian Regularization algoritmasını seçeceğiz (Şekil 10'a bakınız).



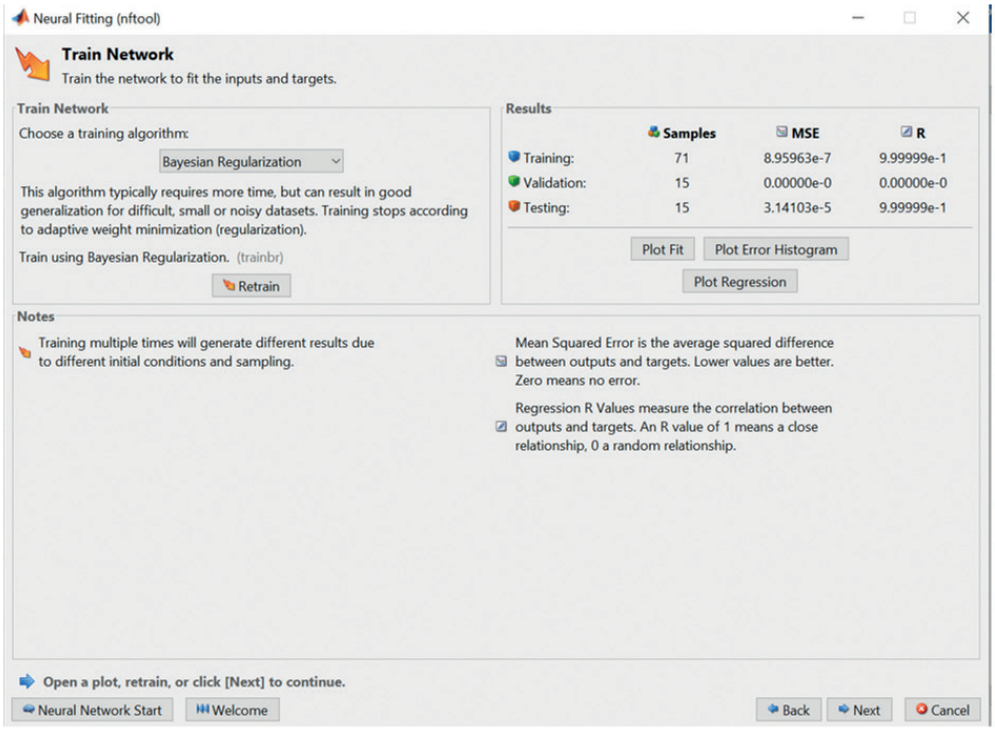
Şekil 10. Öğrenme algoritmasının seçimi

Adım 7: "Train" butonuna basarak ağı öğrenme işlemine başlıyoruz. Öğrenme epok sayısı 1000 olarak ayarlanmıştır ve öğrenme sürecini Şekil 11'de gösterildiği gibi takip edebiliriz.



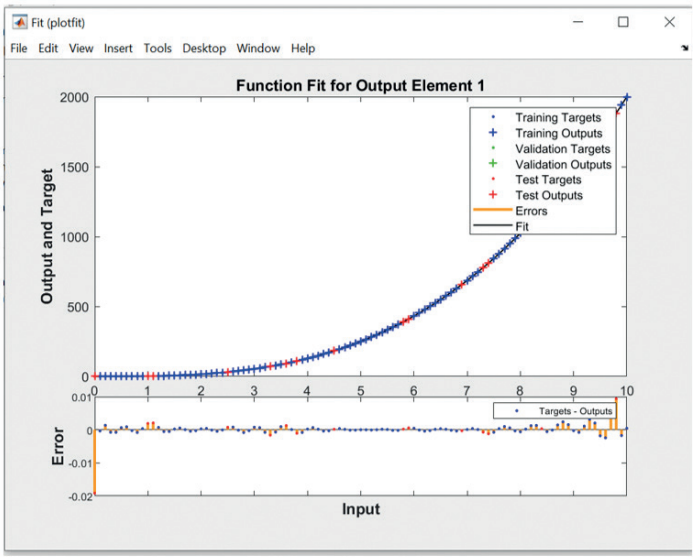
Şekil 11. Ağ öğrenme süreci

Öğrenme doğruluğu MSE ve R kullanılarak ifade edilir (bkz. Şekil 12). Ortalama kare hatası (MSE), eğitim süreci sonrasındaki ağ çıktıları ile eğitim süreci öncesindeki hedefler arasındaki ortalama karesel farktır. Amaç hataların en küçük değerlerini elde etmektir. Sıfır değeri hata olmadığı anlamına gelir. Ağın özelliğimizi önemsiz bir hata olan $8{,}95*10^{-7}$ MSE hatasıyla öğrendiğini görüyoruz. Ağı test etme ağın 3.14*10-5 'lik düşük bir MSE hatasıyla doğru öğrendiğini doğruladı. Regresyon (R), çıktılar ve hedefler arasında ölçülen korelasyonu ifade eder. 1'lik bir R değeri, yakın bir korelasyon anlamına gelir ve 0, korelasyonun olmadığı veya rastgele bir ilişkinin olduğu anlamına gelir. Ağın eğitilmesinden ve ağın test edilmesinden sonra korelasyonların hesaplanması 1 değerine ulaştı; bu da ağın, ağ girdileri ve çıktıları arasındaki ilişkiyi çok iyi öğrendiğini doğrular.



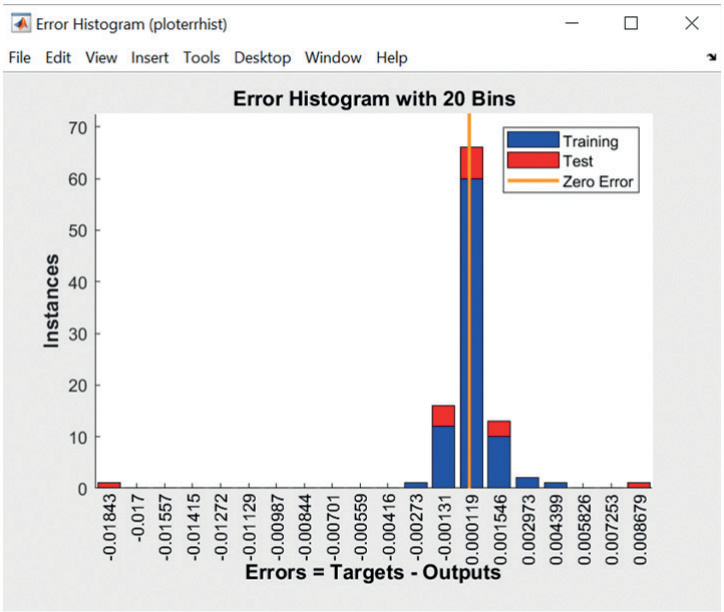
Şekil 12. Ağ öğrenme ve test sürecinden sonraki hatalar ve korelasyonlar

Adım 8: Ağı öğrenme ve test etmede elde edilen doğruluğun yeterli olduğu ve ağın öğrenme sürecinin sona erdiği sonucuna varabiliriz. Bu nedenle lot Fit, Plot Error Histogram ve Plot Regresion butonlarına art arda basarsak öğrenilen fonksiyonun değerlerini ve ilerlemesini daha detaylı görebiliriz (bkz. Şekil 12). Plot Fit'e bastıktan sonra, ağı öğrenip test ettikten sonra girdilere bağlı olarak ağın hedef ve çıktı değerlerinin ilerlemesini biliyoruz (bkz. Şekil 13).



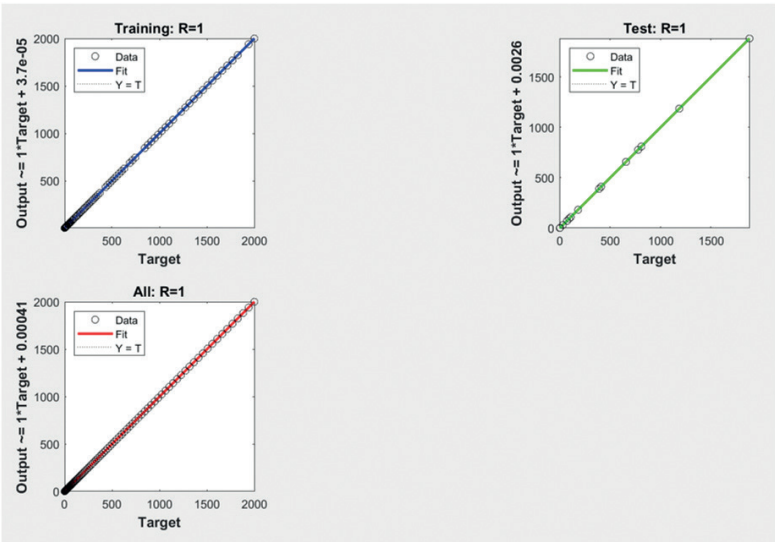
Şekil 13. Ağın öğrenilmesi ve test edilmesi süreci sonrasında ağın hedef ve çıktılarının girdilere bağlı olarak ilerlemesi

Plot Error Histogram'a bastıktan sonra hata değerlerini ve sıklıklarını görebiliriz (bkz. Şekil 14). Bu durumda mutlak hata, belirli bir ağ girişine ilişkin hedef değer ile ağ çıkışı arasındaki farktır. En sık 0.000119 hatasının oluştuğunu görüyoruz.



Şekil 14. Mutlak hataların değerleri ve sıklıkları

Plot Regression tuşuna bastıktan sonra eğitim sürecinde, test sürecinde ve her iki süreçte de hedef değerler ile çıktı değerleri arasındaki korelasyon değerlerini görüyoruz (bkz. Şekil 15). Ağın eğitilmesinden ve ağın test edilmesinden sonra korelasyonların hesaplanması 1 değerine ulaştı; bu da ağın, ağ girdileri ve çıktıları arasındaki ilişkiyi çok iyi öğrendiğini doğruladı.



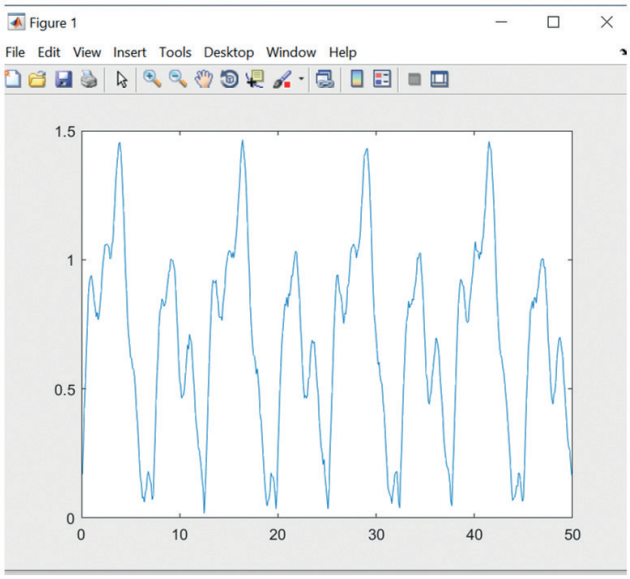
Şekil 15. Eğitim süreci, test süreci ve her iki süreçteki hedef değerler ile çıktı değerleri arasındaki korelasyonlar

Ölçülen değerlerin oluşturulmasıyla sinir ağı uydurma fonksiyonu örneği

Bu örnekte sinir ağında ölçerek elde ettiğimiz değerleri öğrenmeyi amaçlıyoruz. Bu bölümün 2.1 kısmında sunulan metodolojiyi takip ediyoruz.

Adım 1: Veri hazırlama. Data4 dosyasında saklanan 500 ölçülen veri değerimiz var (bkz. Şekil 16). Açıklık sağlamak için, x eksenini 0,1 değerinden 50 değerine kadar 0,1'lik bir adımla çiziyoruz. Aşağıdaki koda bakın:

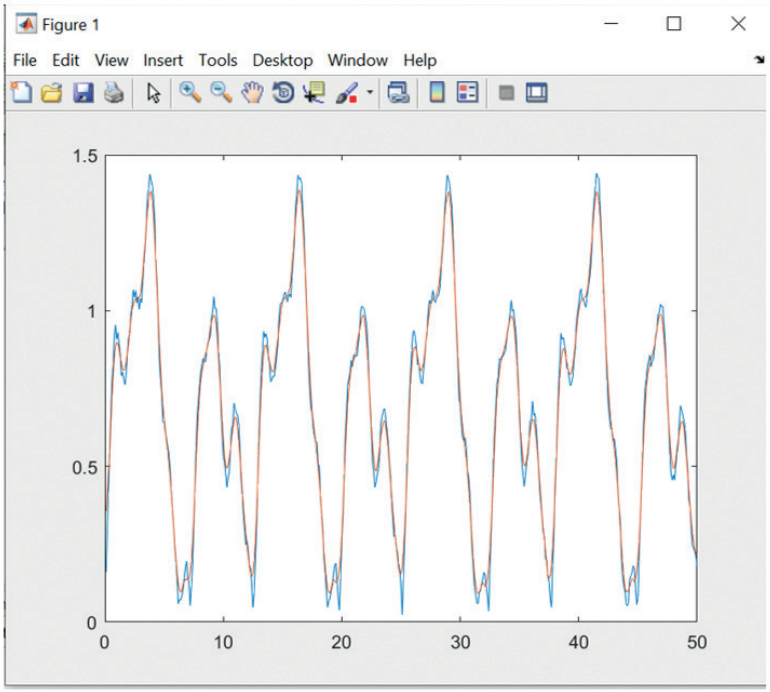
```
x=0.1:0.1:50
save data3 x
load data4 y
plot(x,y)
```



Şekil 16. Ölçülen değerler

Bazen ölçülen veriler gürültülü olabilir ve bu nedenle ayarlanması gerekir [1]. Sinir ağının öğrenebilmesi için verileri yumuşatan hareketli bir araç kullanabiliriz. Hareketli ortalama, kademeli olarak tüm düzeltilmiş değerlerin üzerinden geçer ve mevcut değeri ortalama değerle değiştirir. Pencere, geçerli değerden ve geçerli değerden önceki ve sonraki belirli sayıda değerden oluşur. Şimdi data4.mat dosyasında saklanan ölçülen verilerimizi nasıl düzelteceğimizi göstereceğiz. Dokuz değer genişliğinde bir pencereye sahip hareketli bir ortalama kullanarak ölçülen verileri yumuşatarak listemizi 2 komutlarla genişleteceğiz. M vektöründeki değiştirilen verileri data5.mat dosyasına kaydedeceğiz ve ağı öğrenirken (aşağıdaki koda bakın) ve Şekil 17'de bunları hedef olarak kullanacağız.

```
x=0.1:0.1:50
save data3 x
load data4 y
plot(x,y)
m = movmean(y,7)
plot(x,y,x,m)
save data5 m
```



Şekil 17. Ölçülen değerler mavi renkte ve düzeltilmiş veriler kırmızı renkte

Adım 2: Matlab ortamında Neutral Net Fitting uygulamasını seçiyoruz.

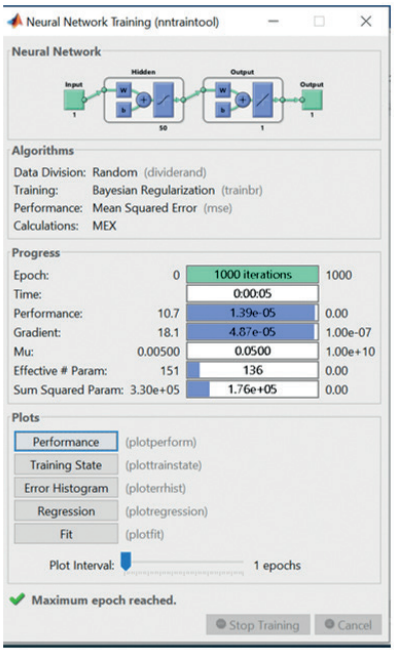
Adım 3: Verileri uygulamaya yüklüyoruz. data3.mat dosyası giriş verilerini içerir ve data5.mat dosyası hedefleri içerir.

Adım 4: Oranı önceki örnekteki gibi bırakıyoruz.

Adım 5: Ağ mimarisini tasarlayacağız. Gizli katmanda 50 adet nöron seçiyoruz.

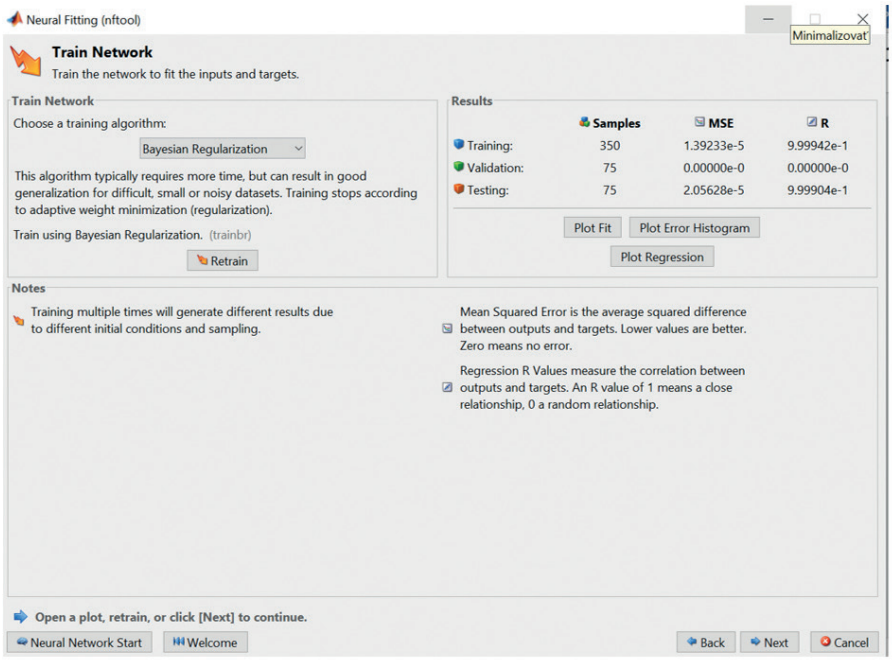
Adım 6: Bayesian Regularization öğrenme algoritmasını seçiyoruz.

Adım 7: Öğrenme sürecine başlıyoruz (bkz. Şekil 18).



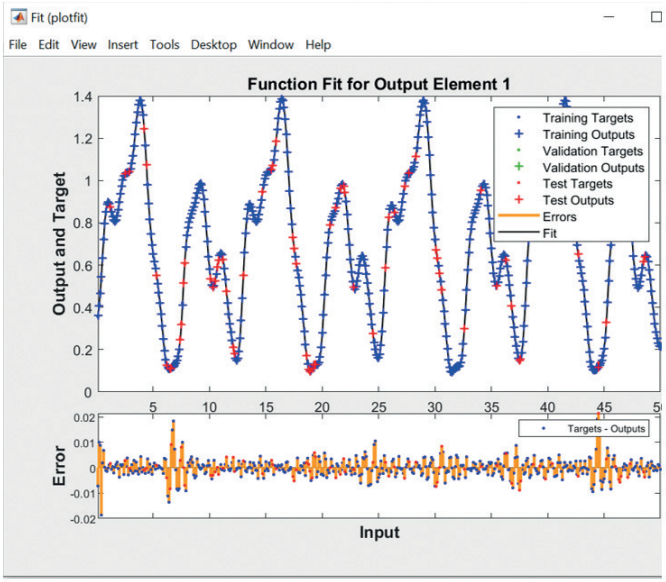
Şekil 18. Ağ öğrenme süreci

Adım 8: Ağın öğrenme sonuçlarına daha yakından bakalım (Şekil 19). 1,39x10^{-5} öğrenme hatası ve 1,39x10^{-5} ağ test hatasına dayanarak ağın doğru değerleri öğrendiği sonucuna varıyoruz.



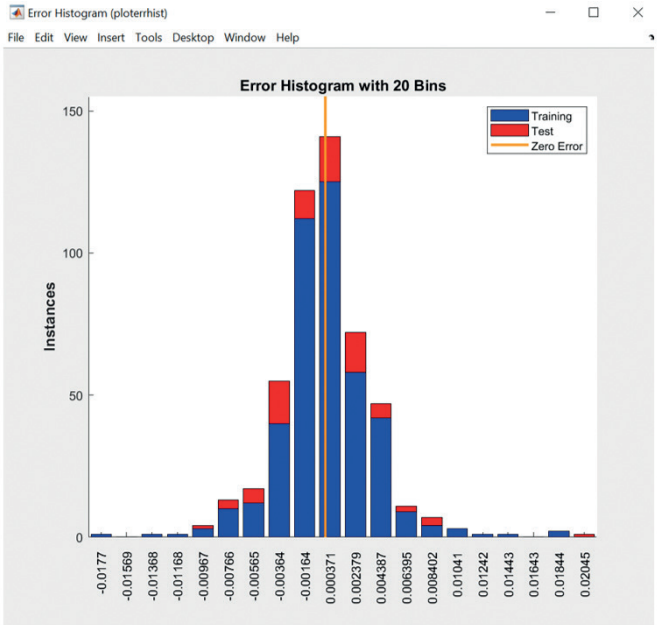
Şekil 19. Öğrenme hatalarını görüntüleme

Şekil 20'de ağın öğrenilmesi ve test edilmesi sonrasında girdilere bağlı olarak ağın hedef ve çıktı değerlerinin ilerlemesini görebiliriz. Öğrenilen değerler hedef değerlerle örtüşür. Görüntünün alt kısmında minimum düzeyde görüntülenen hataları görüyoruz.



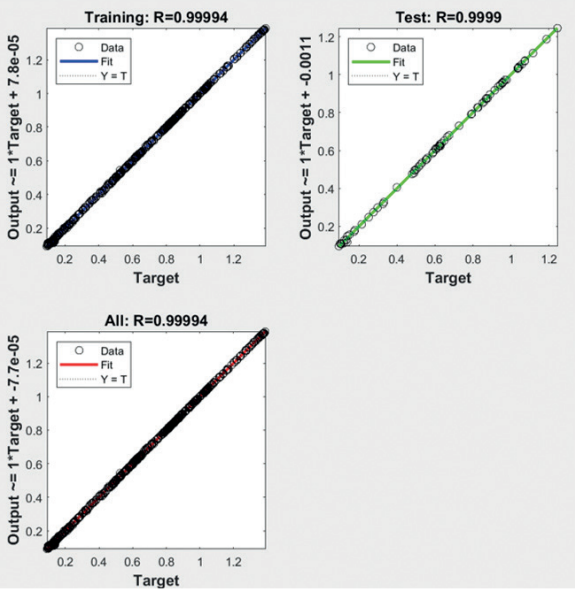
Şekil 20. Ağ öğrenme sonuçları - ilerleme hedefleri ve öğrenilen değerler

Şekil 21'deki histogramda hataların boyutunu ve sıklığını görebiliyoruz, bu da hataların minimum düzeyde olduğunu bir kez daha kanıtlıyor.



Şekil 21. Ağ öğreniminin sonuçları - histogram - boyut ve hataların sıklığı

Eğitilmiş, test edilmiş ve tüm girdiler açısından 1 değerine ulaşan gösterilen regresyonlar (bkz. Şekil 22), çok iyi eğitilmiş bir ağa sahip olduğumuzu göstermektedir. Genel olarak, ağı öğrenme ve test etmede elde edilen doğruluğun yeterli olduğu ve ağın öğrenme sürecinin sona erdiği sonucuna varabiliriz.



Şekil 22. Ağ öğrenme sonuçları - regresyon hesaplaması

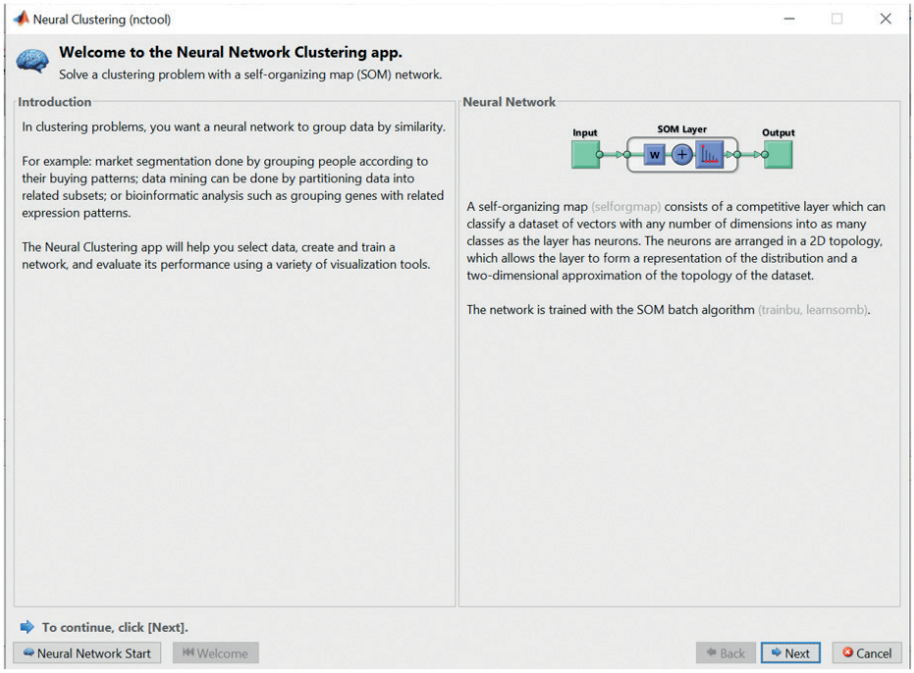
Kendi kendini organize eden harita sinir ağıyla veri kümeleme örneği

Bu örnekte, iyi bilinen Iris çiçeklerini sınıflandırma görevini çözüyoruz. IRIS veri setini kullanacağız ve Matlab ortamının parçası olan çözülmüş bir örneği anlatacağız. İris çiçekleri 4 parametre kullanılarak tanımlanabilmekte olup veri setindeki değerler sepal length (çanak yaprak uzunluğu), sepat width (çanak yaprak genişliği), petal length (taç yaprak uzunluğu) ve petal width (taç yaprak genişliği) santimetre cinsinden verilmektedir. Bu nedenle her çiçek 4 elementle karakterize edilir. Görevimiz, iris çiçeği türlerini, benzer türlerin birbirine yakın bir grupta yer alacağı şekilde sınıflara ayıran, kendi kendini organize eden bir harita sinir ağı oluşturmaktır. Harita, örneklerin benzerliğine göre oluşturulur ve öğrenilen sinir ağı, bilinmeyen örnekleri bile sınıflandırabilir [4].

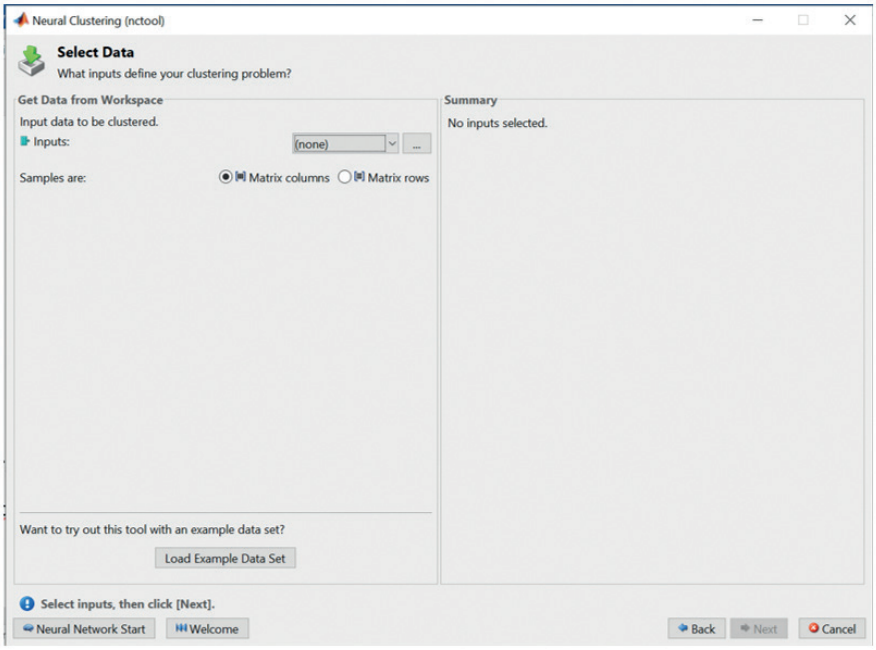
Matlab grafik ortamında sinir ağı uygulama metodolojisine göre ilerliyoruz.

Adım 1: Bu adımı atlıyoruz. Matlab'da bulunan hazır bir veri setini kullanacağız ve bu veri setini 3. adımda detaylı olarak anlatacağız.

Adım 2: Matlab ortamında, APPS sekmesinden Machine Learning kategorisinden uygun uygulamayı seçiniz, Neutral Net Clustering uygulamasını seçiniz ve uygulamayı başlatınız. Bu uygulama, kendi kendini organize eden bir harita sinir ağı oluşturmamıza yardımcı olacaktır. Bkz. Şekil 23

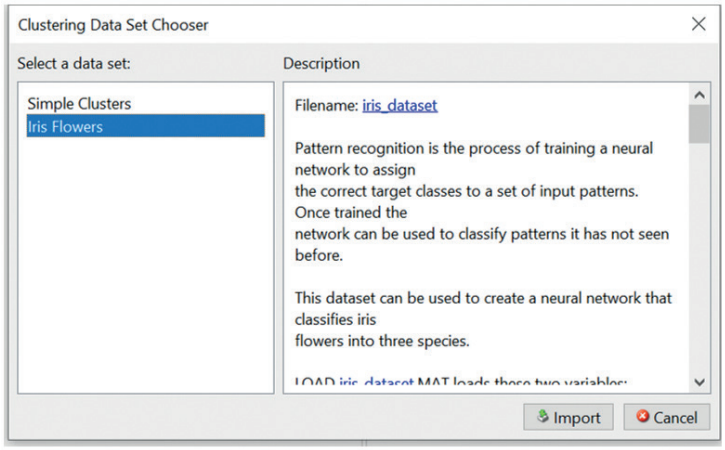


Şekil 23. Sinir Ağı Kümeleme Uygulaması

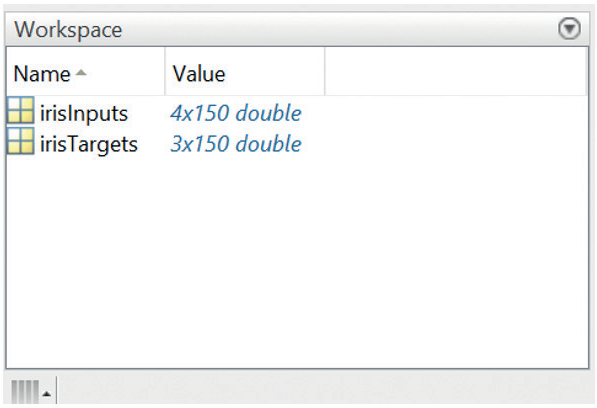


Şekil 24. Hazırlanan veri setinin yüklenmesi

Adım 3: Veri setlerini yüklüyoruz. Bkz. Şekil 24. Hazırlanan IRIS veri setini kullanmak istediğimiz için (bkz. Ek A), Load Example Data Set (Örnek Veri Setini Yükle)'e basıp Iris Flowers'ı seçip verileri içe aktarıyoruz. Bkz. Şekil 25.



Şekil 25. İris çiçekleri veri kümesinin içe aktarılması.



Şekil 26. İris çiçekleri veri kümesi yüklendikten sonra Matlab çalışma alanı.

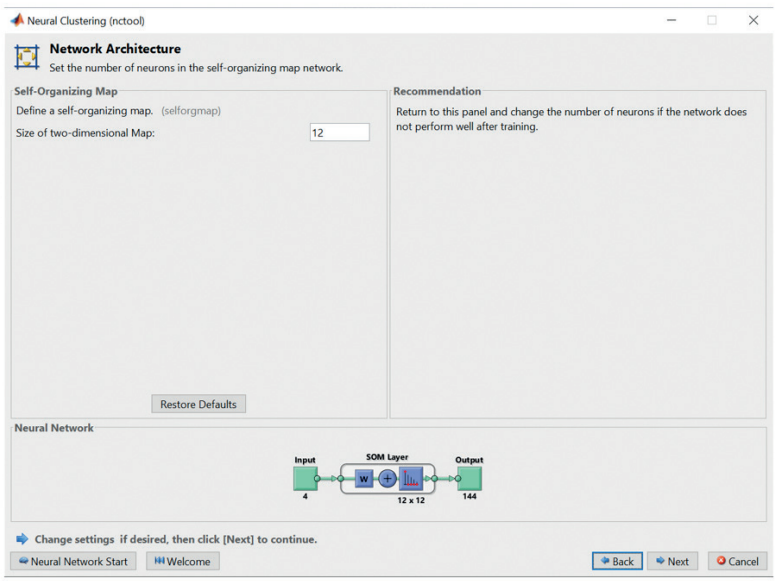
Veri seti yüklendikten sonra irisInputs ve irisTargets matrisleri Matlab çalışma alanı penceresinde görüntülenir, bkz. Şekil 26. İrisInputs matrisinin 4 satır x 150 sütun boyutunda olduğunu görebiliriz. Dört parametre bir çiçeği tanımlar; bu nedenle bir sütun bir çiçek örneğini temsil eder. Giriş veri kümesi 150 çiçek örneği içerir. İrisTargets matrisi, her giriş örneğinin 3 sınıftan birine sınıflandırılmasını belirtir. Verilerin daha iyi anlaşılabilmesi için her iki matrisi de komut penceresine ayrı ayrı yazacağız. Veri örnekleri aşağıdaki kodlarda görülebilir:

```
>> irisInputs
irisInputs =
    Columns 1 through 11
        5.1000 4.9000 4.7000 4.6000 5.0000 5.4000 4.6000 5.0000 4.4000 4.9000 5.4000
        3.5000 3.0000 3.2000 3.1000 3.6000 3.9000 3.4000 3.4000 2.9000 3.1000 3.7000
        1.4000 1.4000 1.3000 1.5000 1.4000 1.7000 1.4000 1.5000 1.4000 1.5000 1.5000
        0.2000 0.2000 0.2000 0.2000 0.2000 0.4000 0.3000 0.2000 0.2000 0.1000 0.2000
```

```
>> irisInputs

irisTargets =

Columns 1 through 18

     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0

Columns 19 through 36

     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0

Columns 37 through 54

     1     1     1     1     1     1     1     1     1     1     1     1     1     1     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     1     1     1
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

Adım 4: Örneklerin eğitim ve test olarak bölünmesi önceden beirlenmiştir ve bu nedenle bu bilgiyi uygulamaya girmek gerekli değildir.

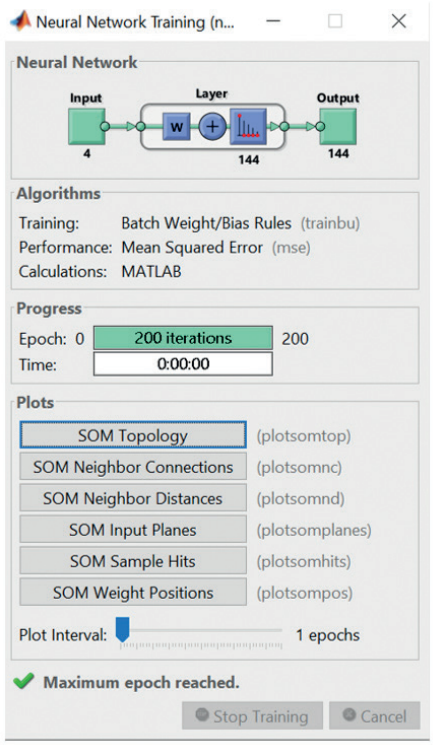
Adım 5: Ağ mimarisini tasarlayacağız. Bu durumda SOM katmanındaki nöron sayısının girilmesi gerekmektedir. Bir katman, iki boyutlu kare bir diziyi temsil eder. Dolayısıyla dizi boyutunu 12 olarak belirttiğimizde 12x12 elemanlı iki boyutlu bir dizi oluşturulur. Bkz. Şekil 27. O zaman ağ çıkışındaki harita 12x12 yani 144 eleman olacaktır. Önceden tanımlanmış çıktı haritası topolojisi altıgendir.



Şekil 27. Ağ mimarisi tasarımı

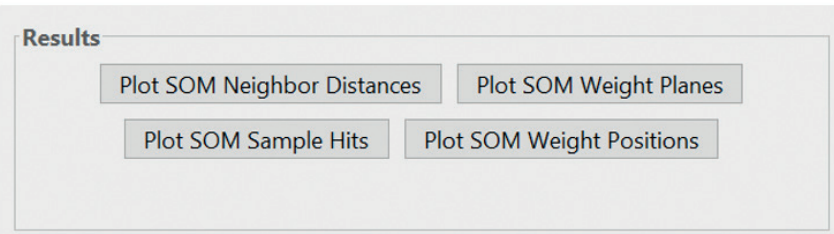
Adım 6: Öğrenme algoritmasının seçimi. Uygulamanın önceden tanımlanmış bir toplu SOM algoritması olduğundan bu adımı atlıyoruz.

Adım 7: "Train" butonuna basarak ağın öğrenme sürecini başlatıyoruz. Öğrenme epok sayısı 200 olarak ayarlanmıştır ve öğrenme sürecini Şekil 28'de gösterildiği gibi takip edebiliriz.

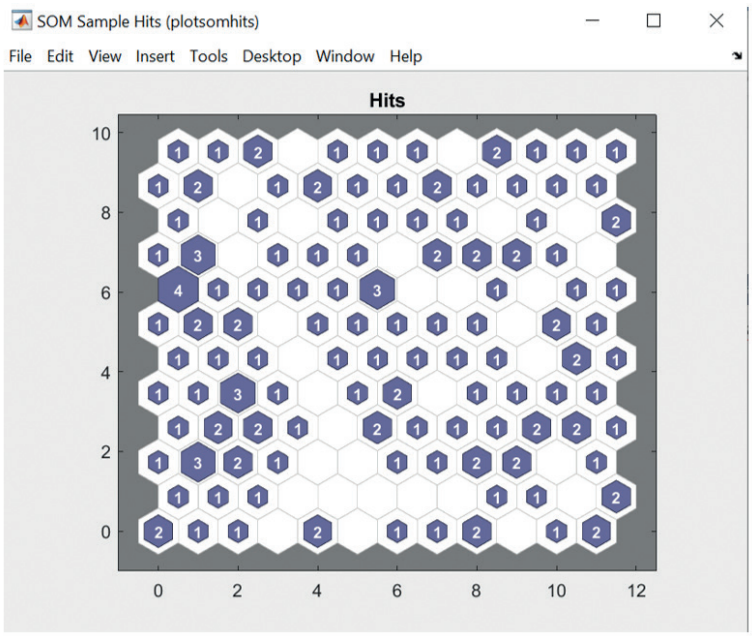


Şekil 28. Ağ öğrenme süreci

Adım 8: Dört düğmeyi kullanarak ağı öğrenmenin sonuçlarını görüntülüyoruz (bkz. Şekil 29). İlk sonuç, her çiçek için sınıfların sınıflandırılmasıdır ve *Plot som sample hits* (Şekil 30'a bakınız) her sınıftaki çiçeklerin sayısını gösterir. Daha yüksek değerlere sahip nöron alanları, benzer şekilde sıklıkla temsil edilen çiçek sınıflarını temsil eder. Tersine, küçük değerlere sahip alanlar, çiçeklerin daha az bol olduğu anlamına gelir.

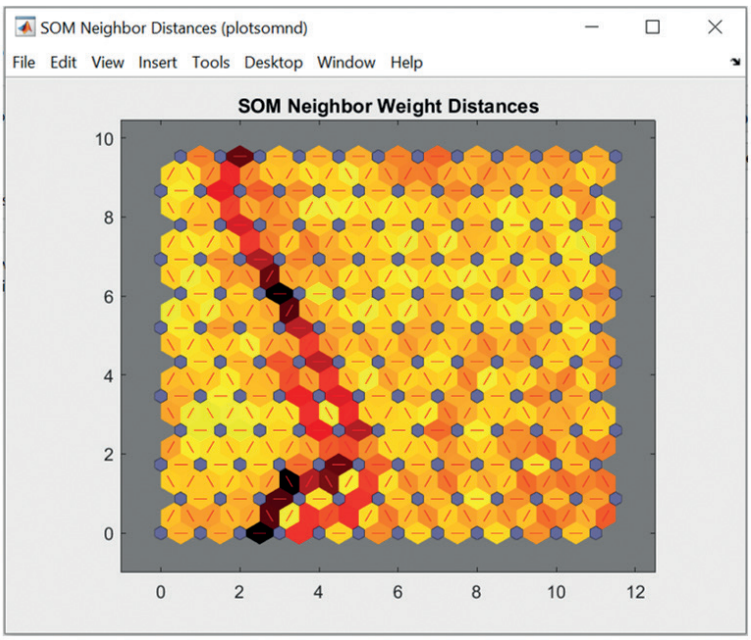


Şekil 29. Ağ öğrenme sonuçları



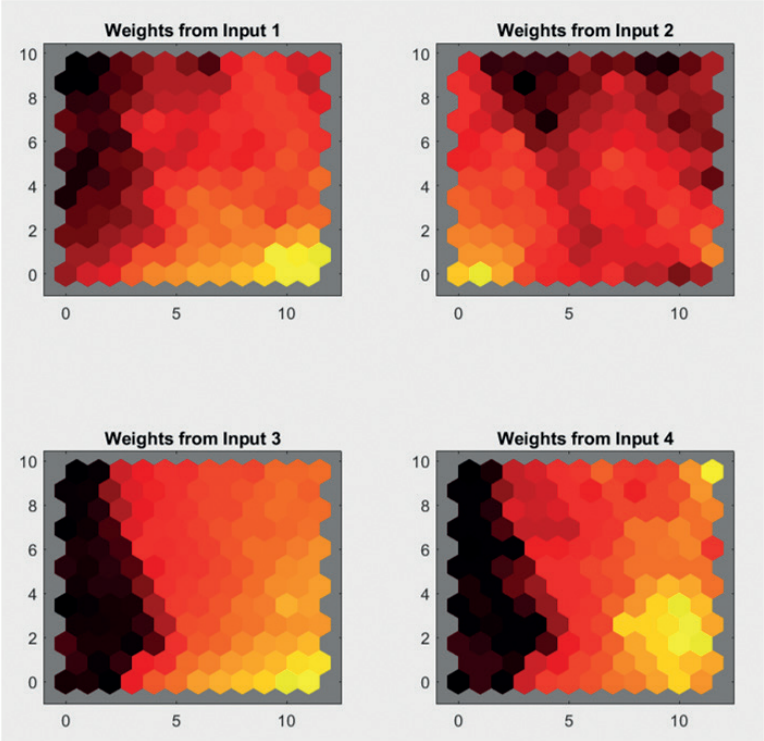
Şekil 30. Ağ öğrenme sonuçları - her sınıftaki çiçek sayısı

Plot SOM Neighbor Distances (SOM Komşu Mesafeleri Grafiği) 'ni kullanarak gösterdiğimiz sonuç, bir nöron sınıfının komşularına olan Öklid mesafesini ifade eder. Parlak bağlantılar oluşturan nöron grupları, girdi kümesindeki çiçeklerin yüksek düzeyde benzerliği anlamına gelir. Tersine, koyu-açık bağlantılar daha az çiçekli veya çiçeksiz ekili alanları temsil eder. Bkz. Şekil 31. Koyu kenarlıklar (birleşim yerleri), giriş alanının geniş alanlarını ayırır ve ayrı alanlardaki çiçeklerin farklı özelliklere sahip olduğunu gösterir.



Şekil 31. Ağ öğreniminin sonuçları - giriş alanındaki çiçeklerin bolluğu ve sınıfları

Çiçeklerin dört giriş özelliği açısından ağın ortaya çıkan ağırlıkları, Plot SOM ağırlık düzlemleri kullanılarak görüntülenecektir. Bkz. Şekil 32.



Şekil 32. Ağ öğrenme sonuçları - bireysel ağ girişleri için ağırlık haritaları

Ağırlıklar, her girişi ağın 144 çıkış nöronunun her birine bağlar. Koyu renkler daha büyük ağırlıkları temsil eder. Haritada aynı renge sahip olan girişler güçlü bir şekilde ilişkilidir.

BÖLÜM 12

EKLER

Bu bölüm, el kitabının ana metninde sunulan çalışmalara ilişkin ekleri içermektedir. Beş ek, özellikle el kitabı ve el kitabının kullanılabileceği dersle ilgili çeşitli verileri ve alıştırma bilgilerini içerir:

- **Ek A,** bölüm 3 – 8'deki örneklerde kullanılan İris veri kümesini açıklamaktadır.
- **Ek B,** bölüm 7'de sunulan sorunların çözüm örneklerini içermektedir.
- **Ek C,** veri analizi için kaynak olarak kullanılabilecek seçilmiş hava kirliliği ve iklim değişikliği veri kümelerinin sunulmasına odaklanmaktadır.
- **Ek D'de** hava kirliliğinin insan sağlığı üzerindeki etkisi açıklanmaktadır.
- **Ek E,** el kitabının kullanılabileceği ders için önerilen müfredatı içermektedir.

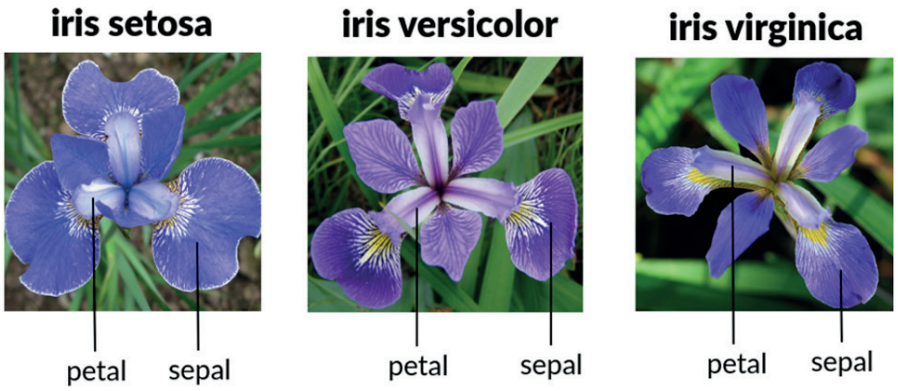
A-İRİS VERİ SETİNİN KISA AÇIKLAMASI

El kitabının bu bölümü Slovakya'nın Banská Bystrica şehrindeki Matej Bel Üniversitesi Doğa Bilimleri Fakültesi Bilgisayar Bilimleri Bölümü'nden Alžbeta Michalíková ve Adam Dudáš tarafından yazılmıştır.

Iris veri seti, veri analizinde en sık kullanılan veri setlerinden biridir ve veri işleme için tahmin modelleri ve ayarlama algoritmalarıyla çalışır.

Edger Andersen, ilk olarak Fisher, R.A."The use of multiple Measurements in Taxonomic Problems" Annual Eugenics, 1936 yayınında veri analizi bağlamında sunulan bu veri setini oluşturdu. Bu veri seti, İris çiçeğinin **150 bireyi** üzerinde ölçülen **beş özellikten** oluşur (veri seti 150x5 boyutundadır). Çiçek iki döngüde yapılandırılmış altı yapraktan oluşur:

- sepaller (çanak yapraklar)çiçeğin iç döngüsünü oluşturur,
- petaller (taç yapraklar) çiçeğin dış döngüsünü oluşturur.



Iris veri seti, her yaprak türü için (genişlik ve uzunluk) iki değer ölçülerek derlenir ve bu da **dört sayısal özellik** oluşturur:

- çanak yaprağın uzunluğu (sepal length) ve çanak yaprağın genişliği (sepal width), santimetre veya milimetre cinsinden ölçülür
- taç yaprağın uzunluğu (petal length) ve taç yaprağın genişliği (petal width) santimetre veya milimetre cinsinden ölçülür.

Veri kümesinin beşinci özelliği, veri kümesinin varlıklarını üç sınıfa bölen kategorik değer **sınıfı** veya bazen **türdür**:

- iris setosa,
- iris versicolor,
- iris virginica.

Bu sınıfların her biri veri kümesinde **50 birim** tarafından eşit olarak temsil edilir. Her Iris veri kümesi sınıfından bir örnek birim örneği sunuyoruz:

Birim	Çanak yaprağın uzunluğu	Çanak yaprağın genişliği	Taç yaprağın uzunluğu	Taç yaprağın genişliği	Türü
1	5.1	3.5	1.4	0.2	setosa
2	7.0	3.2	4.7	1.7	versicolor
3	6.3	3.3	6.0	2.5	virginica

İris veri kümesiyle çalışma

İris veri seti o kadar standartlaştırılmıştır ki çoğu veri işleme ve analiz aracı bu veri setini yüklemek için kullanılabilecek dahili bir komuta sahiptir.

Örneğin R dilinde veri dosyasının adı yerine sadece iris kullanıyoruz.

Örnek: Yüklenen veri kümesinin adını R'ye yazarak, veri kümesinin tüm özniteliklerini ve varlıklarını içeren konsol çıktısı alıyoruz. İris veri seti durumunda iris yazabiliriz (veri setini yüklemeye gerek kalmadan).

```
> iris
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
```

Bu şekilde iris veri seti bulunmayan bir araçla çalışılması durumunda, veri setini örneğin aşağıdaki adresten ücretsiz olarak indirmek mümkündür:

https: //archive.ics.uci.edu/ml/datasets/iris

B-BULANIK SINIFLANDIRMA SORULARINA ÇÖZÜMLER

El kitabının bu bölümü Slovakya'nın Banská Bystrica şehrindeki Matej Bel Üniversitesi Doğa Bilimleri Fakültesi Bilgisayar Bilimleri Bölümü'nden Alžbeta Michalíková tarafından yazılmıştır.

Sugeno yöntemini kullanarak, Iris veri kümesindeki verileri uygun sayıda sınıfa sınıflandırın.

Çözüm:

Aşağıdaki soruları cevaplayın:

1. İris veri kümesinde kaç tane **giriş değişkeni** bulunmaktadır?
   Veri kümesinde dört giriş değişkeni bulunmaktadır.

2. **Giriş değişkenlerini tanımlamak için** ne kullanacağız?
   Giriş değişkenlerini tanımlamak için bulanık üyelik fonksiyonlarını kullanacağız.

3. Hangi tür bulanık **üyelik fonksiyonlarını** kullanacağız?
   Trapezoidal (yamuk) üyelik fonksiyonlarını kullanacağız.

4. **Çıktı** ne olacak?
   Çıktı, bireysel İris çiçeklerinin ait olduğu (tablonun satırları/nesneleri) özel sınıf olacaktır.

5. **Çıktı değişkenlerini tanımlamak** için ne kullanacağız?
   Çıktı değişkenlerini tanımlamak için sabit fonksiyonlarını (sabitler) kullanacağız.

6. Hangi tür kuralları kullanacağız?
   Sugeno EĞER-İSE kurallarını kullanacağız.

7. Bir kuralın örneğini yazın!
   Eğer Girdi1(input1) ve Girdi2 (input2) küçükse, Girdi3 (input3) orta ise ve Girdi4 (input4) yüksekse, o zaman Çıktı (output) sınıf1'dir (veya Iris_Setosa).

Bu verilerden giriş değişkenleri parametrelerinin değerlerini belirleyin ve bunları aşağıdaki tablolara doldurun.

Tablo B.1: Giriş değişkenlerinin parametreleri

INPUT 1:

Ad	Parametre
Evren	[40 80]
Kırmızı	[-20 -10 48 59]
Mavi	[48 55 67 71]
Yeşil	[55 71 81 90]

INPUT 1:

Ad	Parametre
Evren	[20 45]
Kırmızı	[22 39 46 50]
Mavi	[0 10 24 35]
Yeşil	[21 18 34 39]

INPUT 3:

Ad	Parametre
Evren	[10 70]
Kırmızı	[0 5 19 28]
Mavi	[26 30 44 52]
Yeşil	[44 53 75 80]

INPUT 4:

Ad	Parametre
Evren	[0 25]
Kırmızı	[-10 -5 6 10]
Mavi	[6 10 13 19]
Yeşil	[13 19 30 35]

Çıkış parametrelerinin değerlerini belirleyin. Çıkış dilsel değişkeni için sabit fonksiyonları kullanıyorsak, aşağıdaki tabloyu doğru değerlerle doldurun:

Tablo B.2: Çıkış değişkenlerinin parametreleri

OUTPUT:	
Ad	**Parametre**
Evren	**[1 3]**
Kırmızı	1
Mavi	2
Yeşil	3

Kural sayısını tasarlayalım ve doğru biçimde yazın:

Kurallar:

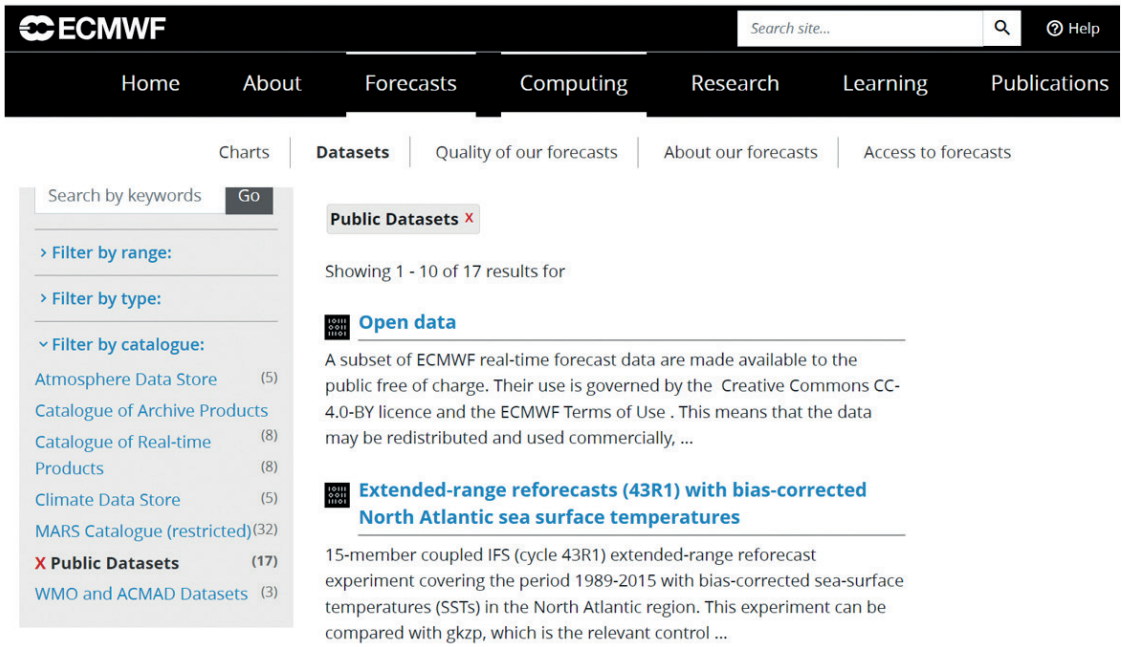
1. Eğer Input1 Kırmızı ve Input2 Kırmızı ise ve Input3 Kırmızı ve Input4 Kırmızı ise, o zaman Output Kırmızı'dır.
2. Eğer Input1 Mavi ve Input2 Mavi ise ve Input3 Mavi ve Input4 Mavi ise, o zaman Output Mavi'dir.
3. Eğer Input1 Yeşil ve Input2 Yeşil ise ve Input3 Yeşil ise ve Input4 Yeşil ise, o zaman Output Yeşil'dir.

C-İKLİM DEĞİŞİKLİĞİ VERİ SETLERİNİN KISA AÇIKLAMASI

El kitabının bu bölümü, Romanya, Craiova Üniversitesi, Fen Fakültesi Fizik Bölümü'nden Mihaela Tinca Udristioiu ve İktisat ve İşletme Fakültesi, Yönetim, Pazarlama ve İşletme Bölümü'nden Silvia Puiu tarafından yazılmıştır.

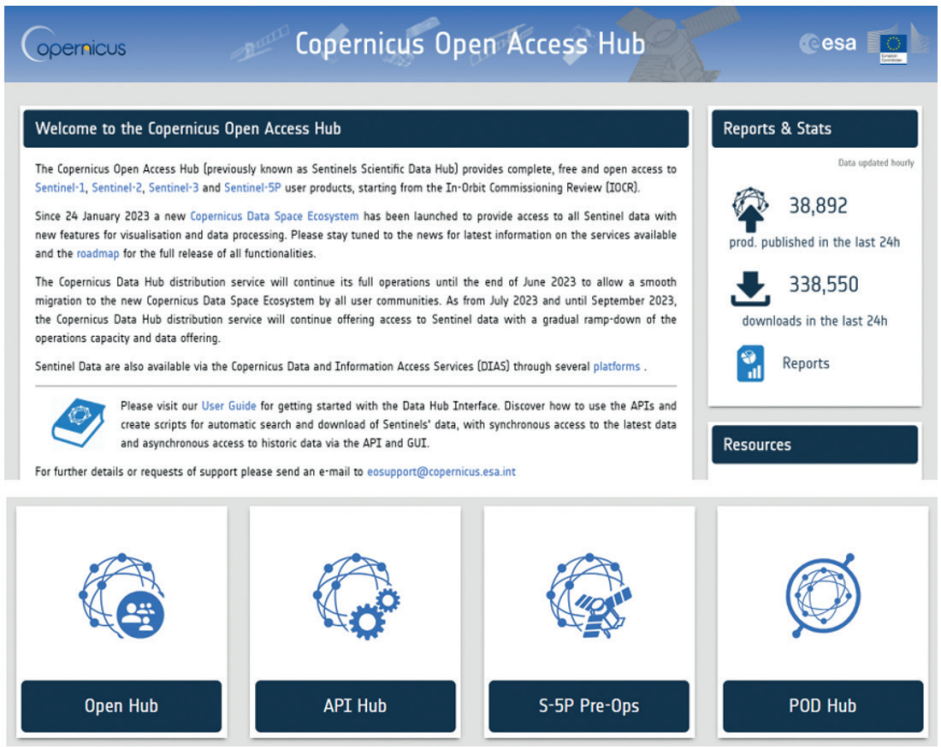
Araştırmacılar ve kullanıcılar genellikle bilgiye açık kaynak erişimine sahip olduklarında araştırma daha kolay ve daha hızlı ilerleyebilir. İnternet parmaklarımızın ucundayken doğru, güncel ve güvenilir bilgi kaynaklarını nerede arayacağımızı bilmemiz gerekiyor. Tüm bu nedenler veritabanlarının rolünün neden bu kadar önemli olduğunu vurgulamaktadır. Veriler yapılandırılmıştır ve genellikle araştırmacının veya kullanıcının ihtiyaçlarına göre kolayca dönüştürülüp işlenebilecek şekildedir. Avrupa Komisyonu iklim değişikliğiyle ilgili birkaç açık kaynak veri oluşturdu. Veri setlerini indirmek kolaydır; veri kümelerini işlemek ve analiz etmek (keşif ve tahmine dayalı analiz) ve matematiksel modeller oluşturmak için farklı algoritmalar kullanmak daha zordur. Copernicus, Avrupa İklim Değerlendirmesi ve Veri Seti, Climate Explorer ve Indecis bunlardan yalnızca birkaçıdır. İklim değişikliğini izlemek ve hava durumunu tahmin etmek, iklimin çeşitli parametrelere duyarlılığını gözlemlemek, farklı senaryolar oluşturmak ve bazı süreçlerin kısa ve uzun vadedeki gelişimini görmek için verilere ihtiyacımız var. Aşağıda yazarlar bazı veritabanlarını kısaca sunacaklar.

***Avrupa Orta Vadeli Hava Tahminleri Merkezi (ECMWF),** günlük operasyonel veri özümleme ve izleme faaliyetlerinde yaklaşık 90 uydu cihazından gelen verileri işler. Entegre Tahmin Sisteminde günlük olarak çoğu uydu ölçümü olmak üzere yaklaşık 60 milyon kalite kontrollü gözlem mevcuttur. ECMWF ayrıca yüzey bazlı raporlar ve uçak raporları dahil olmak üzere uydu dışı kaynaklardan elde edilen tüm gözlemlerden de yararlanır.*



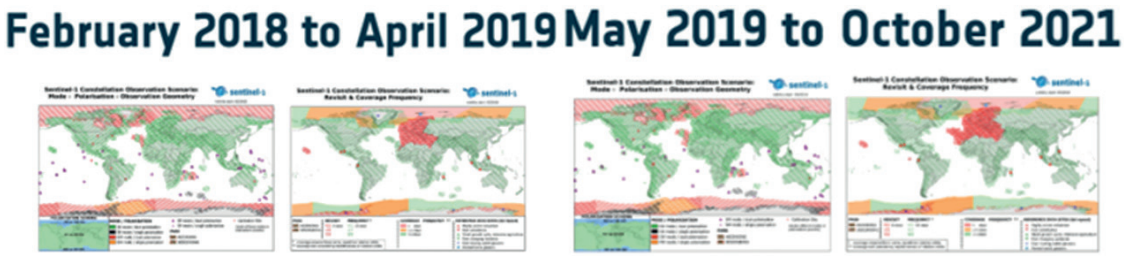
Şekil C.1. ECMWF'den Kamu Veri Seti bölümünün ekran görüntüsü (kaynak: https://www.ecmwf.int/en/forecasts/datasets/search)

Copernicus, AB Uzay Programı'nın Dünya gözlem bileşenidir. Avrupa Komisyonu (EC) tarfafından yönetilmektedir. EC, Copernicus'u AB Üye Devletleri, Avrupa Uzay Ajansı (ESA), Avrupa Meteorolojik Uyduların Kullanımı Örgütü (EUMETSAT), Avrupa Orta Vadeli Hava Tahminleri Merkezi (ECMWF), Ortak Araştırma Merkezi (JRC), Avrupa Çevre Ajansı (EEA), Avrupa Deniz Güvenliği Ajansı (EMSA), Frontex, SatCen ve Mercator Océan ile ortaklaşa uygulamaktadır. Copernicus, çeşitli kaynaklardan (yeniden analizler, uydu ürünleri, iklim tahminleri) iklim verileri içerir. Copernicus veritabanı, iklim değişikliğinin en sık kullanılan veri kümelerinden biridir ve veri işleme için tahmin modelleri ve ayarlama algoritmalarıyla çalışır. Öne çıkan misyonlara sahip uyduları (SENTINELS 1-6) vardır.



Şekil C.2. Copernicus Açık Erişim Merkezi arayüzünün görüntüsü
(kaynak: https://scihub.copernicus.eu/)

SENTINEL-1, iki kutupsal yörüngeli uyduya sahiptir ve 7 gün 24 saat boyunca kesintisiz olarak çalışır, Hava koşullarından bağımsız olarak görüntü toplamak için radar görüntüleme kullanır.



Şekil C.3. SENTINEL 1 tarafından iki zamanlı aralıklarla sağlanan görüntü
(kaynak: https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-1/observation-scenario/archive)

SENTINEL-2, aynı güneş-senkron yörüngede, birbirinden 180° uzakta bulunan, kutupsal yörüngede dönen iki uydudan oluşur. Bu uydular, arazi yüzeyi koşullarındaki değişiklikleri takip ederler. Geniş kapsama alanı genişliği (290 km) ve uzun tekrar ziyaret süresi (bulutsuz koşullarda ekvatorda tek bir uyduyla 10 gün ve iki uydu ile beş gün, orta enlemlerde 2-3 gün ile sonuçlanır) Dünya üzerindeki değişikliklerin izlenmesine yardımcı olacaktır. SENTINEL-2 ürünleri kullanıcıların hizmetine sunulmuştur. Bazı ürünler yalnızca uzmanlar içindir (sensör geometrisinde atmosferin üstü yüzey ışıması) ve diğerleri tüm kullanıcılara yöneliktir (kartografik geometride atmosferin üst kısmındaki yansımalar ve aynı geometride atmosferik olarak düzeltilmiş yüzey yansımaları). Pilot ürünler yalnızca talep üzerine üretilir. Pilot ürünlerin iki kategorisi vardır: Kartografik geometride Harmonize SENTINEL-2+Landsat-8/9 yüzey yansımaları.

SENTINEL-3, deniz yüzeyi topografyası, deniz ve kara yüzey sıcaklığı ile okyanus ve kara yüzey rengi ölçümlerini alır. Buradaki fikir, okyanus tahmin sistemlerini ve çevre ve iklim izlemeyi desteklemektir.

SENTINEL-4, Avrupa'daki temel hava kalitesi eser gazları ve aerosolleri izleyerek Copernicus Atmosfer İzleme Servisi (CAMS) hızlı bir tekrar ziyaret süresiyle destekler. Spektral ve radyometrik olarak kalibre edilmiş ve coğrafi konumu belirlenmiş Dünya parlaklığı ve spektral ve radyometrik olarak kalibre edilmiş güneş ışınımı, tüm kullanıcılar için parametre olarak mevcuttur ancak veri işleme parametreleri, kalibrasyon ve cihaz teşhis verileri yalnızca uzman kullanıcılar içindir.

SENTINEL-5, ultraviyole ile kısa dalga kızılötesi aralığında çalışan, yedi farklı spektral bantlı yüksek çözünürlüklü bir spektrometre sistemidir: UV-1 (270-300nm), UV-2 (300-370nm), VIS (370-500nm), NIR-1 (685-710nm), NIR-2 (745-773nm), SWIR-1 (1590-1675nm) ve SWIR-3 (2305-2385nm). Sentinel-5, hava kalitesi ve bileşim-iklim etkileşimi (O3, NO2, SO2, HCHO, CHOCHO ve aerosoller) hakkında bilgi sağlar. Sentinel-5, iklim, hava kalitesi ve ozon/yer yüzeyi UV uygulamaları için günlük küresel kapsama alanı ile CO, CH4 ve stratosferik O3 için kalite parametreleri sunar.

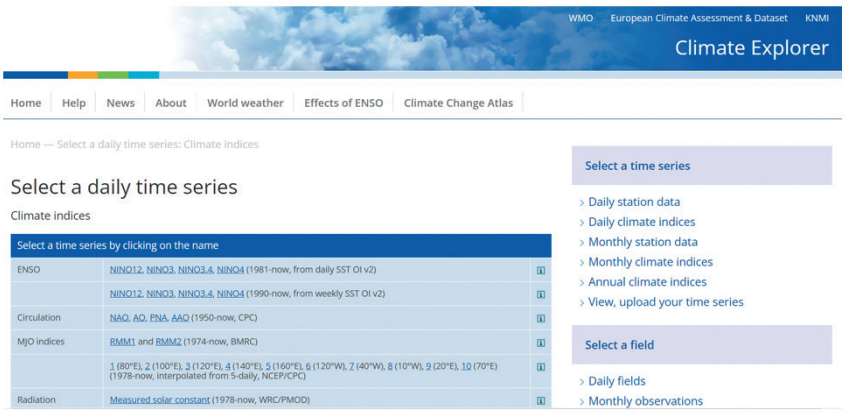
SENTINEL-5P, hava kalitesi, ozon ve UV radyasyonu ile iklim izleme ve tahmini için yüksek bir uzamsal ve zamansal çözünürlükte atmosferik ölçümler gerçekleştirir.

Copernicus SENTINEL-6 Michael Freilich, iklim değişikliği nedeniyle yükselen deniz seviyesine odaklanır ve deniz yüzeyi yüksekliği ölçümlerinin mirasını en az 2030 yılına kadar uzatacak bir sonraki radar altimetri referans görevidir.

Bir diğer önemli veri tabanı ise Avrupa düzeyinde hava istasyonlarından elde edilen gözlemleri ve bunlardan elde edilen veri setlerini içeren **ECA&D'dir**; araştırmacılar bu veri kümelerini referans verileri olarak değerlendiriyor. Bu site, hava ve iklim aşırılıklarındaki değişikliklere ilişkin bilgileri ve bu aşırılıkları izlemek ve analiz etmek için gereken günlük veri setini içerir.

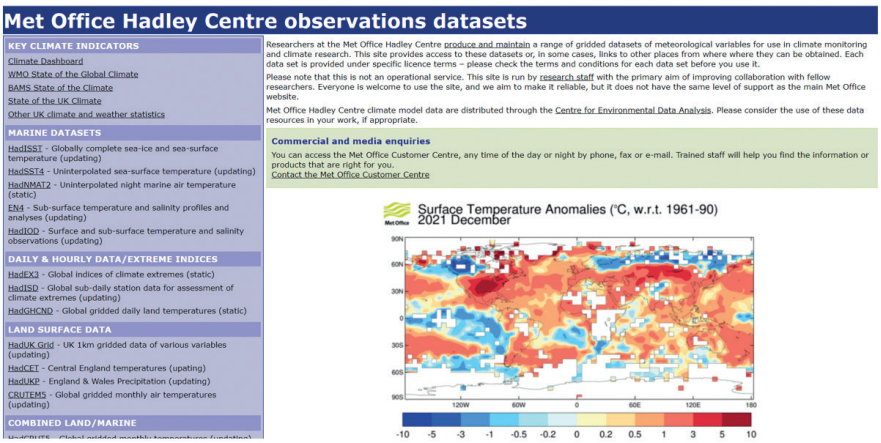
Şekil C.4. ECA&D ve WMO'nun ekran görüntüsü (kaynak: https://www.ecad.eu)

KNMI Climate Explorer, yeniden analiz ve iklim modellerinden (iklim projeksiyonları dahil) iklim verileri (zaman serisi veya alan) içeren başka bir veritabanıdır; daha kullanıcı dostu bir arayüze (grafiksel gösterimler dahil) sahiptir. Bu nedenle iyi bir eğitim aracıdır. Kullanıcılar, günlük ve aylık istasyon verileri ve iklim endeksleri hakkında bir zaman serisi indirebilir. Yıllık düzeyde yalnızca yıllık iklim endeksleri mevcuttur. Araştırmacılar, günlük alanlar, aylık gözlemler, aylık yeniden analiz alanları, aylık ve mevsimsel tarihi yeniden yapılandırmalar, aylık mevsimsel geriye tahminler, aylık CMIP3+ senaryo çalışmaları, aylık CMIP5 senaryo çalışmaları, yıllık CMIP5 ekstremleri, aylık CMIP6 senaryo çalışmaları, aylık CORDEX senaryo çalışmaları, atıf çalışmaları gibi bir alan seçerek bu bilgileri indirebilir.



Şekil C.5. KNMI Climate Explorer'ın ekran görüntüsü
(kaynak: https://climexp.knmi.nl/selectdailyindex.cgi?id=someone@somewhere)

Met Office Hadley Centre, meteorolojik değişkenlerin veri kümelerini sağlar. Araştırmacılar bu bilgileri iklim izleme ve iklim araştırmalarında kullanırlar. Sınıflar şunlardır: ana iklim göstergeleri, deniz veri kümeleri, günlük ve saatlik veriler/ekstrem endeksler, kara yüzeyi verileri, kombine kara/deniz basınc verileri, üst hava verileri, dergi makalelerine eşlik eden tek seferlik veriler ve daha eski veri kümeleri.

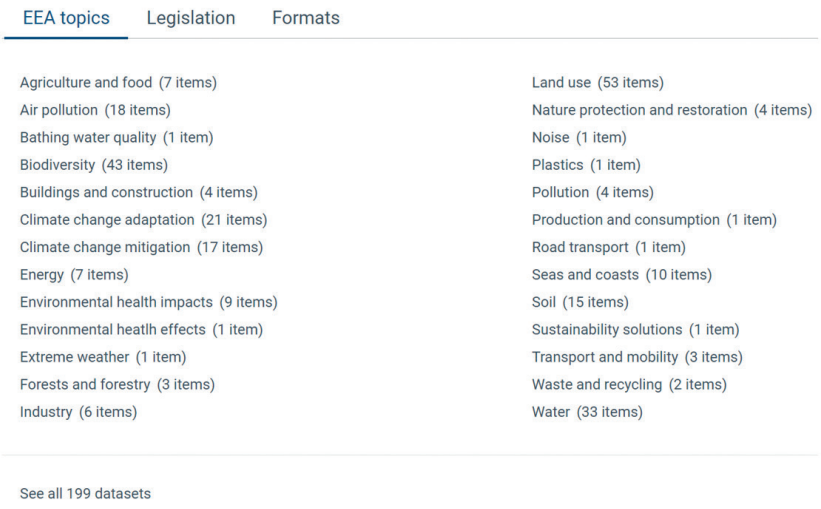


Şekil C.6. Met Office Hadley Merkezinin ekran görüntüsü (kaynak: https://www.metoffice.gov.uk/hadobs/index.html)

Indecis tarım, afet riskinin azaltılması, enerji, sağlık, su ve turizme ilişkin iklim verilerini içerir (http://indecis.eu/indices.php). Burada yalnızca iklim endeksleri var - birçoğu çeşitli uygulamalara sahiptir. Platform, iklim endeksleri için tanımlara, harita olarak grafiksel gösterime ve tek bir noktada bir dizi veriye ve ayrıca indirme özelliğine sahiptir. Bu veritabanı aynı zamanda iyi bir eğitim aracıdır. Günlük istasyon verilerini, kalite kontrollü istasyon verilerini, homojenleştirilmiş istasyon verilerini, kurtarılan istasyon verilerini ve endekslerin ızgaralı versiyonlarını içerir.

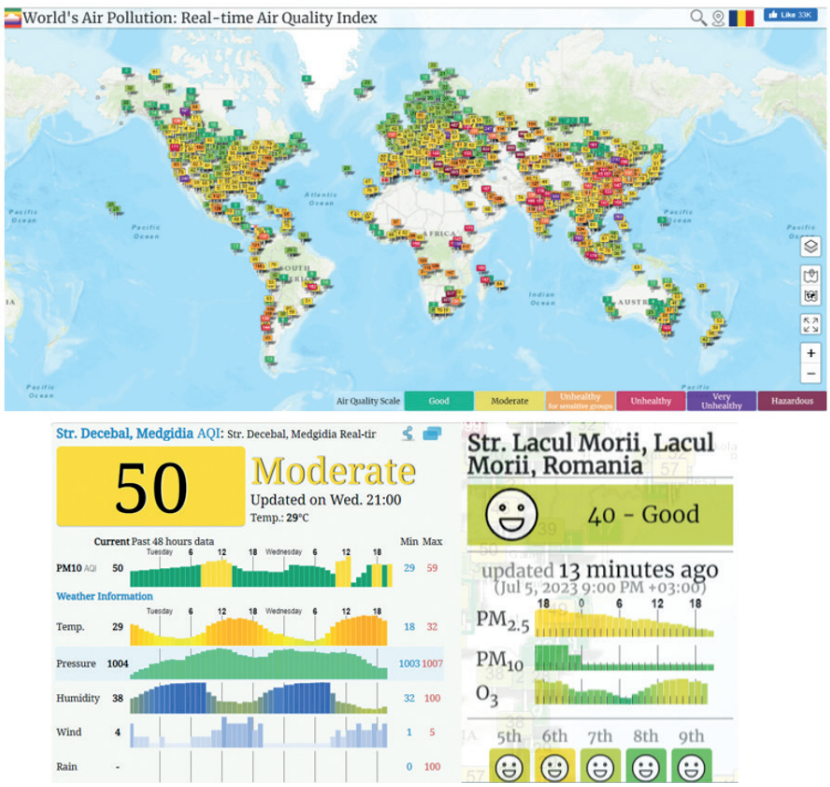
Şekil C.7. Indecis'ten indirilebilecek veri sınıfları
(kaynak: https://www.ecad.eu/dailydata/preDefinitionseries.php)

Hava kalitesi verileri için Avrupa Çevre Ajansı'nın açık kaynaklı verileri bulunmaktadır.



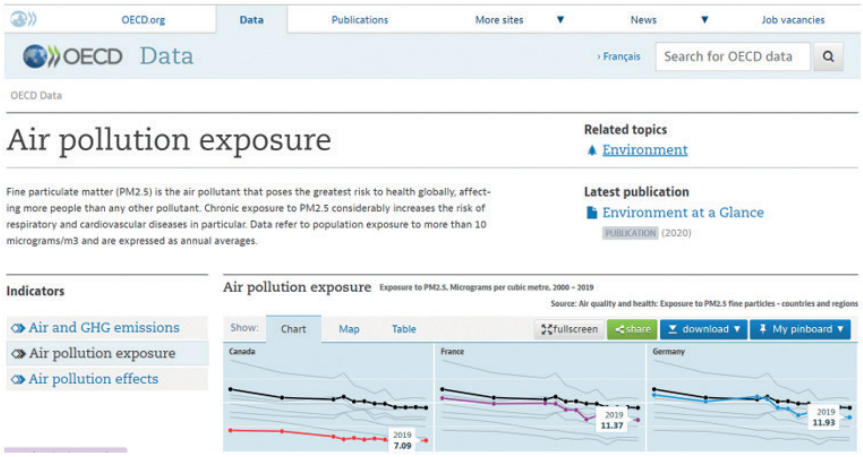
Şekil C.8. Avrupa Çevre Ajansı tarafından verilen veri setleri
(kaynak: https://www.eea.europa.eu/themes/air/explore-air-pollution-data)

Dünyanın Hava Kirliliği, Ulusal Çevre Ajanslarının sensörlerini içerir ve gerçek zamanlı hava kalitesi endeksi hakkında bilgi verir.



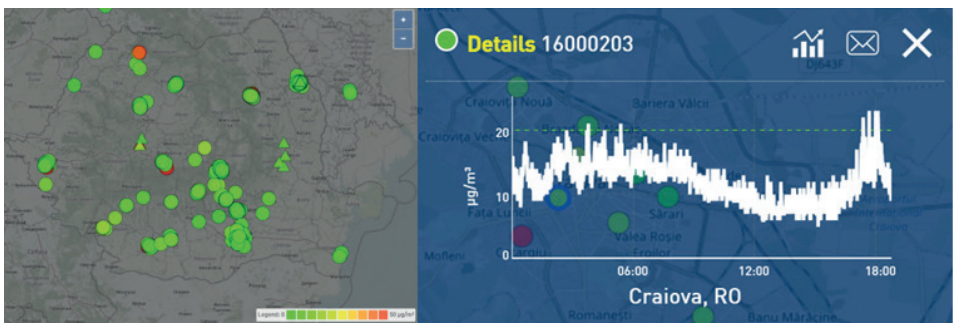
Şekil C.9. Dünya çapındaki hava kalitesi sensörlerinin haritası ve bir sensöre tıklandığında daha fazla bilgi
(kaynak: https://waqi.info/)

OECD, herkesin verilerin zaman içindeki gelişimini görselleştirmesi için basit olacak şekilde, hava ve GHC emisyonları, hava kirliliğine maruz kalma ve hava kirliliğinin etkileri gibi göstergeler hakkında grafikler, haritalar veya tablolar halinde bilgi içerir. Basit bir tıklama, grafikler, haritalar ve tablolar halinde düzenlenmiş verileri açacaktır.



Şekil C.10. OECD ekran görüntüsü (kaynak: https://data.oecd.org/air/air-pollution-exposure.htm)

Ayrıca vatandaş bilimi girişimleri ve topluluk odaklı sensör ağları da var. Bu ağlar, vatandaşların kendi topluluklarındaki hava kalitesini izleyen ve Avrupa'nın geniş bir alanını mükemmel şekilde kapsayan düşük maliyetli sensörler içerir. Bu ağların bir kısmı gönüllüler tarafından bazı projelerde eğitim amaçlı olarak kurulmuştur. uRADMonitor® Romanya'da böyle bir örnektir. Ağ, gerçek zamanlı verilere açık erişim sağlar. Yöneticiler istek üzerine geçmiş verileri sağlayabilir. Vatandaş bilimi girişimleri çevresel izlemede şeffaflığı ve hesap verebilirliği teşvik eder. Diğer örnekler şunlardır: Topluluk Hava Sensörü Ağı (CAIRSENSE), Smart Citizen® ağı, Açık Teknoloji ve Bilim Kamu Laboratuvarı veya Kamu Laboratuvarı ağı, Eye on Earth girişimi, Çevreye Fayda Sağlayacak Küresel Öğrenme ve Gözlemler (GLOBE), HabitatMap©, Imperial County Topluluğu Hava İzleme Projesi ve Vatandaş Hava Durumu Gözlemci Programı (CWOP).



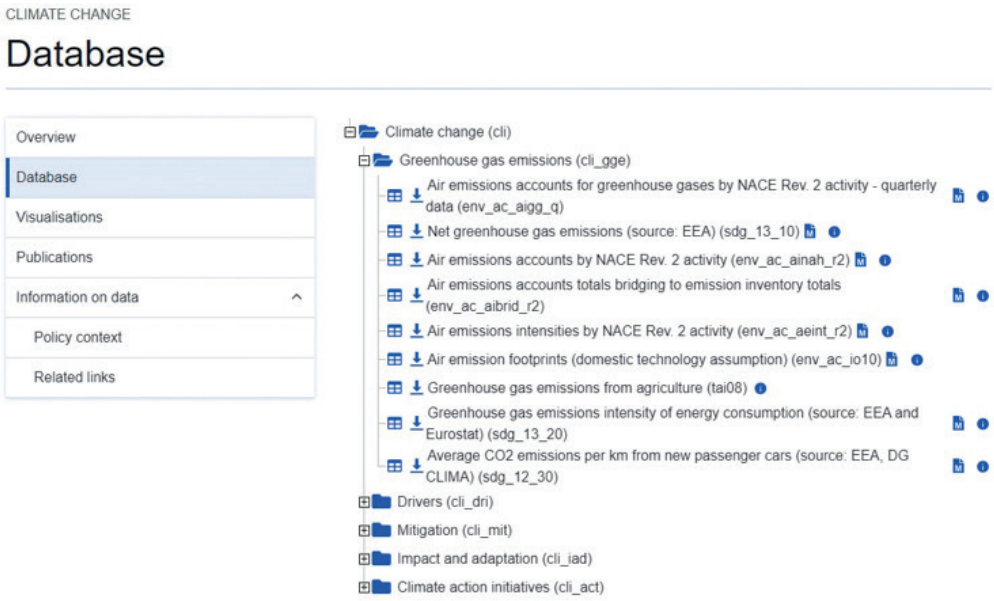
Şekil C.11. uRADMonitor® ağı ekran görüntüsü (kaynak: https://www.uradmonitor.com/)

İstatistiksel veritabanları, birden fazla değişken ve daha uzun dönemler için önemli veriler sağlayarak araştırmanın ilerlemesine yardımcı olur. Bu bilgi, sonuçların çıkarılmasında, senaryoların oluşturulmasında, tahmin edilmesinde ve başlatılmasında temeldir. Örneğin, kirlilik seviyeleri gibi

iklim değişikliğiyle ilgili verilere ihtiyacımız varsa, erişebileceğimiz ve hedeflerimiz doğrultusunda kullanabileceğimiz birden fazla açık kaynak veri tabanına sahibiz. Kararlar veri girişlerine dayanmaktadır; Seçimlerimiz ancak doğru bilgiye sahipsek iyidir. Bu nedenle, tanınmış ulusal ve uluslararası kuruluşlar gibi güvenilir bilgi kaynaklarını seçmek önemlidir.

Bu kaynaklardan biri, Avrupa ülkelerini ilgilendiren pek çok husus hakkında istatistiksel veriler sağlayan Eurostat veri tabanıdır. Bu veri tabanından emin olabilmemizin nedenlerinden biri de köklü (70 yıllık) bir geçmişe sahip olması ve Avrupa Birliği çatısı altında olmasıdır.

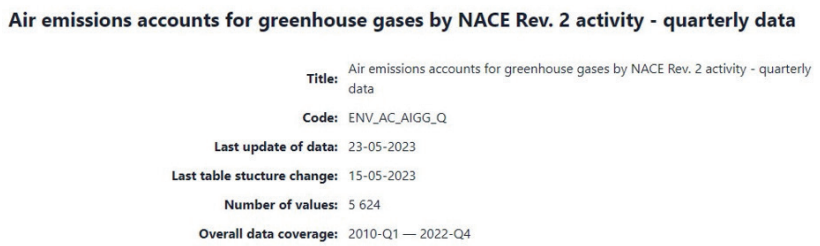
Eurostat veri tabanını kullanarak iklim değişikliği ile ilgili verilere nasıl ulaşabileceğimize bir örnek verelim. Web sitesine doğrudan ulaşıp iklim değişikliğini araştırabilir veya bunu bir arama motoru kullanarak yapabiliriz. https://ec.europa.eu/eurostat/web/climate-change/database adresine gidersek, Şekil C.12'de gösterildiği gibi amacımıza yönelik birden fazla veri bulabiliriz.



Şekil C.12. Eurostat web sitesinden iklim değişikliği ile ilgili bilgilere ilişkin ekran görüntüsü (kaynak:https://ec.europa.eu/eurostat/web/climate-change/database)

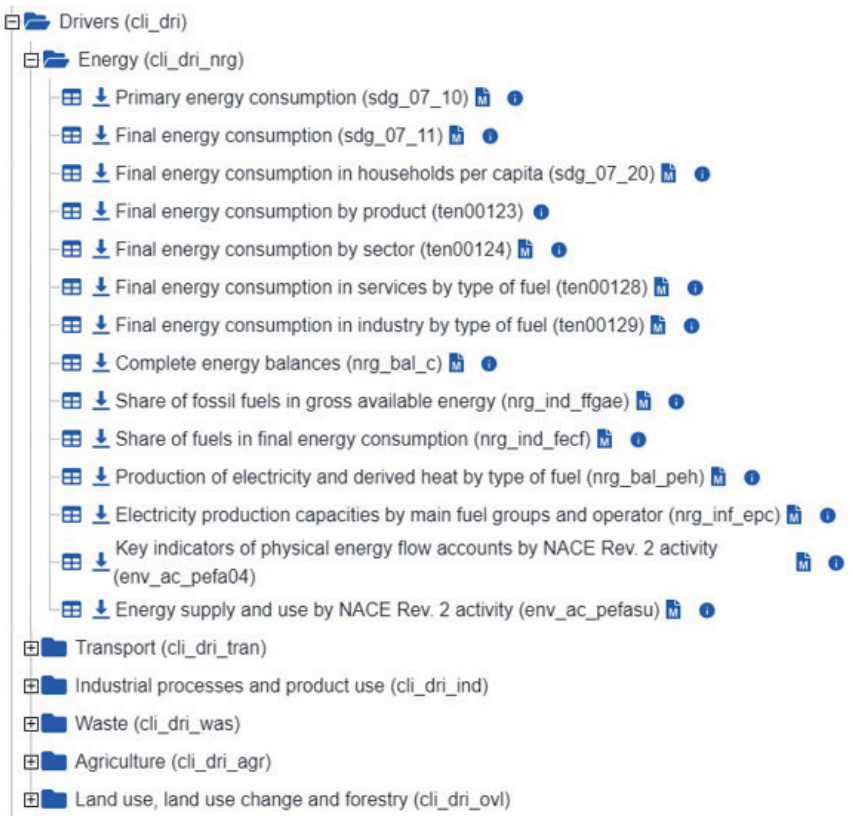
İklim değişikliği veri tabanında birçok klasör bulunmaktadır: sera gazı emisyonları; iklim değişikliğinin etkenleri; iklim değişikliğinin azaltılması; etki ve adaptasyon; ve iklim eylemi girişimleri. Her birinde kullanıcının indirebileceği veriler bulunur. Bilgi ücretsizdir ve çeşitli formatlarda herkese açıktır.

Sera gazı emisyonları klasörünün altında birkaç veri parçasının bulunduğunu görebiliriz. İlki olan Sera gazları için hava emisyonları hesabına (üç aylık veriler) gidersek, sağ butona tıklarsak daha fazla bilgiye ulaşabiliriz. Görüntülenen pencere Şekil C.13'teki penceredir. Böylece 2010'dan 2022'ye kadar 13 yıllık verilerin mevcut olduğunu görebiliyoruz. Veri tabanı Mayıs 2023'te güncellendi.



Şekil C.13. "Information" düğmesine tıkladıktan sonraki ekran görüntüsü (kaynak: https://ec.europa.eu/eurostat/web/climate-change/database)

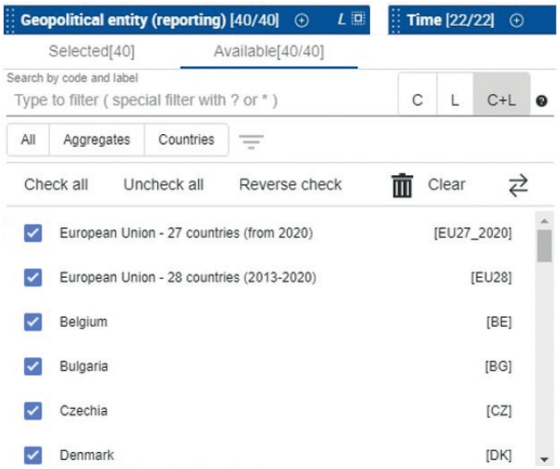
İklim değişikliğinin nedenlerini daha fazla incelemek istiyorsak, ikinci klasörü seçeriz ve Şekil C.14'te görüldüğü gibi enerji, ulaşım, endüstriyel süreçler, atık, tarım ve arazi kullanımı, arazi kullanımı değişikliği ve ormancılık gibi tüm önemli nedenler için verilerin olduğunu fark ederiz. Enerji için, nihai enerji tüketimi, nihai enerji tüketimi kişi başına, sektöre göre nihai enerji tüketimi gibi verileri indirebiliriz. Kullanıcının hedeflerine ulaşmak için gereken verileri seçmek önemlidir. Veriler ham ve işlenmemiş olduğundan, kullanıcı verileri işlemek için çeşitli araçlar kullanabilir, bir eğilim fark edebilir, bazı senaryoları tahmin edebilir ve ortaya koyduğu sonuçları sağlayabilir. İşletmeler, bireyler, hükümetler ve sorumlulukları olan diğer kuruluşlar, bu sonuçları bazı yönlerden önlemek veya iyileştirmek için kullanacaktır.



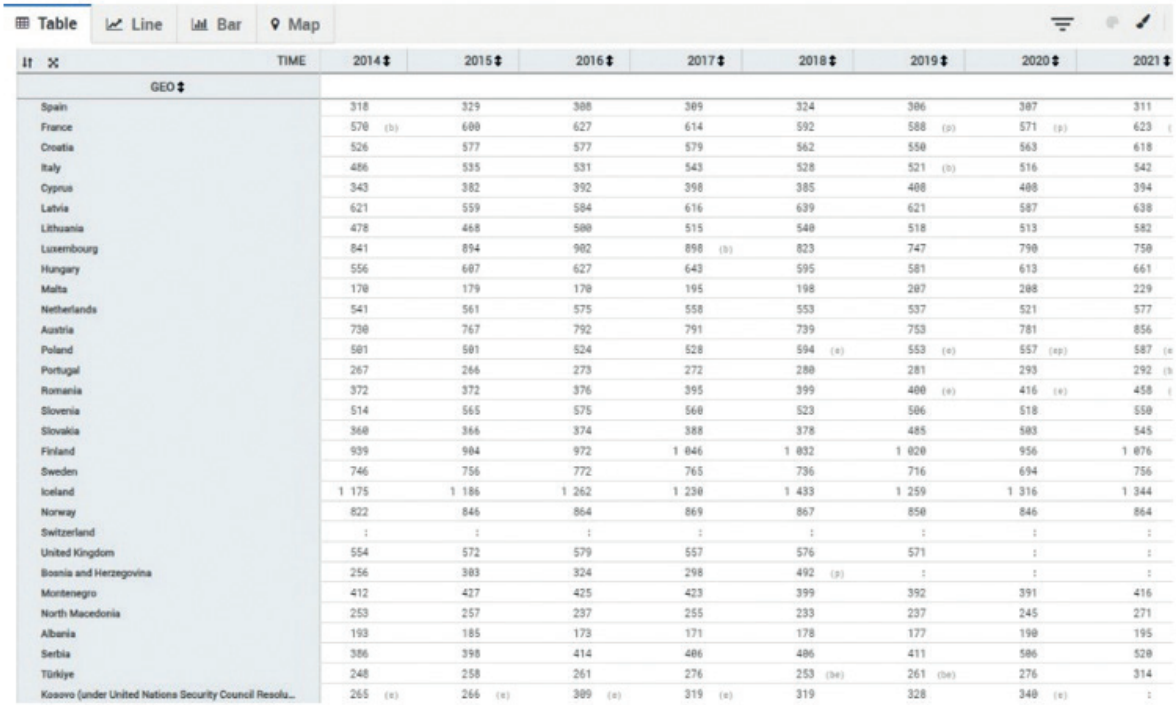
Şekil C.14. Eurostat'tan iklim değişikliğinin etkenlerine ilişkin bilgilerin yazdırıldığı ekran (kaynak: https://ec.europa.eu/eurostat/web/climate-change/database)

Şimdi hanelerde kişi başına düşen nihai enerji tüketimini kontrol etmek istiyorsak bilgilerin nasıl göründüğüne bakalım. Şekil C.12'de bu göstergenin yanında parantez içindeki bir kod gösterilmektedir: SDG 7. Bu bilgi aslında Birleşmiş Milletler 2030 Gündemi'ndeki yedinci sürdürülebilir kalkınma hedefine bir referanstır. Bu Uygun Fiyatlı ve Temiz Enerji anlamına gelir.

Tabloya benzeyen ilk simgeye tıklarsak göstergeye ilişkin açıklamaları okuyabiliriz, ancak aynı zamanda veri formatını (tablo, çizgi, çubuk, harita) ve ihtiyacımız olan değişkenleri (ülkeler ve yıllar) da seçebiliriz - Bkz. Şekil C.15 ve C.16.

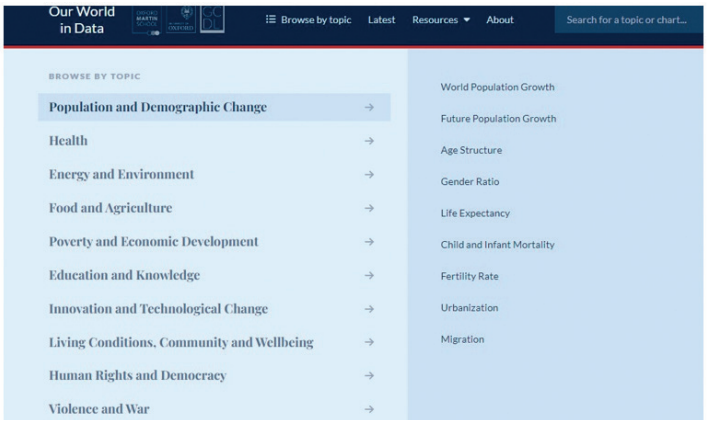


Şekil C.15. Eurostat verileri için kullanılabilecek filtrelerin ekran görüntüsü (kaynak: https://ec.europa.eu/eurostat/web/climate-change/database)



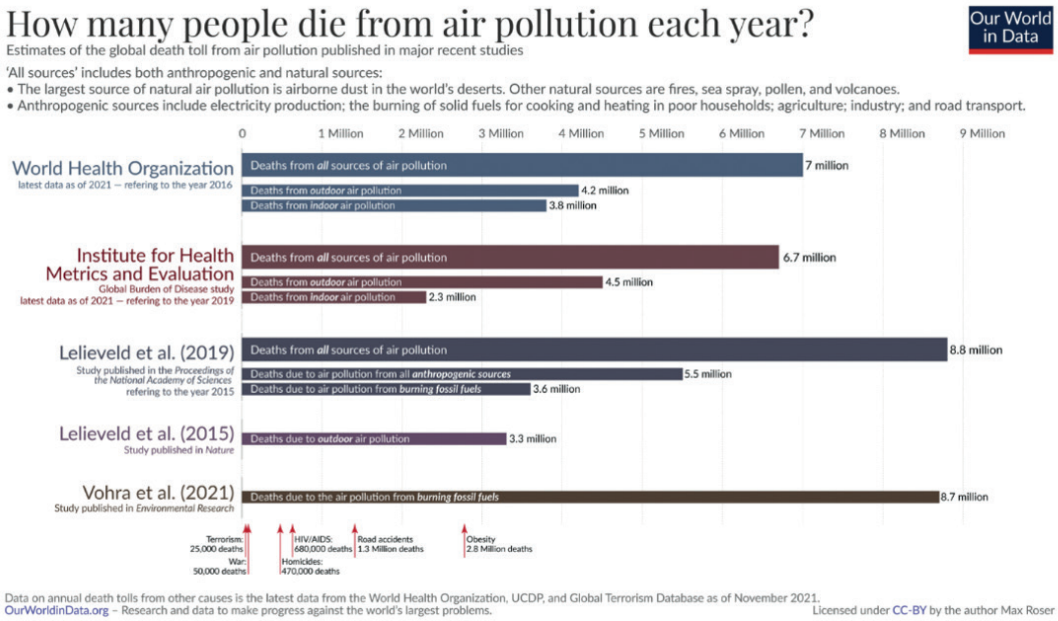
Şekil C.16. Tablo formatını seçtiğimizde görüntülenen bilgilerin ekran görüntüsü (kaynak: https://ec.europa.eu/eurostat/web/climate-change/database)

Araştırmacılar Eurostat'ın yanı sıra diğer açık kaynaklı veritabanlarını da kullanabilirler. Web sitelerinde (https://ourworldindata.org/) ana hedefi belirtilen Our World in Data'dan bahsedebiliriz. Nüfus dinamiği, enerji ve çevre, sağlık, gıda, yoksulluk, eğitim, yaşam koşulları, insan hakları, teknolojik değişimler, şiddet ve savaş gibi dünyanın en büyük sorunlarına karşı ilerleme sağlayacak araştırma ve verileri yayınlıyorlar. Kâr amacı gütmeyen bir kuruluşun şemsiyesi altındadır ancak literatür taramasında ve medyada oldukça fazla alıntı yapılmaktadır.

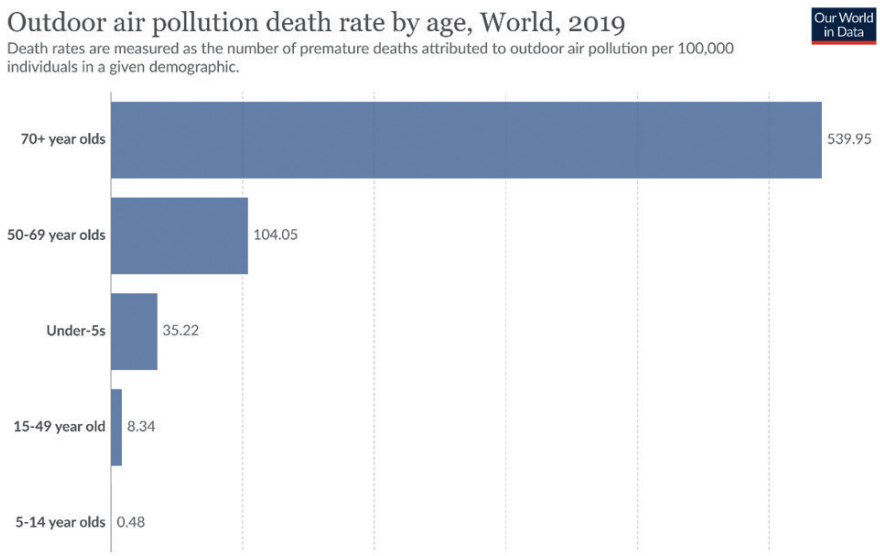


Şekil C.17. Our World in Data'dan burada ele alınan konulara ilişkin ekran görüntüsü (kaynak: https://ourworldindata.org/)

Eğer hava kirliliği ile ilgileniyorsak Energy and Environmet (Enerji ve Çevre)'ı seçiyoruz ve outdoor air pullution (dış mekan hava kirliliği) veya indoor air pollution (iç mekan hava kirliliği) seçebiliyoruz. Bu web sitesi araştırmanız veya etkinliğiniz için makaleler ve ham istatistiksel veriler sunar. Böylece dünya çapında kaç ölümün hava kirliliğine atfedilebileceğini keşfedebilirsiniz (Şekil C.18). Ayrıca yaşa göre dış hava kirliliği oranını bulabilir ve verileri tablo veya grafik olarak indirebilirsiniz (Şekil C.19).



Şekil C.18. Hava kirliliğinden kaynaklanan küresel ölümler (kaynak: https://ourworldindata.org/data-review-air-pollution-deaths)



Şekil C.19. Yaşa göre hava kirliliği ölüm oranı
(kaynak: https://ourworldindata.org/outdoor-air-pollution#outdoor-air-pollution-deaths-by-age)

D-HAVA KİRLİLİĞİNİN İNSAN SAĞLIĞINA ETKİSİ

El kitabının bu bölümü, Bulgaristan'daki Plovdiv Üniversitesi "Paisii Hilendarski" Biyoloji Fakültesi, Ekoloji ve Çevre Koruma Bölümü'nden Slaveya Petrova tarafından yazılmıştır.

Hava kirliliği, atmosferin doğal özelliklerini değiştiren herhangi bir kimyasal, fiziksel veya biyolojik ajanın iç veya dış ortamı kirletmesidir. Evlerdeki yanma cihazları, motorlu taşıtlar ve endüstriyel tesisler hava kirliliğinin yaygın kaynaklarıdır. Halk sağlığı açısından önemli kirleticiler arasında partikül madde (PM), karbon monoksit (CO), ozon (O3), nitrojen dioksit (NO2) ve kükürt dioksit (SO2) yer alır.

Avrupa Çevre Ajansı'na (AÇA) göre, hava kirleticilerin her biri farklı bir kaynak/kaynaklarla ilişkilendirilebilir:

- Konut, ticari ve kurumsal enerji tüketimi 2020 yılında partikül maddenin ana kaynağı oldu. İmalat ve madencilik endüstrisi ile tarım da PM10'un önemli kaynaklarıydı. Partikül madde emisyonlarında (PM10 ve PM2.5) 2005 ile 2020 yılları arasında bir azalma eğilimi gözlemlendi; emisyonlar sırasıyla %30 ve %32 oranında düştü.
- 2020 yılında amonyak (toplam emisyonların %94'ü) ve metanın (%56) ana kaynağı tarımdı. Amonyak emisyonları 2005'ten 2020'ye yalnızca %8 düştü. Bu, tüm kirleticiler arasında en düşük azalma yüzdesiydi.
- Karayolu taşımacılığı, 2020'de nitrojen oksitlerin ana kaynağıydı ve emisyonların %37'sini açığa çıkardı. Nitrojen oksit emisyonlarında 2005 ile 2020 yılları arasında %48'e varan önemli bir düşüş tespit edildi.
- Enerji tedarik sektörü, 2020'deki emisyonların %41'inden sorumlu olan kükürt dioksitin ana kaynağıydı. Kükürt dioksit emisyonları 2005 ile 2020 arasında %79 azaldı.
- İmalat ve çıkarma endüstrileri ile enerji tedarik sektörü, 2020 yılında ağır metal emisyonlarının ana kaynaklarıydı. 2005 ile 2020 arasında emisyonlarda en büyük azalmalar nikel (%64) ve arsenik (%62) için görüldü.

D.1 KİRLETİCİ MADDE TÜRLERİ VE SAĞLIK RİSKLERİ

Partikül madde (PM)

Partikül madde (PM), hava kirliliğinin yaygın bir temsili göstergesidir. PM fraksiyonlarının ana bileşenleri sülfatlar, nitratlar, amonyak, sodyum klorür, siyah karbon, mineral tozu ve sudur.

Çapı 10 ve 2,5 mikrondan (PM_{10} ve $PM_{2,5}$) küçük partikül maddelerle ilişkili sağlık riskleri özellikle iyi belgelenmiştir. PM akciğerin derinliklerine nüfuz edebilir ve

kan dolaşımına neden olarak kardiyovasküler (iskemik kalp hastalığı), serebrovasküler (felç) ve solunumsal etkilere neden olur. PM'ye hem uzun hem de kısa süreli maruz kalma, kardiyovasküler ve solunum yolu hastalıklarından kaynaklanan morbidite ve mortalite ile ilişkilidir. Uzun süreli maruz kalma, olumsuz perinatal sonuçlar ve akciğer kanseri ile ilişkilendirilmiştir.

Karbon monoksit (CO)

Karbon monoksit, odun, petrol, odun kömürü, doğalgaz ve kerosen gibi karbonlu yakıtların tam olarak yanmaması sonucu oluşan renksiz, kokusuz ve tatsız zehirli bir gazdır. Karbon monoksit akciğer dokularına ve kan dolaşımına yayılarak vücut hücrelerinin oksijene bağlanmasını zorlaştırır. Bu oksijen eksikliği dokulara ve hücrelere zarar verir. Karbon monoksite maruz kalmak nefes almada zorluklara, bitkinliğe, baş dönmesine ve diğer grip benzeri semptomlara neden olabilir. Yüksek düzeyde karbon monoksite maruz kalmak ölümcül olabilir.

Ozon (O_3)

Yer seviyesindeki ozon, sıcak kimyasal dumanın ana bileşenlerinden biridir. Güneş ışığı varlığında gazlarla reaksiyona girerek ortaya çıkar. Ozonun taşınabilir hava temizleyicileri gibi ev aletleri tarafından üretilebileceğini belirtmekte fayda var. Aşırı ozona maruz kalmak nefes alma gibi sorunlara neden olabilir, astımı tetikleyebilir, akciğer fonksiyonunu azaltabilir ve akciğer hastalığına yol açabilir.

Azot dioksit (NO_2)

NO2, ulaşım ve sanayi sektörlerinde yakıtların yanmasından yaygın olarak salınan bir gazdır. Evdeki nitrojen oksit (NOx) kaynakları arasında fırınlar, şömineler, gaz sobaları ve fırınlar gibi yakıt yakan ekipmanlar bulunur. Azot dioksite maruz kalmak solunum yollarını tahriş edebilir ve solunum yolu hastalıklarını ağırlaştırabilir.

Kükürt dioksit (SO_2)

SO2, fosil yakıtların (kömür ve petrol) yakılmasından ve kükürt içeren mineral cevherlerinin eritilmesinden üretilen, keskin kokulu, renksiz bir gazdır. SO2'ye maruz kalma, astım hastanesine başvurular ve acil servis ziyaretleriyle ilişkilidir.

Polisiklik aromatik hidrokarbonlar (PAH)

Polisiklik aromatik hidrokarbonlar (PAH) atmosferde parçacık halinde bulunur. Bunlar esas olarak organik maddelerin (örneğin etin pişirilmesi) ve fosil yakıtların kok fırınlarında, dizel motorlarda ve odun sobalarında eksik yanmasından oluşan bir grup kimyasaldır. Ayrıca tütün dumanında da bulunabilirler. Kısa süreli maruz kalma gözleri ve solunum yollarını tahriş edebilir. PAH'a uzun süreli maruz kalma akciğer kanseriyle ilişkilendirilmiştir.

D.2 DSÖ KÜRESEL HAVA KALİTESİ KURALLARI

1987'den bu yana, DSÖ, hükümetlerin ve sivil toplumun, insanların hava kirliliğine maruz kalmasını ve bunun olumsuz etkilerini azaltmasına yardımcı olmak amacıyla periyodik olarak sağlık temelli hava kalitesi kılavuzları yayınlamaktadır. Ana amaç, hava kalitesi yönetimi için, çeşitli önemli hava kirleticiler için uzun veya kısa vadeli konsantrasyonlar olarak ifade edilen, niceliksel, sağlık temelli öneriler sunmaktır. Hava kalitesi kılavuzu (AQG) seviyelerinin aşılması, halk sağlığına yönelik önemli risklerle ilişkilidir. Bu yönergeler yasal olarak bağlayıcı standartlar değildir ve zorunlu bir nitelik taşımamaktadır. Bununla birlikte, DSÖ Üye Devletlerine, dünya çapında hava kirliliğine maruz kalmanın neden olduğu devasa sağlık yükünü azaltmak amacıyla hava kirletici düzeylerini azaltmak için ulusal programlara uygulayabilecekleri, kanıta dayalı bir araç sağlamaktadırlar.

Tablo D.1. Her kirletici için önerilen hava kalitesi kuralları [5]

Kirleticiler	Kılavuz Değeri	Ortalama süre	Kılavuz Kaynağı
$PM_{2,5}$	5 µg/m³	Yıllık	DSÖ 2021
	15 µg/m³	24 saat	
PM_{10}	15 µg/m³	Yıllık	DSÖ 2021
	45 µg/m³	24 saat	
Karbon monoksit (CO)	4 µg/m³	24 saat	DSÖ 2021
Azot dioksit (NO_2)	10 µg/m³	Yıllık	DSÖ 2021
	25 µg/m³	24 saat	
Sülfür Dioksit (SO_2)	40 µg/m³	24 saat	DSÖ 2021
Formaldehit	0,1 µg/m³	30 dakika	DSÖ 2010
Polisiklik Aromatik Hidrokarbonlar	$8,7 \times 10^{-5}$ per ng/m³		DSÖ 2010
Radon	100 Bq/m³		DSÖ 2010
Kurşun	0,5 µg/m³	Anual	DSÖ, Avrupa Bölge Ofisi, 2000

D.3 ÇEVRESEL HAVA KİRLİLİĞİNİN SAĞLIĞA ETKİSİNE İLİŞKİN ÇALIŞMALAR

Geçtiğimiz yüzyılda fosil yakıtların artan yanması ve trafiğin sürekli yoğunlaşması, atmosfer bileşimindeki ilerleyici değişimin sorumlusu olup, bu da yaşam kalitesini olumsuz yönde etkilemektedir.

Bu durumun başlıca nedenlerinden biri, çocukların özellikle hassas olduğu araç egzoz gazlarının sağlık üzerindeki etkisidir. Egzoz gazları 200'ün üzerinde kirletici tür içerir; bunlardan bazıları şunlardır: CO_2, NO_x, CO, SO_x, düşük moleküler ağırlıklı hidrokarbonlar, aldehitler (formaldehit, asetaldehit, akrolein), benzen, 1,3 bütadien, polisiklik hidrokarbonlar, oksidatif bileşen parçacıkları (elementel karbon, adsorbe edilmiş aromatik hidrokarbonlar, az miktarda sülfat, nitrat, metaller, ve diğer unsurlar), vb.

Ekonomik durgunluk sırasında ekonomik faaliyetlerin azalması atmosferik emisyonların azalmasına yol açsa da, genel olarak Avrupa'daki otomobil taşımacılığının, Avrupa Birliği'ndeki zararlı hava kirletici düzeylerden ve sera gazı emisyonlarının dörtte birinden sorumlu olduğu düşünülmektedir. Araçlara yönelik "Euro" standartları bir miktar başarı elde etti ancak NO_2'yi önemli ölçüde azaltmadı.

Karbon monoksit (CO), kükürt dioksit (SO_2), nitrojen oksitler (NOx), uçucu organik bileşikler (VOC'ler), ozon (O3), ağır metaller ve partikül madde (PM2,5 ve PM10) gibi hava kirleticileri kimyasal bileşimler, reaksiyon özellikleri, parçalanma süreleri ve uzun veya kısa mesafelerde dağılma yetenekleri açısından farklılık gösterir.Dış ortam hava kirliliği, düşük, orta ve yüksek gelirli ülkelerdeki herkesi etkileyen önemli bir çevre sağlığı sorunudur çünkü solunum yolu hastalıklarına ve diğer hastalıklara neden olabilir ve önemli hastalık ve ölüm kaynaklarıdır [9]. Hava kirleticilerin insan sağlığı üzerindeki bu etkileri ve etki mekanizmaları aşağıda kısaca tartışılacaktır.

Hava kirliliğinin insan sağlığı üzerinde birçok farklı sistem ve organı etkileyen akut ve kronik etkileri vardır. Hafif üst solunum yolu tahrişinden kronik solunum ve kalp hastalığına, akciğer kanserine,

çocuklarda akut solunum yolu enfeksiyonlarına ve yetişkinlerde kronik bronşite, önceden var olan kalp ve akciğer hastalığının ağırlaşmasına veya astım ataklarına kadar uzanır. Ayrıca, kısa ve uzun vadeli maruz kalmalar aynı zamanda erken ölüm ve azalan yaşam beklentisiyle de ilişkilendirilmiştir.

DSÖ 2019'da dış ortam hava kirliliği ile ilişkilendirilen erken ölümlerin yaklaşık %37'sinin iskemik kalp hastalığı ve inme, %18'inin kronik obstrüktif akciğer hastalığı ve akut alt solunum yolu enfeksiyonları nedeniyle gerçekleştiğini tahmin etmektedir. Ayrıca, ölümlerin %23'ü kronik obstrüktif akciğer hastalığı ve %11'i solunum yolu kanseri nedeniyle meydana gelmiştir. Hem şehirlerde hem de kırsal bölgelerdeki dış ortam hava kirliliğinin, 2019'da dünya genelinde yılda 4.2 milyon erken ölüme neden olduğu tahmin edilmektedir; bu ölüm, kardiyovasküler ve solunum hastalıkları ile kansere yol açan ince partikül maddeye maruziyetten kaynaklanmaktadır.

- AB genelinde, en son DSÖ önerilerinden daha yüksek hava kirliliği seviyelerine sahip olmak yaygındır. Yine de iyileşme işaretleri var, ancak bazı gerçekler aşağıda belirtilmiştir:
- 2021'de kentsel nüfusun %97'si, Dünya Sağlık Örgütü tarafından belirlenen sağlık temelli kılavuz düzeyinin üzerinde ince partikül madde konsantrasyonlarına maruz kaldı.
- Her yıl, AÇA üyesi ve işbirlikçi ülkelerde 18 yaşın altındaki kişilerde 1.200'den fazla ölümün hava kirliliğinden kaynaklandığı tahmin edilmektedir [10].
- 2021 verileri, Orta Doğu Avrupa ve İtalya'nın, esas olarak katı yakıtların evsel ısıtma amacıyla yakılması ve bunların sanayide kullanılması nedeniyle en yüksek partikül madde konsantrasyonlarını bildirdiğini göstermektedir.
- Tüm AB ülkeleri ozon ve nitrojen dioksit seviyelerinin Dünya Sağlık Örgütü tarafından belirlenen sağlık temelli kılavuz seviyelerinin üzerinde olduğunu bildirdi.
- Her yıl yaklaşık 275.000 erken ölüm, ince parçacıklı maddelerden ve 64.000'i ise nitrojen dioksitten (NO_2) kaynaklanmaktadır.
- Genel olarak, AB'nin kentsel nüfusunun %97'si, DSÖ'nün 2021'de belirlediği en son yönergelerin üzerinde seviyelerde ince partikül maddeye maruz kaldı.

Hava kirliliğine maruz kalmanın olumsuz etkileri, hem gelişmekte olan hem de gelişmiş ülkelerde küresel bir halk sağlığı sorunudur; çünkü çocuklar ve gençler hava kirliliğinin etkilerine karşı özellikle savunmasızdır.

Epidemiyolojik çalışmalar, hava kirliliğinin sağlık üzerindeki etkilerini değerlendirmenin en göstergesidir. En savunmasız ve çalışma grubu için uygun olanlardan biri, okul öncesi ve erken okul çağındaki çocuklardır çünkü onlar açık havada daha fazla zaman geçirirler, metabolik süreçlerin yoğunluğu daha yüksektir ve yetişkinlere göre nispeten daha fazla miktarda hava çekerler. Aynı zamanda henüz kötü alışkanlıklar (sigara, alkol tüketimi vb.) edinmemişlerdir ve endüstriyel tehlikelere maruz kalmamaktadırlar. Egzoz emisyonlarının sağlık üzerindeki olumsuz etkilerinin geniş bir kompleksinde, en açık şekilde solunum fonksiyonu, kardiyovasküler ve bağışıklık sistemi, hematopoietik ve diğer alanlardaki ihlaller göze çarpmaktadır.

Kuzey Çin'in 6 şehrinde okul öncesi ve erken okul çağındaki çocukları kapsayan geniş bir çalışma, solunum semptomları (öksürük, nefes almada zorluk, hırıltı ve balgam) ile toplam asılı toz, kükürt dioksit ve nitrojen seviyeleri arasında güçlü bir pozitif korelasyon gösterdi.

Nitrojen dioksit ile ozon arasındaki ilişki ve başta astım olmak üzere obstrüktif sendromla birlikte solunum yolu hastalıklarını tetikleyen veya şiddetlendiren ilişki özellikle vurgulanmaktadır. Bunun başlıca örneği, araç trafiğinin yoğunluğunun geçici olarak azaltılmasının bile üst solunum yollarındaki solunum semptomlarını azalttığı durumlardır.

Hava kirliliği çocukluk çağındaki fiziksel ve zihinsel gelişimi etkiler ve astım ve daha yaygın olarak saman nezlesi olarak adlandırılan Mevsimsel Alerjik Rinit (SAR) gibi solunum rahatsızlıklarını şiddetlendirir. Saman nezlesi çocuklarda en sık görülen kronik durumdur ve en çok okul öğrencileri arasında görülür. Ozon (O3) gibi hava kirleticilerin polenin alerjenitesini arttırabileceğine ve bunun da bilişsel gelişimi etkileyebileceğine dair artan kanıtlar vardır.

D.4 HAVA KİRLİLİĞİNİN SAĞLIĞA VE ÇEVREYE ETKİSİ

Hava kalitesi Avrupalılar için büyük bir endişe kaynağıdır ve AB'nin özellikle 30 yılı aşkın süredir aktif olduğu bir alandır. AB'nin hava kalitesiyle ilgili temel hedefi "insan sağlığı ve çevre üzerinde kabul edilemez etkilere ve risklere yol açmayacak hava kalitesi seviyelerine ulaşmaktır." Yıllık Flash Eurobarometer anketindeki sorular, Avrupa halkının hava kalitesi ve hava kirliliği hakkındaki görüşlerine daha fazla ışık tutarak bu çalışmayı desteklemek üzere tasarlanmıştır.

Eurobarometer araştırması aşağıdakileri incelemek için tasarlanmıştır:

- hava kalitesi sorunlarına ilişkin bilgi düzeyi;
- hava kalitesi sorunlarının algılanan ciddiyeti ve son on yılda hava kalitesinde algılanan değişiklikler;
- çeşitli sektörlerin ve faaliyetlerin hava kalitesi üzerinde algılanan etkisi;
- hava kalitesine yönelik ana tehditler;
- çevre dostu enerji ve ulaşım seçenekleri;
- hava kalitesi sorunlarını azaltmaya yönelik bireysel ve diğer eylemler;
- ve diğerleri.

2022'deki anket sonuçları, hava kalitesinin hâlâ Avrupalı vatandaşlar için ciddi bir endişe kaynağı olduğunu ortaya koyuyor. Anketten elde edilen tüm ham veriler ücretsiz olarak mevcuttur ve çevrimiçi olarak erişilebilir.

- Çoğu Avrupalı (%60) kendini yeterli bilgiye sahip hissetmezken, ankete katılanların neredeyse yarısı hava kalitesinin son on yılda kötüleştiğine inanıyor (%47).
- Çoğu Avrupalı, solunum yolu hastalıkları (%89), astım (%88) ve kalp-damar hastalıkları gibi sağlık sorunlarının ülkelerinde hava kirliliğinden kaynaklanan ciddi sorunlar olduğunu düşünüyor. Eurobarometer, vatandaşların ülkelerindeki hava kalitesi sorunları hakkında bilgi sahibi olmadıklarını ortaya koyuyor.
- Çoğu Avrupalı, mevcut AB hava kalitesi standartları hakkında yeterince bilgi sahibi değil; çünkü yanıt verenlerin yalnızca küçük bir kısmı (%27) bu standartları duymuş.
- Ancak AB hava kalitesi standartlarının farkında olan katılımcıların büyük çoğunluğu (%67) bu standartların güçlendirilmesi gerektiğini söylüyor.

Hava kirliliği algısının ve iç ve dış ortam hava kirliliğine maruz kalma riskinin değerlendirilmesine yönelik bir tarama anketi

Anketin geliştirilmesi için, hava kirliliğinin korunmasına ilişkin benzer çalışmalardan elde edilen mekanizmalara ek olarak, birçok standartlaştırılmış anket tavsiyesine dayanan bir madde havuzu kullanıldı. Belirsizliği en aza indirmek ve anlaşılırlığı artırmak için maddeler dikkatlice yazılmıştır. Toplamda madde havuzu 25 maddeden oluşmuştur. Anket, nüfusun hava kirliliğine yönelik tutum ve algılarını ve açık ve kapalı hava kirliliğine maruz kalma riskini değerlendirmek için umut verici bir araçtır. Bu anket bilim adamları, araştırmacılar, yetkililer ve sağlığın teşviki ve geliştirilmesi planlayıcıları tarafından hava kirliliğini korumayı teşvik edici programlar geliştirmek ve uygulamak için kullanılabilir.

Anket A – ana maddeleri

Lütfen tüm soruları okuyun ve kutuyu işaretleyerek veya uygun olduğu yerde kısa bir açıklama yaparak cevaplayın.

Anket anonimdir ve bireysel yanıtlarınızın gizliliğinin korunacağını garanti ederiz.

1. Cinsiyet

☐ erkek ☐ kadın

2. Yaş

☐3 yaş altı ☐ 3-7 yaş ☐ 8-14 yaş
☐ 15-20 yaş ☐ 21-30 yaş ☐ 31-40 years old
☐ 41-50 yaş ☐ 51-60 yaş ☐ 60 yaş üzeri

3. Hangi bölgede yaşıyorsunuz?

Ülke..... Yerleşim yeri .....

4. ……’da yaşadığınızı söyleyebilir misiniz?… “”

☐ kırsal bölge ☐ köy ☐ küçük kasaba
☐ orta büyüklükteki kasaba ☐ büyük şehir

5. İş

☐ öğrenci(ilkokul) ☐ öğrenci(ortaokul/lise/üniversite) ☐ serbest meslek ☐ işçi/çalışan
☐ beden işçisi ☐ mesleki faaliyet olmadan ☐ söylemeyi reddediyorum
☐ diğer
Lütfen belirtiniz…….

6. Siz de dahil olmak üzere, hanenizde 15 yaş ve üzeri kaç kişi yaşıyor?

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ diğer..... ..... *Lütfen belirtiniz*

7. Ailenizin ortalama aylık geliri nedir (üye başına)?

☐ 300 avro’ya kadar ☐ 300-600 avro ☐ 600-1000 avro
☐ 1000-1500 avro ☐ 1500 avro’dan fazla ☐ diğer..... *Lütfen belirtiniz*

8. Ne tür bir ev ısıtma sisteminiz var?

☐ elektrikli ısıtıcı ☐ gaz ısıtıcı ☐ klima ☐ fırın
☐ ahşap/pelet ısıtma ☐ güneş enerjisiyle ısıtma ☐ diğer ..... *Lütfen belirtiniz.*

9. Ülkenizdeki hava kalitesi sorunları hakkında ne kadar bilgi sahibi olduğunuzu düşünüyorsunuz?

☐ çok bilgili ☐ bilgili ☐ iyi bilgilendirilmemiş
☐ hiç bilgilendirilmemiş ☐ diğer ..... *Lütfen belirtiniz.*

10. Son 10 yılda ülkenizdeki hava kalitesinin..... olduğunu düşünüyor musunuz?

☐ gelişmiş ☐ aynı ☐ kötüleşmiş
☐ diğer ..... *Lütfen belirtiniz.*

11. Aşağıdakilerden her birinin ülkenizdeki hava kalitesi üzerinde ne kadar etkisi olduğunu düşünüyorsunuz? Etkisi büyük mü, orta düzeyde mi, küçük bir etkisi mi var, yoksa hiç etkisi yok mu?

	Büyük etki	Orta etki	Biraz etki	Etkisiz
Konutlarda enerji kullanımı (örneğin, bireysel evlerin ısıtılması için kömür ve odun)				
Tarım – çiftliklerden, gübrelerden ve tarımsal atıkların yakılmasından kaynaklanan emisyonlar				
Otomobil ve kamyonlardan kaynaklanan emisyonlar				
Uluslararası taşımacılıktan kaynaklanan emisyonlar (örneğin gemiler ve uçaklar)				
Endüstriyel üretimden (çelik, çimento, kağıt hamuru, kağıt vb.) ve fosil yakıtlı enerji santrallerinden kaynaklanan emisyonlar				
Manzara				
Nehirler / göller				
Temiz hava				
Diğer				

12. Ülkenizdeki hava kalitesine yönelik başlıca tehditlerin aşağıdaki emisyon yayıcılarından hangisi olduğunu düşünüyorsunuz?

☐ diğer ülkelerden/bölgelerden kaynaklanan sınır ötesi emisyonlar
☐ ulaşım faaliyetleri
☐ elektrik ve ısı üretimi
☐ doğal kirleticiler (deniz tuzu, çöl kumu, volkanik kül)
☐ endüstriyel faaliyetler
☐ bireysel hanelerden kaynaklanan emisyonlar
☐ çiftliklerden kaynaklanan emisyonlar
☐diğer ..... Lütfen belirtiniz.

13. Aşağıdaki araç yakıt sistemlerinden hangi ikisinin hava kalitesi açısından en çevre dostu olduğunu düşünüyorsunuz?

☐ benzin ☐ dizel ☐ biyoyakıt
☐ hibrit elektrikli/benzinli arabalar ☐ Hibrit elektrikli/dizel otomobiller
☐ electrikli arabalar ☐ diğer ..... Lütfen belirtiniz.

14. Evsel ısıtmaya yönelik aşağıdaki enerji sistemlerinden hangi ikisini hava kalitesi açısından en çevre dostu olarak değerlendiriyorsunuz?

☐ petrol ☐ doğalgaz ☐kömür ☐ bitki artıkları (odun)
☐ bitki artıkları (peletler) ☐ elektrik ☐ merkezi ısıtma
☐ diğer ..... Lütfen belirtiniz.

15. Havaya zararlı emisyonları azaltmanın farklı yolları vardır. Bu sorunları azaltmak için son iki yılda aşağıdakilerden herhangi birini yaptınız mı? Lütfen geçerli olanların tümünü seçin.

☐Konut ısıtma sisteminizi yüksek emisyonlu (ör. kömür, petrol veya odun yakıtlı) yerine daha düşük emisyonlu (ör. doğal gaz, pelet, elektrik) sistemle değiştirdiniz

☐ Enerji kullanan eski ekipmanlarınızı (sıcak su kazanı, fırın, bulaşık makinesi vb.) enerji verimliliği daha iyi olan yeni ekipmanlarla (ör. enerji verimliliği için A+++) değiştirdiniz

☐ Arabanız yerine sıklıkla toplu taşımayı, bisiklete binmeyi veya yürümeyi kullandınız

☐ Düşük emisyonlu bir araba satın aldınız

☐ Açık alandaki ateşinizi veya barbekünüzü yakmak için düşük emisyonlu ürünler satın aldınız (ör. kömür yerine briket)

☐ diğer ..... Lütfen belirtin.

16. Aşağıdaki sorunun ülkenizde çok ciddi bir sorun olduğunu, neredeyse ciddi bir sorun olduğunu, çok ciddi bir sorun olmadığını veya ciddi olmadığını mı düşünüyorsunuz?

	Çok ciddi bir sorun	Neredeyse ciddi bir sorun	Çok ciddi bir sorun değil	Hiç ciddi bir sorun değil
Solunum hastalıkları (örneğin akciğer hastalıkları)				
Kardiyo-vasküler hastalıklar (kalp hastalıkları)				
Astım ve alerji				
Asitlenme (asit yağmurları, ormanları etkileyen vb.)				
Ötrofikasyon (nehirlerde veya göllerde balıkların ölmesine neden olan alglerin aşırı büyümesi gibi bir ekosistemdeki organik madde artışı)				

17. Sizce aşağıdakilerden her biri ülkenizde iyi hava kalitesini teşvik etmek için çok fazla mı yapılıyor, uygun miktarda mı yapılıyor, yoksa yeterince mi yapılmıyor?

	Çok fazla yapılıyor	Uygun miktarda yapılıyor	Yeterince yapılmıyor	Bilmiyorum
Aileler				
Çiftçiler				
Enerji üreticileri				
Araba üreticileri				
Kamu yetkilileri				

18. Sizce hava kirliliği sorunları en iyi şekilde nasıl ele alınabilir?

☐ yerel düzeyde ☐ Ulusal düzeyde ☐ Avrupa düzeyinde

☐ diğer .....*Lütfen belirtiniz*

19. Geçen yıla kıyasla şu anda şehrinizde/kasabanızda/köyünüzdeki genel hava kalitesini nasıl değerlendirirsiniz?

☐ çok daha iyi ☐ biraz daha iyi ☐ neredeyse aynı ☐biraz daha kötü
☐ çok daha kötü ☐ diğer .....

20. Şehrinizdeki hava kirliliğinin ana nedenleri nelerdir? Lütfen geçerli olanların tümünü seçin.

☐ inşaat
☐ endüstriyel kaynaklar/üretim tesisleri
☐ motorlu taşıtlar
☐ evde yemek pişirme ve ısıtma
☐ klima kullanımının artması
☐ nüfus artışı
☐ enerji santralleri
☐ sigara dumanı
☐ atıkların yokedilmesi
☐ atıkların yakılması
☐ diğer bölgelerden kaynaklanan kirlilik
☐ diğer .....

21. Hava kirliliği sizi ne kadar etkiliyor?

☐ Nefes darlığı/nefes almada daha fazla zorluk yaşama
☐ Daha az açık hava etkinliği yapma
☐ Cildime bakmak için daha fazlasını yapmak
☐ Sağlıklı kalmak için daha fazlasını yapmak
☐ Depresyonda hissetmek
☐ Gözlerde/burunda/boğazda tahriş
☐ Cilt sorunları
☐ Daha az kirli başka yerlere taşınmak istemek
☐ Astım DURUMU
☐ Zayıf görünürlük
☐ Yaşam ortamı hakkında endişelenmek
☐ diğer ..... *Lütfen belirtiniz.*

22. Eviniz…….

☐ sakin bir bölgede, araç trafiğinin az olduğu bir yerde
☐ gürültülü bir alanda, yoğun araç trafiğinde
☐ trafik kaynağından farklı olması nedeniyle gürültülü bir alanda
☐ diğer ..... Lütfen belirtiniz.

23. Evinizdeki araba trafiğinden kaynaklanan egzoz gazlarının kokusunu alabiliyor musunuz?

☐ Evet, her gün ☐ Evet, bazen ☐ Nadiren
☐ diğer ..... *Lütfen belirtiniz.*

24. Evinizde araçlardan kaynaklanan kirlilik (gürültü, egzoz gazları vb.) ne ölçüde yaşıyorsunuz?

☐ çok yüksek ☐orta ☐ düşük ☐ diğer ..... *Lütfen belirtiniz..*

25Aileniz gece uyurken (trafik gürültüsünden uyanma) sorun yaşıyor mu?

☐ evet, çok sık ☐ sıksık ☐ nadiren
☐ diğer ..... *Lütfen belirtiniz..*

B bölümü – Çocuk sağlığına ilişkin özel bir bölüm
Lütfen tüm soruları okuyun ve kutuyu işaretleyerek veya uygun olduğu yerde kısa bir açıklama yaparak cevaplayın.
Anket anonimdir ve bireysel yanıtlarınızın gizliliğinin korunacağını garanti ederiz.

1. Cinsiyet
☐ erkek ☐ kadın

2. Yaş
☐ 3 yaş altı ☐ 3-7 yaş ☐ 8-15 yaş ☐ 15 yaş üstü

3. Çocuk ağırlığı
☐ doğumda …… …….. ☐ şimdi…….

4. Çocuğun doğumunda annenin yaşı
☐ 20 yaşa kadar ☐ 21-30 yaş ☐ 31-40 yaş
☐ 41-50 yaş ☐ 50 yaş üstü

5 .Bebek ne kadar süre emzirildi (ay olarak)?
☐ 1 aya kadar ☐ 1-3 ay ☐ 3-6 ay
☐ 6-9 ay ☐ 9-12 ay ☐diğer ..... *Lütfen belirtiniz*

6. Ailenizde sigara içen var mı? Kaç tane?
☐ 1 ☐ 2 ☐ 3 ☐ diğer ……..... *Lütfen anne olup olmadığını belirtin.*

7. Evinizde evcil hayvan var mı? Kaç tane?
☐ evet ☐ hayır ☐ diğer ..... *Lütfen belirtiniz*

8. Anne-babanız veya erkek/kız kardeşlerinizde alerjik hastalık olan var mı?
☐ evet ☐ hayır ☐ diğer ..... *Lütfen belirtiniz*

9. Çocuğunuzun (katılımcının) alerjik bir hastalığı var mı?
☐ evet ☐ hayır ☐ diğer ..... *Lütfen belirtiniz*

10. Çocuğunuz (katılımcı) şu ana kadar herhangi bir ciddi hastalık geçirdi mi (hastaneye yatma ihtiyacı oldu mu)?
☐ evet ☐ hayır ☐diğer ..... *Lütfen belirtiniz*

11. Çocuğunuz yılda dört kereden daha sık solunum yolu hastalıkları (burun akıntısı, bronşit, zatürre) geçiriyor mu?
☐ evet ☐ hayır ☐ diğer ..... *Lütfen belirtiniz*

12. Son altı ay içinde çocuğunuzda (katılımcıda) aşağıdaki belirtilerden bazılarını kaydettiniz mi?

	yes	no	I don't know
inatçı öksürük			
ıslık sesi hırıltı			
gece kuru öksürük			
saman nezlesi			
Nefes almada zorluk atakları (astım)			
Grip veya solunum sistemini etkileyen başka bir hastalık			
kanlanmış gözler (konjonktivit)			

E- MÜFREDAT

El kitabının bu bölümü "İleri Büyük Veri İşleme Teknolojileri" dersinin müfredatını temsil etmektedir. Bu müfredat Slovakya'nın Banská Bystrica kentindeki Matej Bel Üniversitesi Doğa Bilimleri Fakültesi ekibi tarafından sağlanmaktadır. Bu ortak bu kursu uygulamıştır ve tüm ortaklar bunu projenin sürdürülebilirliği sırasında uygulayacaktır.

Üniversite: Matej Bel Üniversitesi, Banská Bystrica, Slovakya
Fakülte: Doğa Bilimleri Fakültesi
Kod: DEK FPV/2d-fpv-401
Dersin Adı: Doğa Bilimlerinde İleri Büyük Veri İşleme Teknolojileri
Eğitim faaliyetlerinin türü, iş yükü ve yöntemleri: Ders türü: isteğe bağlı Önerilen iş yükü: Haftada 2 saat seminer Çalışma yöntemi: kombine Çalışma şekli: tam zamanlı Kredi sayısı: 3 Önerilen dönem: Yüksek Lisans çalışmalarının 2. dönemi
Eğitim derecesi: ikinci (Yüksek Lisans)
Önkoşullu dersler: Önkoşul yok
Kursu geçme ve tamamlama koşulları: a) sürekli değerlendirme: alıştırmalara aktif katılım, verilen görevlerin %100 tamamlanması b) son değerlendirme: %0 Konunun değerlendirilmesi UMB çalışma yönetmeliğinin belirlediği sınıflandırma ölçeğine göre yapılır.
Kazanımlar: 1. Öğrenciler aşağıdaki alanlarda bilgi ve beceri kazanacaklardır: 2. Veri işleme ve analize giriş 3. Temel veri analizi görevlerine giriş – regresyon ve sınıflandırma 4. Büyük Veri ile çalışmaya giriş – veri örnekleme yöntemleri 5. Veri analizinde istatistiksel yöntemler 6. Keşif Amaçlı Veri Analizinin Temelleri – teori ve uygulama 7. Bulanık kümelere giriş 8. Bulanık kümeler ve regresyon görevi 9. Bulanık kümeler ve sınıflandırma görevleri 10. Sinir Ağlarına Giriş 11. Kurs sırasında öğrenci aşağıdaki konularda çalışma deneyimi kazanacaktır: 12. MATLAB yazılım aracı 13. R yazılım aracı

KAYNAKLAR

1 - 4. Bölümler için Kaynaklar

- C.J. Date. An Introduction to Database Systems (8th. ed.). Addison-Wesley Longman Publishing Co., 2003. ISBN: 978-0-321-19784-9
- Felix Kutsanedzie, Sylvester Achio, Edmund Ameko. Practical Approaches to Measurements, Sampling Techniques and Data Analysis. Science Publishing Group, 2016. ISBN: 978-1-940366-58-6.
- William J. Lammers, Pietro Badia. Fundamentals of Behavioral Research Textbook. Online: https://uca.edu/psychology/fundamentals-of-behavioral-research-textbook/
- Jimin Quian et al. Introducing self-organized maps (SOM) as a visualization tool for materials research and education. Results in Materials, Volume 4, 2019, ISSN 2590-048X.
- Naseer Raheem. Big Data: A tutorial-based approach. Chapman and Hal-l/CRC, 2019. ISBN: 978-0-367-67024-5
- Lior Rokach, Oded Maimon. Data mining with decision trees. 2015.
- Steven S. Skiena. The Data Science Design Manual. Springer, 2017. ISBN: 978-3-319-55443-3
- Karthik Ramasubramanian, Abhishek Singh. Machine Learning Using R. Springer, 2019. ISBN: 978-1-4842-4214-8
- Patrik Očenáš. Parallel and distributed methods of big data sampling (in Slovak). 2023.
- Bianka Modrovičová. Decision trees for sizable graph datasets (in Slovak). 2023.
- Aneta Szoliková. Explorative data analysis in document databases (in Slovak). 2023.
- Adam Dudáš, Bianka Modrovičová. Decision Trees in Proper Edge k-coloring of Cubic Graphs. In Proceedings of 33rd FRUCT conference. 2023.

5 - 8. Bölümler için Kaynaklar

- ZADEH, L. A. Fuzzy Sets. In: Information and Control, 8, 1965, 338-353.
- MICHALÍKOVÁ, A.: Fuzzy množiny v informatike. rec. Mirko Navara, Martin Kalina, Martin Klimo. Belianum. Matej Bel University in Banská Bystrica, 1, 2020, 206p. ISBN 978-80-557-1707-4
- Sendai Subway. Japan Visitor [cit. 2023-02-02]. Online: https://www.japanvisitor.com/japan-transport/sendai-subway
- RUAN D.: Fuzzy Logic Applications in Nuclear Industry. Fuzzy Logic Foundations and Industrial Applications. 1996, 8, ISBN 978-1-4612-8627-1.
- TAKAGI, T., SUGENO, M. Fuzzy Identifications of Fuzzy Systems and its Applications to Modelling and Control. In: IEEE Transactions on Systems, Man, and Cybernetics, 15(1), 1985, 116-132.
- ROSS, T. J. Fuzzy Logic with Engineering Applications. John Wiley & Sons, 2005, 585s., ISBN 9780470743768.
- ZADEH, L. A., The Concept of a Linguistic Variable and its Application to Approximate Reasoning - 1, In: Information Sciences, 8, 1975, 199–249.

9. Bölüm için Kaynaklar

- Ahmed, Z. H. (2010). Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator. International Journal of Biometrics & Bioinformatics (IJBB), 3(6), 96.
- Aktaş, M., Yetgin, Z., Kılıç, F., & Sünbül, Ö. (2022). Automated test design using swarm and evolutionary intelligence algorithms. Expert Systems, 39(4), e12918.
- Bartz-Beielstein, T., Branke, J., Mehnen, J., & Mersmann, O. (2014). Evolutionary algorithms. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(3), 178-195.
- Bertsimas, D., & Tsitsiklis, J. (1993). Simulated annealing. Statistical science, 8(1), 10-15.
- Blickle, T. (2000). Tournament selection. Evolutionary computation, 1, 181-186.
- Cui, Y., Geng, Z., Zhu, Q., & Han, Y. (2017). Multi-objective optimization methods and application in energy saving. Energy, 125, 681-704.
- De La Iglesia, B. (2013). Evolutionary computation for feature selection in classification problems. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3(6), 381-407.
- Gaivoronski, A. A., Lisser, A., Lopez, R., & Xu, H. (2011). Knapsack problem with probability constraints. Journal of Global Optimization, 49, 397-413.
- Glover, F., & Laguna, M. (1998). Tabu search (pp. 2093-2229). Springer US.
- Hansen P, Mladenovi'c N (1999) An introduction to variable neighborhood search. In: Voß S, Martello S, Osman IH, Roucairol C (eds) Metaheuristics: advances and trends in local search paradigms for optimization, chapter 30. Kluwer Academic Publishers, Dordrecht, pp 433–458
- Hayyolalam, V., & Kazem, A. A. P. (2020). Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems. Engineering Applications of Artificial Intelligence, 87, 103249.

- Hinson, J. M., & Staddon, J. E. R. (1983). Matching, maximizing, and hill-climbing. Journal of the experimental analysis of behavior, 40(3), 321-331.
- Holland JH. Outline for a logical theory of adaptive systems. J ACM. 1962;9(3):297–314
- Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. SIAM journal on computing, 2(2), 88-105.
- Hoos, H. H., & Stützle, T. (2004). Stochastic local search: Foundations and applications. Elsevier.
- I. Rechenberg, Cybernetic solution path of an experimental problem. Royal Air-craft Establishment, Library Translation 1122, Farnborough, Reprint in: D.B. Fogel (Ed.), Evolutionary Computation, The Fossil Record, IEEE Press, Piscataway, NJ, 1965, pp. 301–309
- I. Rechenberg, Evolutionsstrategie—Optimisierung technischer Systeme nach Prinzipien der biologischen Evolution, Frommann-Holzboog, Stuttgart, 1973
- Kılıç, F., Yılmaz, İ. H., & Kaya, Ö. (2021). Adaptive co-optimization of artificial neural networks using evolutionary algorithm for global radiation forecasting. Renewable Energy, 171, 176-190.
- Kiliç, F., & Gök, M. (2013). A public transit network route generation algorithm. IFAC Proceedings Volumes, 46(25), 162-166.
- Li, X., Tang, K., Omidvar, M. N., Yang, Z., Qin, K., & China, H. (2013). Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. gene, 7(33), 8.
- Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. Knowledge-based systems, 96, 120-133.
- Rossi, F., Van Beek, P., & Walsh, T. (Eds.). (2006). Handbook of constraint programming. Elsevier.
- Salkin, H. M., & De Kluyver, C. A. (1975). The knapsack problem: a survey. Naval Research Logistics Quarterly, 22(1), 127-144.
- Sharifi, A. A., & Aghdam, M. H. (2019). A novel hybrid genetic algorithm to reduce the peak-to-average power ratio of OFDM signals. Computers & Electrical Engineering, 80, 106498.
- Wang, L., Cao, Q., Zhang, Z., Mirjalili, S., & Zhao, W. (2022). Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. Engineering Applications of Artificial Intelligence, 114, 105082.
- Yang, J., & Soh, C. K. (1997). Structural optimization by genetic algorithms with tournament selection. Journal of computing in civil engineering, 11(3), 195-200.

10. Bölüm için Kaynaklar

- Basic Neural Networks 1 - https://docs.google.com/a/atu.edu.tr/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxpaHNhbnlhc3NpbjJ8Z3g6NGY4MjNjN2Y4ZTdhNWM2MQ
- Basic Neural Networks 2 - http://www.cs.stir.ac.uk/courses/ITNP4B/lectures/
- Basic Neural Networks 3

  https://www.cs.bham.ac.uk/~jxb/inn.html
- Basic Neural Network 4

  https://www.fer.unizg.hr/en/course/neunet_a/lecture_notes
- Basic Neural Network 5

  http://users.monash.edu/~cema/courses/FIT3094/lecturePDFs/

11. Bölüm için Kaynaklar

- Paluszek, M., Thomas, S. Matlab machine learning recepies. 2019. Plainsboro, NJ, USA. ISBN-13 (pbk): 978-1-4842-3915-5. DOI 10.1007/978-1-4842-3916-2.
- Kim, P. MATLAB Deep Learning. With Machine Learning, Neural Networks and Artificial Intelligence. 2017. Apress Korea ISBN-13 (pbk): 978-1-4842-2844-9. DOI 10.1007/978-1-4842-2845-6.
- Get Started with Matlab. https://www.mathworks.com/help/matlab/getting-started-with-matlab.html
- Iris Clustering. https://www.mathworks.com/help/deeplearning/ug/iris-clustering.html

Ekler için Kaynaklar

- Fisher, R.A. (1936) "The use of multiple measurements in taxonomic problems". Annual Eugenics, 7, Part II, pages 179-188
- Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule". IEEE Transactions on Information Theory, May 1972, pages 431-433
- Duda, R.O., Hart, P.E. (1973) Pattern Classification and Scene Analysis. (Q327.D83) John Wiley & Sons. ISBN 0-471-22361-1, page 218
- Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recogni-

tion in Partially Exposed Environments". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, pages 67-71
- https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-3/data-products
- https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-4/data-products
- https://climexp.knmi.nl/
- https://www.uradmonitor.com/
- Velea L, Udriștioiu MT, Puiu S, Motișan R, Amarie (2023)D. A Community-Based Sensor Network for Monitoring the Air Quality in Urban Romania. Atmosphere; 14(5):840. https://doi.org/10.3390/atmos14050840
- https://bookdown.org/floriandierickx/bookdown-demo/climate-data-from-models.html#differences-between-climate-projections-predictions-and-scenarios
- https://ec.europa.eu/eurostat/web/climate-change/database
- https://ourworldindata.org/
- https://ourworldindata.org/data-review-air-pollution-deaths
- •https://ourworldindata.org/outdoor-air-pollution#outdoor-air-pollution-deaths-by-age
- •https://www.who.int/health-topics/air-pollution#tab=tab_1
- •https://www.eea.europa.eu/en/topics/in-depth/air-pollution
- •https://www.who.int/teams/environment-climate-change-and-health/air-quality-and-health/health-impacts/types-of-pollutants
- https://www.who.int/publications/i/item/9789240034228
- •https://apps.who.int/iris/bitstream/handle/10665/345329/9789240034228-eng.pdf
- EEA, 2012, The contribution of transport to air quality, EEA Report no. 10/2012, European Environment Agency.
- EEA. A closer look at urban transport TERM 2013: transport indicators tracking progress towards environmental targets in Europe EEA Report No 11/2013 Copenhagen, ISSN 1725-9177.
- http://dx.doi.org/10.1016/j.envpol.2007.06.012
- https://www.who.int/health-topics/air-pollution#tab=tab_1
- Report no. 05/2022, Air quality in Europe 2022. doi: 10.2800/488115. https://www.eea.europa.eu/publications/air-quality-in-europe-2022
- Xin Zhang, X. Chen, Xiaobo Zhang. The impact of exposure to air pollution on cognitive performance. Proc. Natl. Acad. Sci. Unit. States Am., 115 (2018), pp. 9193-9197, 10.1073/pnas.1809474115
- J. Currie, J.S.G. Zivin, J. Mullins, M.J. Neidell. What do we know about short and long term effects of early life exposure to pollution? NBER Work. Pap., 6 (2013), pp. 217-247, 10.3386/w19571
- Escamilla-Nuñez M-C., Barraza-Villarreal A., Hernandez-Cadena L., Moreno-Macias H., Ramirez-Aguilar M., Sienra-Monge J-J., Cortez-Lugo M., Texcalac J-L., del Rio-Navarro B., Romieu I. Traffic-Related Air Pollution and Respiratory Symptoms Among Asthmatic Children, Resident in Mexico City: The EVA Cohort Study. http://www.medscape.com/viewarticle/585875.
- Juvin P., Fournier T., Boland S. et al. Diesel particles are taken up by alveolar type II tumor cells and alter cytokines secretion. Arch Environ Health. 2002; 57(1):53-60.
- Le Tertre A., S. Medina, E. Samoli et al: Short term effects of particulate air pollution on cardiovascular disease in eight European cities. J. Epidemiol Community Health, 2002; 56, (10):773-9.
- Nordling E., Berglind N., Melén E., Emenius G., Hallberg J., Nyberg F., Pershagen G., Svartengren M., Wickman M., Bellander T. Traffic related air pollution and childhood respiratory symptoms, function and allergies. Epidemiology. 2008; 19(3):401-8.
- Pan G., Zhang S., Feng Y., Takahashi K., Kagawa J., Yu L., Wang P., Liu M., Liu Q., Hou S., Pan B., Li J. Air pollution and children's respiratory symptoms in six cities of Northern China. Respiratory Medicine 2010;104(12):1903–11.
- Richardson E.A., Pearce J., Tunstall H., Mitchell R., Shortt N.K.: Particulate air pollution and health inequalities: a Europe-wide ecological analysis. Int J Health Geogr 2013;12:34
- I. Jáuregui, J. Mullol, I. Dávila, M. Ferrer, J. Bartra, A. Del Cuvillo, J. Montoro, J. Sastre, A. Valero. Allergic rhinitis and school performance. J Investig. Allergol. Clin. Immunol., 19 (2009), pp. 32-39
- D.P. Skoner. Allergic rhinitis: definition, epidemiology, pathophysiology, detection, and diagnosis. J. Allergy Clin. Immunol., 108 (2001), pp. 2-8, 10.1067/mai.2001.115569
- I. Beck, S. Jochner, S. Gilles, M. McIntyre, J.T.M. Buters, C. Schmidt-Weber, H. Behrendt, J. Ring, A. Menzel, C. Traidl-Hoffmann. High environmental ozone levels lead to enhanced allergenicity of birch pollen. PloS One, 8 (2013), 10.1371/journal.pone.0080147
- P. Sturdy, S. Bremner, G. Harper, L. Mayhew, S. Eldridge, J. Eversley, A. Sheikh, S. Hunter, K. Boomla, G. Feder, K. Prescott, C. Griffiths. Impact of asthma on educational attainment in a socioeconomically deprived population: a study linking health, education and social care datasets. PloS One, 7 (2012), pp. 1-8, 10.1371/journal.pone.0043977
- https://europa.eu/eurobarometer/surveys/detail/2660
- https://data.europa.eu/data/datasets/s2660_97_2_sp524_eng?locale=en
- https://www.surveymonkey.com/r/airpollutionperceptionsurvey
- https://apps.who.int/iris/rest/bitstreams/1350812/retrieve
- https://www.ab.gov.tr/files/ardb/evt/Attitudes_of_Europeans_towards_air_quality_2013.pdf