VERİ MADENCİLİĞİ

- **Veri madenciliği**, veriden örüntülerin çıkarılması amacıyla çeşitli algoritmaların uygulanmasıdır. Elde edilen örüntü ve kurallar karar vermeye ve bu kararların sonuçlarını tahmin etmeye destek olacak biçimde kullanılabilecektir.
- **Veri madenciliğinin ortaya çıkışı**, büyük miktarda veriyi analiz edebilme ve işleyebilme ihtiyacından kaynaklanmıştır.
- **Veri madenciliğinin amacı**, çok büyük miktarda ve karmaşık durumdaki veriler içinden geleneksel yöntemlerle elde edilemeyecek bilgilere ulaşma ve bu bilgileri rakiplere fark yaratacak kararlarda kullanabilmeye olanak sağlamaktır.
- **Perseptron**, insan beyninde yer alan **sinir hücrelerinin (nöronların) ilk yapay modeli**ne verilen isim olup algılayıcı, fark edici anlamındadır. **1957** yılında **Frank Rosenblatt** tarafından geliştirilen ve **tekrar eden, benzerlik gösteren özelliklerin bilgisayar tarafından algılanabilmesini sağlayan** bir algoritmadır.
- **İlk veri modelleri;** Hiyerarşik Veri Modeli ve Ağ Veri Modeli olarak adlandırılan basit veri modelleridir.
- Veri Madenciliği için **1989**'da yapılan **KDD (Knowledge Discovery in Database) IJCAI-89 Veri Tabanlarında Bilgi Keşfi** Çalışma Grubu toplantısı önemli bir gelişme olmuştur. Süreç devam etmiş ve Veri Madenciliği ile ilgili ilk yazılım **1992** yılında geliştirilmiştir.
- **Veri Madenciliğinin Etkileşimde Olduğu Disiplinler**
  - **Makine öğrenimi**, kısaca bilgisayarların bazı işlemlerden çıkarsamalar yaparak yeni işlemler üretmesi olarak tanımlanabilir. Makine öğrenimi, insan öğrenmesinde söz konusu olan özelliklerin algoritmalar yardımıyla bilgisayarlara da uygulanabileceği ve bilgisayarların da insanlar gibi öğrenebileceği düşüncesini temel alan bir disiplindir.
  - **İstatistik**, verilerin analizi ve değerlendirilmesi konusunda geçmişten günümüze yoğun bir biçimde kullanılan bir disiplindir.
  - **Görselleştirme**, verilerin, tablolar ve grafikler gibi görseller yardımıyla sunulmasını sağlayan teknolojileri ifade eder. Verilerin daha kolay anlaşılmasına, analiz edilmesine ve geleceğe yönelik tahminlerde bulunulmasına önemli katkı sağlamaktadır.
  - **Veritabanı Sistemleri,** kısaca veritabanı tanımlamak, oluşturmak, veritabanında işlem yapmak, veritabanının farklı kullanıcı yetkilerini belirlemek, veritabanının bakımını ve yedeklemesini yapmak için geliştirilmiş programlar bütünüdür.
  - **Örüntü tanıma,** olaylar ve nesneler arasında daha önceden tanımlanmış, **düzenli ve sistematik biçimde tekrar eden ilişkileri** bir model olarak kabul eden ve bu modelin (örüntünün) benzerlerini ya da en benzerini veritabanı içinden arama ve bulmaya yönelik teknolojidir.
- **Veri**, ham gözlemler, işlenmemiş gerçekler ya da izlenimlerdir. **Enformasyon**, verinin bir anlam oluşturacak şekilde düzenlenmiş hâlidir. **Bilgi** ise en yalın tanımıyla verinin işlenmiş ve dönüştürülmüş halidir.
- **Veri ambarı**, işletmelerde **iç veri kaynakları ile dış veri kaynaklarının birleştirilmesi ve düzenlenmesi ile oluşturulmuş**, üzerinde veri madenciliği işlemlerinin gerçekleştirileceği veriyi sağlayan daha geniş ve özel veritabanlarına verilen isimdir. İngilizce karşılığı **metadata** olan **üst veri**, veri ambarında yer alan veriler hakkındaki tanımlamalar olup veri ambarına ilişkin **veri kataloğu** olarak düşünülebilir.
- **OLAP (Online Analytical Processing - Çevrimiçi Analitik İşleme)** veri ambarında yer alan veriler üzerinde çok boyutlu, çok yönlü analiz ve sorgulama yapılmasını sağlayan sistemlerdir.
- **Bilgi Keşfi sürecinde izlenmesi gereken temel aşamalar**
  - **Amacın Tanımlanması:** İşletmenin ya da kurumun veri madenciliğini hangi amaca yönelik olarak gerçekleştirmek istediği belirlenir.
  - **Veriler Üzerinde Ön İşlemlerin Yapılması**
    - Verilerin toplanması ve birleştirilmesi
    - Verilerin temizlenmesi
    - Verilerin yeniden yapılandırılması
      - Verilerin normalizasyonu.
      - Verilerin azaltılması.
      - Verilerin dönüştürülmesi.
  - **Modelin Kurulması ve Değerlendirilmesi:** En uygun modelin belirlenmesi için, çok sayıda modelin denenmesi gerekebilir.
  - **Modelin Kullanılması ve Yorumlanması:** Kurulan modellerden elde edilen sonuçlar değerlendirilerek yorumlanmalıdır.
  - **Modelin İzlenmesi:** Sistemin ne kadar iyi çalıştığının sürekli olarak izlenmesi ve ölçülmesi bir gerekliliktir.
- **Kayıp veri:** Veritabanlarındaki kayıtlarda eksik olan verilerdir.
- **Gürültülü veri:** Veritabanlarında doğru olmayacak kadar uç değerler, aykırı değer ya da sıra dışı değer olarak tanımlanır.
- **Kayıp veri problemini çözmek için kullanılan yaklaşımlar**
  - Kayıp veri içeren kaydı veri kümesinden çıkarmak.
  - Kayıp verileri tek tek yazmak.
  - Kayıp verilerin hepsi için aynı veriyi girmek.
  - Kayıp veri yerine tüm verilerin ortalama değerinin girilmesi.
  - Kayıtlarda yer alan diğer değişkenler yardımıyla kayıp verilerin tahmin edilmesi. (regresyon, zaman serisi, bayesian vb. ile)
- **Gürültülü veri problemini çözmek için kullanılan yaklaşımlar**
  - Bölümleme yöntemiyle gürültünün temizlenmesi.
  - Sınır değerleri kullanılarak gürültünün temizlenmesi.
  - Kümeleme yöntemiyle düzeltme yapılması ve gürültünün temizlenmesi.
  - Regresyon yöntemiyle düzeltme yapılması ve gürültünün temizlenmesi.
- **Veri İndirgeme:** Verilerin azaltılması, veri kümesi içinde gereksiz olduğu düşünülen verinin kaldırılması biçiminde olabileceği gibi daha çok birden fazla değişkenin birleştirilerek tek bir değişkenle ifade edilmesi biçiminde de gerçekleştirilir.
- **Veri Madenciliği yazılımları**, kendilerine verilen örnek veriler üzerinde inceleme yaparak kullandıkları algoritmalarla bu verilerden bazı sonuçlar ve kurallar çıkarırlar. Buna **öğrenme** denir.

- **Aşırı öğrenme**, algoritmanın çıkardığı kuralların sadece üzerinde çalıştığı veriler için geçerli olmasını, dışarıdan başka verilere uygulandığında ise geçersiz olması durumunu ifade eder.
- **Veri madenciliğinde kullanılan modeller ikiye ayrılır:**
  - **Tahmin edici modeller:** Eldeki verilerden hareketle bir model geliştirilmesi ve geliştirilen bu model kullanılarak önceden sonuçları bilinmeyen veri kümeleri için sonuçların tahmin edilmesini amaçlar.
    - **Regresyon Modelleri:** Bilindiği gibi regresyon, bağımsız değişkenler ile bağımlı değişkenler arasındaki ilişkiyi en iyi tanımlayan fonksiyonu elde etmek için uygulanan istatistiksel tekniktir.
    - **Sınıflandırma Modelleri:** Verinin sınıfını tanımlama ve ayırt etmeyi sağlayan bir model kümesini bulma sürecidir.
  - **Tanımlayıcı modeller:** Bu modeller tahmin edici modellerin aksine, analiz edilen verilerin özelliklerini incelemek için kullanılan modellerdir. Veritabanındaki **kayıtlar arasında bir bağlantı, ilişki kurulmaya çalışılır**. Böylece bir veritabanındaki kayıtlar arasında çok rastlanan kurallar ortaya çıkarılır.
- **Denetimli öğrenmede** ilgili veriler seçilen algoritmaya uygun olarak hazırlandıktan sonra, ilk aşamada verinin bir kısmı modelin öğrenimi, diğer kısmı ise modelin geçerliliğinin test edilmesi için ayrılır. Modelin öğrenimi, öğrenim kümesi kullanılarak gerçekleştirildikten sonra, test kümesi ile modelin doğruluk derecesi belirlenir.
- **Denetimsiz öğrenmede**, kümeleme analizinde olduğu gibi ilgili örneklerin gözlenmesi ve bu örneklerin özellikleri arasındaki benzerliklerden hareket ederek sınıfların tanımlanması amaçlanmaktadır.
- **Regresyon ve sınıflandırma modellerinden en yaygın kullanılanlar;**
  1) **Karar ağaçları**, 2) **Yapay sinir ağları**, 3) **Genetik algoritmalar**, 4) **Zaman serisi analizi**, 5) **K-en yakın komşu**, 6) **Bayes sınıflandırması**.
- **En yaygın kullanılan tanımlayıcı modeller**;
  1) **Kümeleme**, 2) **Birliktelik kuralları**, 3) **Sıra örüntü analizi**, 4) **Özetleme**.
- **Veri Madenciliğinde keşfedilebilen bilgi türleri**
- Seçilen kayıtlara ait **ortalama ve toplam değer** gibi özet bilgiler **sığ bilgi** olarak tanımlanır. (SQL)
- **Farklı özelliklerin** ortaya çıkma sıklığı hakkındaki bilgi **çok boyutlu bilgi** olarak nitelendirilir. (OLAP)
- **Önceden tahmin edilemeyen** örüntü ve ilişkiler **gizli bilgi** olarak ifade edilebilir.
- Sadece **önsel teknik veya meta bilginin** kullanımıyla keşfedilebilecek gizli örüntüler ve ilişkiler hakkında bilgi ise **derin bilgi**'dir.
- R programlama dili, **Becker and Chembers** tarafından geliştirilen S dilinin bir çeşidi dir.
- R yazılımında, komutların girilmesi için kullanılan bölgeye "**R Console**" denir. Komut satırlarında yer alan **>** işareti komut satırının kendisini temsil eder. Herhangi bir atama yapılması ya da matematiksel bir ifadenin hesaplanması için en basit komutlar olarak meydana çıkan komutlar grubuna **temel komutlar** denir.
- **R dilinde değişken** atama işlemi için "**değişken <- işlem**" yapısı kurulmalıdır. Örnek: > x <- 72+45
- **Bir fonksiyonun nasıl çalıştığı hakkında yardım almak** için **help(fonksiyonismi)** komutu kullanılır.
- **R dilinde bir vektörü en basit şekilde oluşturmak** için **c(,,,)** fonksiyonu kullanılır. Örnek: > x <- c(1,2,3,4,5)
  Vektördeki belirli bir elemana ulaşmak için indeks kullanılabilir. İndeks numaraları **1**'den başlar.
  > isimler <- c("ali","veli","ayşe","fatma")
  > isimler[3] # bu satırdaki komut ekrana ayşe yazar, ayşe elemanı isimler vektörünün 3. elemanıdır.
- Önceden tanımlanmış bir vektörün birim sayısını öğrenmek için **length(vektor)** fonksiyonu kullanılır.
- Belirli bir düzene sahip vektör oluşturulmasında **seq(altlimit, üstlimit, artışmiktarı)** fonksiyonu kullanılır. Artış miktarı girilmezse 1'dir.
- Tekrarlanan verilerin oluşturulması için **rep(istenendüzen, tekrarsayısı)** fonksiyonu kullanılır. Örnek: > rep(1:5,4) veya > rep(seq(1,5),4)
- Bir değişkenin karekökü **sqrt()** fonksiyonu yardımıyla hesaplanabilir.
- **Matris oluşturmak için matrix(veri, nrow(satırsayısı), ncol(sütunsayısı), byrow=F(veri sütun olarak girilsin))** fonksiyonu kullanılır.
  Örnek: > matrix(c(6,5,4,3,2,1,1,2,3,4,5,6), ncol=2)

  Öncelikle satırlar veri kısmında tanımlanıyor ise **byrow=T** parametresinin kullanılması gerekir.
  Örnek: > veri <- c(6,5,4,3,2,1,1,2,3,4,5,6)
  > matrisim <- matrix(veri, ncol=2, byrow=T)

  Matrisin ikinci satır, ikinci sütun elemanı **matris[satır, sütun]** şeklinde bulunur. Örnek: > matrisim[2,2]
  Matrisin herhangi bir satırına **[satırno,]** ve herhangi bir sütununa da **[,sütunno]** köşeli parantezleri yardımıyla ulaşılabilir.
  Örnek: > matrisim[1,]
  > matrisim[,2]

  Matris elemanlarına aritmetik işlemler uygulanabilir. Bu işlemler tek tek karşılıklı matris elemanlarına uygulanır.
  Örnek: > matris2 <- matrix(c(2,7,1,4), ncol=2, byrow=T)
  > matris2 + matris2
  > matris2 * matris2
  > matris2 - matris2

  Matris çarpımı yapmak için ise **%*%** ifadesi kullanılır. matris2 * matris2 işleminden farklıdır!
  Örnek: > matris2 **%*%** matris2
- **Matrisin devriğ**inin (transpoz) elde edilmesi için **t(matris)** fonksiyonu kullanılır. Örnek: > t(matris2)
- **Belli bir değere göre vektör elemanlarına ulaşmak:** > x[x>20] # x vektöründeki 20'den büyük elemanlara ulaşılır.

- **R dilinde Mantıksal Operatörler**

Operatör	Kullanımı
<	Küçüktür
>	Büyüktür
<=	Küçük ya da eşittir
>=	Büyük ya da eşittir
==	Eşittir
!=	Eşit değildir
&	Ve
|	Veya
!	Değil

Örnekler: > değişken<-23 == 4
> değişken<-123 < 514

- **Mantık fonksiyonları** yardımı ile ilgilenilen değişkenin bir karakter değişkeni mi yoksa sayısal bir değişken mi olduğu anlaşılabilir.
  > is.character("Kuzey") # TRUE
  > is.numeric(27) # TRUE
  > is.infinite(25/0) # TRUE
- Sıralama işlemi için **sort(vektör)** fonksiyonu kullanılır.
  > x<-c(23,12,43,37,11)
  > sort(x) # vektör elemanları küçükten büyüğe doğru sıralanır.
- Çeşitli istatistiksel analizler için oluşturulan farklı nesnelerin bir araya getirilmesinde **List Nesneleri**nden faydalanılır.
  > birlikte<-list(matris1,matris2)
  > birlikte # iki matris de ekranda görüntülenir.

  Bu noktadan sonra sadece matris2'ye ulaşmak istersek,
  > birlikte[[2]] # ekrana sadece matris2 yazılır.

  List değişkenleri oluşturulurken bu değişken içerisinde yer alan elemanlar isimleriyle de atanabilir.
  > birlikte<-list(veri=matris1,korelasyon=matris2)
  > birlikte # iki matris de ekranda görüntülenir.

  Bu noktadan sonra sadece matris2'ye ulaşmak istersek $ işaretini kullanmamız gerekir,
  > birlikte$korelasyon # ekrana sadece matris2 yazılır.
- R yazılımında veri seti içerisindeki **faktör listeleri ve gözlem birimleri** **data frame** olarak bir araya getirilirler.
  "data.frame" fonksiyonunda **her sütunda eşit** sayıda birim yer alır.
  > bilgisayar <- c(45,25,12,21,42,32,14,54)
  > matematik <- c(34,65,76,12,37,83,90,48)
  > isim <- c("Tuncay", "Serhat", "Volkan", "Rüştü", "Ümit", "Önder", "Zafer", "Selçuk")
  > öğrencinot <- data.frame(isim, bilgisayar, matematik)
  > öğrencinot

  List nesnelerinde olduğu gibi istenen sütun $ işareti yardımıyla görüntülenebilir.
  Örneğin; öğrencilerin matematik notları için betimleyici istatistikler şu şekilde görüntülenebilir.
  > **summary**(öğrencinot**$**matematik)
- Eğer herhangi bir gözlem biriminin değeri "kayıp değer" ise bu işlem gözlem birimine **NA** değerinin atanması ile gerçekleştirilir.
- R yazılımında **kişisel fonksiyon oluşturmak için** **function(parametre) { … }** yapısı kullanılır.
- Standart sapma (varyans) **var()** komutu ile hesaplanır. Artimetik ortlama **mean()** fonksiyonu ile hesaplanır.
- Histogram grafiği **hist()** ile kutu grafiği de **boxplot()** komutu ile çizdirilir.
- **Verinin okutulabilmesi için** bir kaç farklı teknik bulunur:
  - **scan()** düşük seviyeli veri okutma işlemi,
  - **read.table()** dosyalardan formatlanmış data frame elde edilmesi işlemi,
  - **read.fwf()** belirgin bir genişlik tanımlanmış veri dosyalarından okuma işlemi,
  - **read.csv()** değişkenlerin virgülle veya noktalı virgülle ayrıldığı dosyalardan okuma işlemi.
    > **read.csv**("c:\\veriseti.csv",header=T,sep=";")
- **library(kütüphaneismi)** fonksiyonu yardımıyla R yazılımındaki yüklü kütüphanelerden faydalanılabilir.
- **Verinin elde edilme yolları:** Veri; insan tarafından oluşturulmuş bir bilgisayar dosyasından, verileri tasarlamak ve yönetmek için kullanılan bir işletme veri tabanı yönetim sisteminden, standart bir veri tabanı sisteminden, otomatik bilgi kaydı oluşturan bir araçtan, uydu üzerinden ve bunlara benzer şekilde kaynaklardan gelmiş olabilir.
- **Özellik (Feature)**, hakkında bilgi edinilmek istenen canlı, cansız varlıklar veya olayların sahip oldukları ve birbirinden ayırt edilmesine yardımcı olan değişkenler veri madenciliğinde bir veri setinin sunumunda kullanılan tablo gösteriminde sütunlarda yer alır.

- Veri madenciliğinde bir veri setinin sunumunda kullanılan tablo gösteriminde satırlarda **nesneler** yer alır.
- Birimlerin sahip olduğu özelliklerin **derecesinin belirlenerek** sonuçların sayısal olarak ifade edilmesine **ölçme** adı verilir.
- **Sınıflayıcı ölçek**, gözlem değerlerinin tek tek nitel kategori ya da sınıra atanması sonucu oluşan ölçektir.
- **Temel Değişken Tipleri**
  - **Kategorik Değişkenler:**
    - **İsimsel (Nominal) Değişkenler:** Sayısal bir formda olabilir. Yani isimleri olan 5 kişi; 1, 2, 3, 4, 5 olarak sayılarla ifade edilebilir. Ancak bu sayısal değer matematiksel bir hesaplama ya da işlem yapmak için uygun değildir. Sayılar sadece bir etiket görevi görecektir.
    - **İkili (Binary) Değişkenler:** İsimsel değişkenlerin özel bir şekli olan ikili değişkenler 0 ve 1, doğru ve yanlış, pozitif ve negatif, cinsiyet özelliğinde olduğu gibi erkek ve kadın gibi sonuçları sadece iki şekilde ortaya çıkan değişkenlerdir.
    - **Sıra Gösteren (Ordinal) Değişkenler:** Bu değişken tipi de isimsel değişken tipine benzerdir. Ancak değişkenin almış olduğu değer, derecesi bakımından sıraya dizilmesinde önemlilik gösteriyorsa sıra gösteren değişken söz konusu olur.
  - **Sürekli Değişkenler:**
    - **Tam Sayılı (Integer) Değişkenler:** Alacağı değerler **0**, **1**, **2**, ... gibi tamsayılar olarak belirtilebilen değişkenlerdir.
    - **Aralıklı Ölçümlendirilmiş (Interval-Scaled) Değişkenler:** Sıra gösteren **(ordinal) değişkenin tüm özelliklerini içerir** ve birimler arasında özellik farkları matematiksel olarak belirlenebilir. Kullanılan ölçüm için belirli bir **yokluk anlamına gelmeyen sıfır** ölçme düzeyi bulunabilir. Örneğin; hava sıcaklığı niceldir ve yokluk anlamına gelmeyen sıfır değeri bulunabilir.
    - **Oranlı Ölçümlendirilmiş (Ratio-Scaled) Değişkenler:** Aralıklı ölçümlendirilmiş (interval-scaled) değişkenlere benzer olmakla beraber bu değişkende sıfır başlangıç noktası tüm ölçüm araçlarında aynı anlamı taşır. Örneğin; bir varlığın ağırlığı için "sıfır" ifadesi kullanıldığında ölçüm metrik türüne bakılmadan bu varlığın ağırlığının olmadığı anlamı çıkarılır. **0 kg = 0 gr = 0 ton**
- **Veri hazırlama aşamasında,** veri kalitesini anlamak ve iyileştirmek, veri madenciliği **çıktı kalitesini** arttırır.
- **Veri Hazırlama Süreçleri**
  - **Veri Temizleme:** Verideki eksiklik, gürültü ve tutarsızlığın giderilmesi için uygulanır.
  - **Veri Birleştirme:** Çoklu kaynaklardan gelen verinin uygun bir veri ambarında birleştirilmesidir. Veri birleştirmede 1) şema birleştirmesi, 2) fazla veri sorunları ve 3) veri değer karmaşalarının belirlenmesi ve çözümlenmesi, şeklinde üç konu vardır.
  - **Veri İndirgeme:** Çok daha küçük hacme indirgenmiş veri kümelerinin oluşturulması için kullanılır. Elde edilen veri seti ile yapılan veri madenciliğinin sonucu, verinin tamamından elde edilen sonuçtan çok farklı olmamalıdır.
    Veri indirgeme yöntemleri; 1) **Veri küpü birleştirme**, 2) **Boyut indirgeme**, 3) **Veri sıkıştırma**, 4) **Büyük sayıların indirgenmesi**.
  - **Veri Dönüştürme:** Bazı durumlarda orijinal veri kümelerindeki özellikler gerekli enformasyonu içerdiği halde veri madenciliği algoritmaları için uygun yapıda olmayabilirler. Bu durumda orijinal özelliklerinden dönüştürülerek oluşturulan yeni özellik, orijinal özelliklerden daha faydalı olabilir.
    Veri dönüşümü yöntemleri; 1) **Düzeltme**, 2) **Bir araya getirme**, 3) **Genelleme**, 4) **Normalleştirme**, 5) **Özellik oluşturma**.
- **Normalleştirme yöntemleri**
  - **En küçük – En büyük Normalleştirme:** Veri içindeki en büyük ve en küçük sayısal değerin belirlenerek diğer değerleri buna uygun bir şekilde dönüştürülmesiyle yapılır. Bu formül, uzaklık ve simetri değerlerinin normalleştirilmesinde de kullanılacaktır.

$$X^* = \frac{X - X_{enk}}{X_{enb} - X_{enk}}$$

X* dönüştürülmüş değeri, **X** gözlem değerini, **Xenk** verideki en küçük gözlem değerini, **Xenb** verideki en büyük değeri ifade eder.

  - **z-Skor Normalleştirme:** Uygulamada en çok kullanılan dönüştürme yöntemidir. Bir değişkene ilişkin aritmetik ortalama ve standart sapma hesaplamasından sonra elde edilir. z-Skor normalleştirme sonucunda **veri sıfır ile bir arasında sayısal bir değere dönüşür**.

$$X^* = \frac{X - \bar{X}}{s}$$

X* dönüştürülmüş değeri, **X** gözlem değerini, $\bar{X}$ değişkenin aritmetik ortalamasını, **s** değişkenin standart sapmasını ifade eder.

  - **Ondalık ölçekleme:** Değişkene ilişkin, gözlem değerlerinin ondalık bölümü hareket ettirilerek normalleştirme gerçekleştirilir.

$$X^* = \frac{X}{10^J}, \qquad J = \text{enb}\left(\left|X^*\right|\right) < 1$$

X* dönüştürülmüş değeri, **X** gözlem değerini ifade eder.

- **Yakınlık ölçüm değerleri** her zaman sonlu aralıkta olmayabilir. Örnek olarak [0,∞) aralığında değerler alan bir uzaklık ölçümü için, aşağıdaki eşitlik yardımıyla ölçüm değerleri [0,1] sonlu aralığına dönüştürülmüş olur.

$$d' = \frac{d}{1+d}$$

- **İki nesne arasındaki uzaklık**, iki nesnenin birbirinden farklılık derecesinin sayısal bir ölçüsüdür. İki nesne arasındaki düzensizliğin veya bozukluğun bir ölçüsü olan uzaklık, farklılığın özel bir sınıfı, alt kümesidir. **İki nesne arasındaki yüksek benzerlik değeri,** nesnelerin benzer olduklarını, **yüksek uzaklık değeri ise** nesnelerin benzer olmadıklarını ifade eder.
- Benzerlik değerlerinin **[0,1] sonlu aralığında olduğu durumda**, ilgili uzaklık değerleri **d = 1 – s** ile hesaplanabilir. Aynı şekilde [0,1] kapalı aralığındaki uzaklık değerlerine karşı gelen **benzerlik değerleri elde edilmek istendiğinde ise s = 1 – d** ile hesaplanabilir.
- Bir araştırmada elde edilen uzaklık değerleri **[0,∞) aralığında değerler alıyor iken** istenilen benzerlik değerlerini elde edebilmek için aşağıdaki eşitliklerden faydalanılır.

$$s=\frac{1}{1+d},\quad s=e^{-d}\quad veya\quad s=1-\frac{d-enk(d)}{enb(d)-enk(d)}$$

Nitelik Türü	Uzaklık	Benzerlik
Sınıflayıcı	$d(x,y)=\begin{cases}0, & x=y\ ise\\ 1, & x\neq y\ ise\end{cases}$	$s(x,y)=\begin{cases}1, & x=y\ ise\\ 0, & x\neq y\ ise\end{cases}$
Sıralayıcı	$d(x,y)=\frac{\lvert x-y\rvert}{(n-1)^*}$	$s(x,y)=1-d$
Aralıklı/ Oransal	$d(x,y)=\lvert x-y\rvert$	$s(x,y)=-d,\quad s(x,y)=\frac{1}{1+d},\quad s(x,y)=e^{-d},\quad s(x,y)=1-\frac{d-enk(d)}{enb(d)-enk(d)}$

- **Pisagor bağıntısına göre**, A ve B gibi **iki nokta arasındaki uzaklık** aşağıdaki gibi bulunur.

$$d(A,B)=\sqrt{(x_1,x_1)^2+(y_2,y_2)^2}$$

- Sınıflayıcı ve sıralayıcı ölçek ile ölçülebilen değişkenler **nitel**;
  Aralıklı ve oransal ölçek ile ölçülebilen değişkenler ise **nicel** değişkenler olarak adlandırılırlar.
- **Nicel değişkenler için yakınlık ölçüleri**
  Yakınlığın belirlenmesinde 1) **Öklid uzaklığı**, 2) **Karesel Öklid uzaklığı**, 3) **Karl Pearson uzaklığı**, 4) **Manhattan uzaklığı**, 5) **Minkowski uzaklığı**, 6) **Korelasyon uzaklığı**, 7) **Açısal benzerlik**, 8) **Mahalanobis uzaklığı** ölçülerinden yararlanılır.
  - **1) Öklid Uzaklığı:**

$$d_{ij}=\sqrt{\sum_{k=1}^{p}\left(x_{ik}-x_{jk}\right)^2}\qquad ;i=1,2,\ldots,n\quad ;j=1,2,\ldots,n\quad ;k=1,2,\ldots,p$$

n nesne sayısı, **p** değişken sayısı, $\mathbf{d_{ij}}$ i'nci ve j'inci nesneler arasındaki uzaklık,
$\mathbf{x_{ik}}$ i'inci nesnenin k'ıncı değişkendeki değeri, $\mathbf{x_{jk}}$ j'inci nesnenin k'ıncı değişkendeki değeri

  - **2) Karesel Öklid Uzaklığı:** Öklid uzaklığından tek farkı, değişkenlere göre toplam uzaklığın karekök alınmadan hesaplanmasıdır.
  - **3) Karl Pearson uzaklığı:** Öklid uzaklığının değişkenin varyansına oranlanması ile elde edilen bir uzaklıktır. Standartlaştırılmış Öklid Uzaklığı olarak da bilinir.

$$d_{ij}=\sqrt{\sum_{k=1}^{p}\left(\frac{x_{ik}-x_{jk}}{s_k^2}\right)^2}$$

  - **4) Manhattan (City-Block) Uzaklığı:** Birimler arası farkların mutlak değerinin toplamı alınmak suretiyle hesaplanır. Aynı zamanda L1 norm olarak da bilinen Manhattan uzaklığı bir başka uzaklık ölçüsü olan Minkowski uzaklığının özel bir hâlidir. Manhattan uzaklığı, değişkenler arasında ilişki olmaması durumunda hesaplanması gereken bir uzaklık ölçüsüdür. Ayrıca Manhattan uzaklığının aykırı değerlere karşı hassasiyeti düşüktür. Değişkenler arasında yüksek derecede ilişki olması durumunda veya değişkenlerin ölçü birimleri farklı olduğunda **kullanılmamalıdır**.

$$d_{ij}=\sum_{k=1}^{p}\left|x_{ik}-x_{jk}\right|$$

- **5) Minkowski Uzaklığı:** Değişkenler arasındaki uzaklıkları hesaplamak için kullanılan genel bir uzaklık ölçüsüdür. **L_λ norm olarak da bilinir**. λ değeri büyük ve küçük farklara verilen ağırlığı değiştirir.
  Örneğin, **λ = 1** iken Manhattan uzaklığı, **λ = 2** iken Öklid uzaklığı elde edilir.

$$d_{ij} = \left[\sum_{k=1}^{p} \left| x_{ik} - x_{jk} \right|^{\lambda} \right]^{1/\lambda} \quad , \quad \lambda \geq 1$$

- **6) Pearson Korelasyon Katsayısı ve Korelasyon Uzaklığı:** Değişkenler arasındaki doğrusal ilişkinin yönünün ve derecesinin belirlenmesinde kullanılan bir katsayıdır ve r sembolü ile gösterilir. İki değişkenli bir veri için hazır olarak verilen **r_{xy} korelasyon katsayısı** kullanılarak, korelasyon uzaklığı aşağıdaki gibi bulunur.

$$d_{xy} = \frac{1 - r_{xy}}{2}$$

Pearson korelasyon katsayısı [-1,+1] arasında değerler alsa da **korelasyon uzaklığının değerleri** [0,1] aralığında değerler alır.

- **7) Açısal Benzerlik:** İki vektör arasındaki açı farkının kosinüsünün, bu iki vektör arasındaki uzaklık olarak alınması suretiyle, değişkenler arasındaki benzerliğin belirlenmesine yönelik bir benzerlik ölçüsüdür. Elde edilen değer 1 ise, tam bir benzerlik var demektir. 0 olması durumunda değişkenlerin hiç benzerliğin olmadığı anlaşılır. Özellikle belge ve çoklu ortam nesnelerinin kıyaslanmasında ve metin madenciliğinde kullanılmaktadır. Veri madenciliğinde özellikle kümeleme analizinde tespit edilen küme içi uyumu ölçmek için kullanılır. **Seyrek (sparse) matrisler**in olduğu veri madenciliği problemlerinde de etkin olarak kullanılmaktadır.

$$s_{xy} = cos\theta = \frac{x \cdot y}{\|x\| \|y\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$$

- **8) Mahalanobis Uzaklığı:** Sürekli değişkenler arasındaki yakınlığın belirlenmesinde kullanılır. İki vektör veya değişken arasındaki uzaklığın belirlenmesinde verilerin kovaryans yapılarını da dikkate almaktadır.

$$d_{xy} = D^2 = (x - y)^T S^{-1} (x - y)$$

S, **n × n** boyutlu örneklem ya da küme içi kovaryans matrisidir.

- **dist()** fonksiyonu yardımıyla, verilerin matrisi olarak girilen **x**'e ait **nesneler arasındaki belirli uzaklık ölçüm değerleri** hesaplanabilir. Bu fonksiyon R'nin **stats** paketinde yer alır ve kod içerisinde **library(stats)** yazılarak, kullanıma hazır hale gelmesi sağlanır.
- **cor()** fonksiyonu yardımıyla her biri n adet gözlem değeri içeren x ve y değişkenleri sütun vektörleri arasındaki **Pearson korelasyon katsayısı** hesaplanır. Bu fonksiyon da, tıpkı dist() gibi, R'nin **stats** paketinde yer alır.
- **cosine()** fonksiyonu ile her biri n adet gözlem değeri içeren x ve y değişkenleri sütun vektörleri arasındaki açının kosinüsü, yani **açısal benzerliği** hesaplanır. Bu fonksiyon R'nin **lsa** paketinde yer alır ve kodda **library(lsa)** yazılarak, kullanıma hazır hale gelmesi sağlanır.
- **mahalanobis()** fonksiyonu, **Mahalanobis uzaklığı**nın R ile hesaplanabilmesi için **stats** paketi içerisinde yer alan bir fonksiyodur.

- **Binary değişkenler için yakınlık ölçüleri**
  İki sonuçlu (binary) değişkenler, ölçüm değerleri sınıflama yoluyla elde edilen **nitel değişkenler**dir. Bu değişkenler sadece evet/hayır, var/yok, erkek/kadın, doğru/yanlış gibi değerler alırlar. Hesaplama yapmadan önce **kontenjans** ya da diğer adıyla **çapraz sınıflama tablosu**nun oluşturulması gerekir.
  - **1) Basit Eşleştirme Katsayısı ve Uzaklığı:** p tane değişken açısından ilgilenilen nesnelerin her ikisinde de olmama (0-0) ve olma (1-1) durum sayılarının oranını gösteren bir benzerlik ölçüsüdür. Diğer bir deyişle, tesadüfi olarak seçilen bir değişkende her iki nesnenin de aynı özelliğe sahip olma olasılığını veren bir katsayıdır.
  - **2) Binary Öklid ve Binary Karesel Öklid Uzaklığı:** iki sonuçlu değişkenler arasındaki yakınlığın belirlenmesinde yaygın olarak kullanılan ve tutarlı bir ölçü olduğu kabul edilen bir uzaklık ölçüsüdür.
  - **3) Jaccard Benzerlik Katsayısı ve Uzaklığı:** Jaccard benzerlik katsayısı **özellikle ekolojik araştırmalarda belirli bir nesnenin farklı bölgelerde var olup olmadığının belirlenmesinde** kullanılmaktadır.

- **smc()** fonksiyonu, R yazılımında **Basit Eşleştirme Katsayısı**nın hesaplanması için kullanılan, **scrime** paketinde yer alan bir fonksiyondur.
- **vegdist()** fonksiyonu, R yazılımında **Binary Öklid uzaklığı**nın ve **Jaccard uzaklığı**nın hesaplanması için kullanılan ve **vegan** paketinde yer alan bir fonksiyondur. Jaccard hesaplanırken, method parametresinin method = "jaccard" şeklinde olması gerekir.

- **İlişki Kuralları:** Büyük veri kümeleri içerisinde belirli veriler arasındaki ilişkileri bulan ve **olayların birlikte gerçekleşme ihtimallerini** geçmiş verileri analiz edip ortaya koyarak, **geleceğe yönelik çalışmaları destekleyen** veri madenciliği yöntemine denir.
- **Pazar sepeti analizi (İlişki analizi),** müşterilerin **alışveriş alışkanlıklarının veritabanındaki bilgiler aracılığı ile ortaya çıkartılması** işlemi.
- **İlişki analizinin amacı**, birçok kural içerisinden belirlenen **destek ve güven eşik değerlerini sağlayan** kuralların seçilmesidir.

Tid	Nesneler
1	Süt, Ekmek
2	Ekmek, Yumurta
3	Ekmek, Şeker
4	Süt, Ekmek, Yumurta
5	Ekmek, Yumurta

- **m** nesne içeren bir **I** nesneler kümesinden elemanları birbirinden farklı oluşturulması mümkün tüm nesne setlerinin sayısı 2^m tanedir. Ancak bu nesne setlerinden bir tanesi boş kümedir. Boş küme ilişki kuralı belirlemek için kullanılamayacağından, ilişki kuralı belirlemede kullanılacak **nesne seti sayısı** $2^m - 1$ tane olur.
- **m** adet nesne içeren bir **I** nesneler kümesinden ilişki kuralı oluşturmada kullanılabilecek **k** ($1 \le k \le m$) tane nesne içeren nesne kümelerin sayısı C_k^m (m'nin k'lı kombinasyonu) kadardır.
- **m** adet nesne içeren bir **I** nesneler kümesinden toplam $3^m - 2^{m+1} + 1$ adet **ilişki kuralı** oluşturulabilir.
- Bir ilişki kuralının gücü, o kural için hesaplanacak **destek ve güven** değerleri ile ölçümlenebilir.
- **İlginç Kural:** Bir alışveriş veritabanından oluşturulacak **ilişki kuralları arasından** işe yarayacak bilgiyi üretmek amacıyla kullanılacak ilişki kuralı **ilginç kural** olarak tanımlanabilir. İlginç kuralların belirlenebilmesi için kullanılan objektif ölçütler **destek ve güven** ölçütleridir.
- Bir A nesne setinin **destek değeri**, aslında Destek(A) = P(A)'dır. Yani A nesne setinin gözlenme olasılığıdır.

$$Destek(A) = Destek\{Süt, Ekmek\} = \frac{|Süt, Ekmek|}{|D|} = \frac{2}{5} = 0,40\ (\%40)$$

Üstteki tabloda yer alan toplam 5 işlemin 2 tanesinde (1. ve 4. işlemlerde) süt ve ekmek nesneleri birlikte alınmış olduğundan, elde edilen bu destek değeri alışverişlerin %40'ında süt ve ekmeğin birlikte alınmış olduğunu ifade eder.

- **Bir ilişki kuralının destek değeri**, o kuralın öncül (A) ve sonuç (B) kısmındaki nesne setlerinin birlikte gözlenme olasılığıdır ve P(A∪B) şeklinde ifade edilir. **A⇒B şeklindeki bir ilişki kuralının destek değeri**, aslında P(A∪B)'dir. Yani A ve B nesne setlerinin birlikte gözlenme olasılığıdır. "Süt + Ekmek" bir Nesne seti, "Süt + Ekmek + Yumurta" ise İlişki kuralıdır.

$$Destek(A) = \frac{|A \cup B|}{|D|}$$

$$Destek(\{Süt, Ekmek\} \Rightarrow \{Yumurta\}) = \frac{|\{Süt, Ekmek, Yumurta\}|}{|D|} = \frac{1}{5} = 0,20\ (\%20)$$

- **İlginç kural elde etmede ilk eleme işlemi**, önceden belli olan bir destek eşik değerine göre yapılır. Belirlenen destek eşik değerine eşit veya daha büyük destek değerine sahip nesne setine **sık görülen nesne seti** denir. **Destek eşik değeri** ne çok fazla ne de çok az olmalıdır. Orta bir değerde seçilmelidir. Aksi halde çok az veya çok fazla ilginç kural setleri oluşur. Bu istenmeyen bir durumdur.
- **İlginç ilişki kuralı elde edebilmek için kullanılan ikinci ölçüt**, güven değeridir. Karar verici tarafından belirlenmiş olan güven eşik değerine eşit ya da daha büyük güven değerine sahip ilişki kuralları ilginç kural elde etmek için değerlendirilmeye alınırken, bu değerin altında güven değerine sahip ilişki kuralları ise elenir, değerlendirilmez.
- **A⇒B şeklindeki bir ilişki kuralının güven değeri**, aslında A'yı içeren işlemlerin aynı zamanda B'yi de içerme olasılığıdır yani P(B|A) koşullu olasılığıdır. Yani A bilindiğinde B'nin ortaya çıkma olasılığıdır.

$$Güven(A \Rightarrow B) = \frac{Destek(A \cup B)}{Destek(A)} = \frac{|A \cup B|}{|A|}$$

- Belirlenen destek ve güven **eşik değerleri üzerinde** destek ve güven değerine sahip ilişki kuralına **güçlü kural** denir. Yani **hem destek eşik değeri, hem de güven eşik değerinin üzerine çıkılmış** olan nesne setleridir.
- **Güven eşik değerinin eleme gücü yüksek olmalı**, yani olabildiğince büyük seçilmelidir.
- Bazı durumlarda, destek ve güven eşik değerleri kullanılmasına rağmen, değerlendirilmesi gereken güçlü kural sayısı yine de fazla olabilmektedir. Böyle durumlarda, öncül (A) ve sonuç (B) nesne setleri arasındaki ilişkinin (korelasyonun) belirlenmesi temeline dayanarak hesaplanan **kaldıraç (lift)** değeri kullanılır.

$$Kaldıraç(A \Rightarrow B) = \frac{Güven(A \Rightarrow B)}{Destek(B)} = \frac{Destek(A \cup B)}{Destek(A).Destek(B)}$$

  - Kaldıraç (A⇒B) **< 1** olması, A ve B nesne setleri arasında ters yönlü (negatif) bir ilişki olduğunu
  - Kaldıraç (A⇒B) **= 1** olması, A ve B nesne setleri arasında ilişki olmadığını.
  - Kaldıraç (A⇒B) **> 1** olması, A ve B nesne setleri arasında aynı yönlü (pozitif) bir ilişki olduğunu ifade eder.

$$Kaldıraç\left(\left\{Havuç\right\}\Rightarrow\left\{Turp\right\}\right)=\frac{Güven\left(\left\{Havuç\right\}\Rightarrow\left\{Turp\right\}\right)}{Destek\left\{Turp\right\}}=\frac{0,85}{0,50}=1,70$$

Yorumu: Havuç alındığında turbun da alınma olasılığı, sadece turbun alınma olasılığından %70 daha fazladır.

- **Apriori Algoritması**: **İlişki kuralı oluşturabilmek için geliştirilen** algoritmalar içerisinde en çok bilinen ve en sık kullanılan algoritmadır. Apriori algoritması **1994** yılında **AgraXal** ve **Srikant** tarafından geliştirilmiştir.
- **Apriori Özelliği:** "Eğer k nesneden oluşan nesne setleri kümesi en küçük destek kriterini sağlıyorsa, bu kümenin alt kümeleri de en küçük destek kriterini sağlar." **I**={a,b,c,d} nesne kümesi için, şayet {a,b,c} nesne kümesi bir sık görülen nesne kümesi ise, onun tüm alt kümeleri olan Ø, {a}, {b}, {c}, {a, b}, {a, c} ve {b, c} kümeleri de sık görülen nesne kümeleridir. Bu özelliğe **apriori özelliği** adı verilir.
- **Destek Bazlı Budama Özelliği:** Apriori özelliğinin aksine, "Eğer bir alt küme sık görülen nesne kümesi değil ise, onun bütün üst kümeleri de sık görülen nesne kümesi değildir" temel yaklaşımına sahiptir. **I**={a,b,c,d} nesne kümesi için, şayet {c, d} nesne kümesi bir sık görülen nesne kümesi değil ise, **bu kümenin elemanlarını içeren tüm üst kümeleri olan** {a, c, d}, {b, c, d} ve {a, b, c, d} kümeleri de sık görülen nesne kümeleri **değildir**.
- **apriori()** fonksiyonu R yazılımında, **arules** paketi içerisinde yer alır ve bu fonksiyon ile **güçlü ilişki kuralları oluşturulur**.
- **Karar verme**, karar vericinin karşılaştığı bir problem çözümünde olumlu bir sonuca ulaşabilmek için, problemin sunduğu birden fazla olası seçenek içerisinden **seçim yapması** işlemidir.
  Çok farklı sebeplere dayanabilir: 1) **İç güdüsel**, 2) **İhtiyaç duyulan gereksinimler**, 3) **Bireysel**, 4) **İşletme**.
- Karar probleminin zaman içerisinde doğuracağı sonuçlardan etkilenen **sorumlu kişi**ye **karar verici** adı verilir.
- **Karar ağaçları**, karar vericinin içinde bulunduğu karar verme probleminde ortaya çıkabilecek tüm durumları ve karar vericinin karşılaşabileceği **tüm senaryoları bir arada gösterebilen bir grafiksel yaklaşım**dır.
- **Karar ağaçlarının avantajları:**
  - Açıklanması ve yorumlanması kolaydır.
  - İnsani kararı diğer yaklaşımlara göre daha iyi yansıtır.
  - Grafiksel gösterim mümkündür.
  - **Nitel** değişkenlerle çalışabilmesi.
- **Sınıflandırma:** Bir kaydı, **önceden tanımlanmış çeşitli sınıflardan birine** atayan bir modelin uygulanması işlemi olarak tanımlanabilir.
- **Entropi:** Bir veri yığınındaki düzensizliğin, rassallığın miktarını ölçmek için kullanılan bir ölçüdür.
- **Kestirim**, bir rassal değişkenin seçtiğimiz modele göre **parametrelerinin yerine konulması** ile elde edilen değerdir.
- **Kök ve iç düğüm** bir karar ağacını başlatan ve büyüten düğümler, **yaprak düğüm** ise dallanmayı sonlandıran düğümdür.
- **Ayırma Kriteri:** Düğümün temsil ettiği ve **ayırma işlemini en iyi şekilde gerçekleştirecek olan nitelik**tir. Çeşitli ayırma kriterleri vardır.
  Nitel veri için; **Entropi İndeksi, Gini İndeksi, Sınıflandırma Hatası İndeksi ve Twoing ölçüleri** kullanılır.
  Nicel veriler için ise **En Küçük Kareler Sapması** yöntemi en sık kullanılan ölçüdür.
- **Gini İndeksi:** İkili bölünmeye dayanan bir tekniktir ve nitelik değerlerinin sola ve sağa olmak üzere iki bölüme ayrılmasına dayanır.
- **Karar Ağacı Oluşturma Algoritmaları:** ID3, C4.5, **CART**, CHAID, **QUEST**, **SLIQ**, **SPRINT** ve **MARS**.
  - **ID3 algoritması**, en basit karar ağacı oluşturma algoritmasıdır. Ayırma kriteri olarak kazanç ölçütünden yararlanılmaktadır.
  - **C4.5 algoritması**, ID3 algoritmasının geliştirilmiş hâlidir. Ayırma kriteri olarak **kazanç oranı**ndan yararlanılmaktadır.
  - **CART**, **ikili (binary) karar ağacı** yapısından dolayı diğer algoritmalardan farklılık gösterir. Ayırma kriteri için **Entropi, Gini ve Twoing** indekslerinden karar ağacını budamak için ise **maliyet karmaşıklığı** kriterinden faydalanılır. CART algoritmasının önemli bir işlevi ise yaprak düğümlerinde bir **sınıf kestirimi yerine, sayısal bir değer kestirimi**ni içeren **regresyon ağacı**nın da oluşturulabilmesidir.
  - **CHAID algoritması**, genellikle sayısal olmayan (ölçüm düzeyi sınırlayıcı) nitelikleri işleyebilecek şekilde geliştirilmiştir.
  - **QUEST algoritması,** tek değişkenli ve doğrusal kombinasyon ayırmaları destekler.
- **Budama**, **bir ya da daha fazla dalı çıkartarak, karar ağacını daha basitleştirmek** amacıyla, yaprak düğüm ile değiştirme işlemidir. Bu işlem, çıkartılmasına karar verilen dalın içerdiği kayıtların, bağlı olduğu üst düğüme dahil edilerek, düğümün yaprak düğüme dönüştürülmesine dayanır. Kestirim hata oranının, ortaya çıkan **aşırı uyum (overfitting) sorununun giderilmesi**, azaltılması ve **sınıflandırma modelinin kalitesinin arttırılması** hedeflenir.
- **Karar ağacının doğruluğunun ölçülmesi için kullanılan teknikler**:
  1) **hold-out**, 2) **tekrarlı hold-out (repeated hold-out)**, 3) **çapraz-doğrulama** (cross-validation), 4) **bootstrap**.
- **Sınıflandırma ve regresyon ağaçları (CART)** veri madenciliğindeki sınıflandırma problemlerinde sık kullanılan bir yöntemdir. **İkili (binary) karar ağaçları oluşturulduğu için** diğer algoritmalardan ayrılır. Karar ağacındaki her bir düğüm sadece iki dala ayırır. Ayırma kriteri için **Entropi, Gini ve Twoing** indekslerinden, karar ağacını budamak için ise **maliyet karmaşıklığı** kriterinden yararlanmaktadır.
- **CART'ın R yazılımında uygulanabilmesi için rpart** paketinden yararlanılır. rpart paketi içerisinde yer alan **rpart()** fonksiyonunda kullanılan parametreler sırasıyla, hedef niteliği de içeren herhangi bir etkileşimin söz konusu olmadığı ilişki formülünü ifade eden ***formula***, formüldeki değişkenlerin çevrilebilmesi için gerekli olan veri yığınını içeren değişkeni ifade eden ***data*** ve karar ağacının oluşturulma amacını ifade eden ***method*** parametreleridir. Sonuçlara göre, sırasıyla düğüm numarası **(node)**, düğümü yaratan ayırıcı niteliğin tanımı **(split)**, düğümdeki kayıt sayısı **(n)**, düğümdeki kayıp kayıt sayısı **(loss)**, düğüm için yapılan sınıf kestirimi **(yval)** ve ilgili düğümde yer alan kayıtların sınıflayıcı nitelik değerlerinin olasılıkları **(yprob)** yer almaktadır. "*" ile işaretlenen düğümler yaprak düğümleri ifade etmektedir. Karar ağacının çizimi için **rpart.plot** paketinde yer alan **prp()** fonksiyonu kullanılabilir. Grafiksel gösterimi için kullanılabilecek diğer bir fonksiyon ise **rattle** paketi içinde yer alan **fancyRpartPlot()** fonksiyonudur.

- **Kümeleme (Clustering)**, veri setinde bulunan gözlemlerin ya da değişkenlerin **kendi aralarındaki benzerlikleri göz önünde bulundurularak gruplandırılması** işlemidir. Kümeleme yöntemlerinin çoğu **veri arasındaki uzaklıkları kullanır**.
- **Kümeleme analizinin amacı**, gruplanmamış verileri benzerliklerine göre **sınıflandırmak ve özetleyici bilgiler elde etmede yardımcı** olmaktır. Kümeleme **tanımlayıcı bir yöntemdir.**
- **Birliktelik Kuralları**, veri seti içerisinde yer alan kayıtların birbiriyle olan ilişkilerini inceleyerek, **hangi olayların eş zamanlı olarak birlikte gerçekleşebileceklerini ortaya koymaya çalışan** yöntemler veri madenciliği yöntemleridir. **(Pazar sepet analizleri)**
- **Kümeleme analizi,** diğer çok değişkenli analiz yöntemi olan **diskriminant analizinde olduğu gibi tahmin amaçlı kullanılmamakta** ve faktör analizinde olduğu gibi de varsayımları bulunmamaktadır.
- **Kümeleme analizi dört aşamadan oluşur:** 1) **veri matrisinin oluşturulması**, 2) **benzerlik veya uzaklık matrislerinin hesaplanması**, 3) **kümelemede esas alınacak yöntemlerin belirlenmesi** ve 4) **elde edilen sonuçların yorumlanmasıdır**.
- **Kümeleme yöntemleri;** 1) Aşamalı Kümeleme, 2) Aşamalı Olmayan Kümeleme.
- **Aşamalı Kümeleme Yöntemleri;** 1) Birleştirici, 2) Ayırıcı.
- **Birleştirici (Agglomerative) Aşamalı Kümeleme Yöntemleri:** Bu yöntemde, her birim başlangıçta tek başına farklı birer küme olarak kabul edilir. Daha sonra birbirleri ile yüksek derecede benzerlik gösteren iki birim, bir küme oluşturur. Bir sonraki adımda bu kümeye farklı benzerlik düzeylerinde diğer birimler eklenerek birimlerin tamamı bir kümede toplanacak biçimde birbirleri ile bağlanırlar (birleştirilirler, kümelenirler).
  - **Tek Bağlantı Kümeleme Yöntemi (TekBKY, SINGLE Linkage [SLINK], En Yakın Komşuluk, Nearest Neighbour Method):** Küme elemanları arasındaki en küçük uzaklık değeri temel alınarak kümelerin oluşturulması esasına dayanır. Bu yöntemde, m ve j kümeleri arasındaki uzaklık; $d_{mj} = \min (d_{kj}, d_{lj})$ formülüyle hesaplanır.
  - **Tam Bağlantı Kümeleme Yöntemi (TamBKY, COMPLETE linkage Method [CLINK], Furthest Neighbor Method):** Üstteki yöntemle tek farklılık oluşturulan her kümedeki eleman çiftleri arasındaki uzaklığın maksimum olanının ele alınmasıdır. Tam bağlantı tekniğindeki uzaklıklar, $d_{mj} = \text{maks} (d_{kj}, d_{lj})$ formülüyle hesaplanır.
  - **Ortalama Bağlantı Kümeleme Yöntemi (OrtBKY, AVERAGE Linkage Method, [ALINK]):** Kriter olarak, bir küme içindeki birim ile diğer küme içindeki birimler arasındaki ortalama uzaklıklar dikkate alınır. Uzaklıklar $d_{mj} = (N_k d_{kj} + N_l d_{lj}) / N_m$ formülüyle hesaplanır.
  - **McQuitty Bağlantı Kümeleme Yöntemi (McQuitty linkage Method):** m. kümenin oluşumunda k. ve l. kümelerin j. küme ile olan uzaklıkları toplamının yarısı (ortalaması) hesaplanır. $d_{mj} = (d_{kj} + d_{lj}) / 2$
  - **Küresel Ortalama Bağlantı Kümeleme Yöntemi (KOBKY, CENTROID linkage Method)**
  - **Medyan Bağlantı Kümeleme Yöntemi (MBKY, MEDIAN linkage Method)**
  - **Ward Bağlantı Kümeleme Yöntemi (WBKY, WARD linkage Method, En Küçük Varyans Kümeleme Yöntemi)**
- **Ayırıcı (Divisive) Aşamalı Kümeleme Yöntemleri:** Başlangıçta veri setinde bulunan tüm birimlerin bir küme olduğu varsayılarak analize başlanır. Diğer bir ifadeyle işlem, birleştirici aşamalı kümeleme yönteminde olan aşamaların tam tersine işler. Tüm terimleri içeren büyük bir küme ele alınır. İzleyen aşamalarda en farklı (uzak) birimler birbirinden ayrılarak daha küçük kümeler oluşturulur.
- **Dendrogramlar (Ağaç Diyagramları):** Kümeleme analizinde sonuçlar dendrogram (ağaç diyagramı) adı verilen grafiksel yöntemle sunulurlar. Genellikle dendrogramlar; x ekseninde birimler ve y ekseninde de uzaklıklar olacak şekilde yapılandırılırlar. R yazılımında; > **plot**(h,labels=x$Ülke) şeklinde uygulanır.
- **Aşamalı Olmayan Kümeleme Yöntemleri:** Bu yöntemlerde küme sayısı önceden belirlenir. Aşamalı kümeleme yöntemleri daha çok küçük veri setleri için uygundur. Buna karşılık aşamalı olmayan kümeleme yöntemleri ise daha çok büyük veri setlerine uygulanmaktadır. Ayrıca aşamalı olmayan kümeleme yöntemleri veri setinde bulunan aşırı uç değerlerden daha az etkilenmektedir.
  - **k-Ortalamalar (k-Means) Yöntemi:** Bu yöntem, değişkenlerin **ortalama vektörlerini küme merkezi olarak ele alır** ve kümeleme süreci bunun etrafında şekillenir. Bu kümeleme yöntemi, veri setinde bulunan birimleri küme içi kareler toplamlarını **minimize** (en küçük) edecek biçimde k sayıda kümeye ayırmayı amaçlar. **Karışık yapıda ya da kesikli değişken içeren veri setleri için uygun bir seçim değildir**. R yazılımında kısaca; > **results = kmeans(ulkeler.variable, 3)** şeklinde kullanılır.
  - **k-Medyanlar Yöntemi:** Medyan değerlerine ait vektörleri küme merkezi olarak kullanan k merkezli algoritmadır. Veri setindeki değişkenlerin asimetrik olduğu durumlarda kullanılmaktadır. Bu algoritmada, **Manhattan uzaklık ölçüsü en sık tercih edilen** kümelenme ölçütüdür.
  - **k-Medoidler Yöntemi:** Özellikle değişkenlerin birbirinden bağımsız olmadığı ve değişkenler arasında korelasyon olduğu durumlarda k-medyanlar yöntemi veri setini gruplamada (kümelemede) başarılı olmamaktadır. Bu durumda kümeleme için k-medoidler yöntemi önerilmektedir.
- **Veritabanı**, büyük miktardaki bilgileri depolamada yetersiz kalan dosya-işlem sistemine alternatif olarak geliştirilen ve birbirleriyle ilişkili bilgilerin depolandığı alandır.
- **Web madenciliği,** web dokümanlarından bilginin ayıklanması veya keşfedilmesini sağlayan bir veri madenciliği tekniğidir.
- **Veri ambarı**, veritabanı üzerindeki yükü hafifletmek için oluşturulmuş, birbiriyle ilişkili verileri kolay, hızlı ve doğru bir biçimde sorgulama ve analiz yapabilmek için gerekli işlemlerin yapılabildiği bir veri deposudur.
- **Veri madenciliği süreçleri:** 1) **Verinin elde edilmesi**, 2) **Verinin saklanması ve yönetimi**, 3) **Veri erişiminin sağlanması**, 4) **Verinin analiz edilmesi**, 5) **Analiz sonuçlarının anlaşılır bir biçimde sunulması**.
- **Web madenciliği süreçleri:** 1) **Kaynakların Tespiti**, 2) **Bilgi Seçimi ve Ön İşleme**, 3) **Genelleştirme**, 4) **Analiz**.
- **Sunucu (Server):** Yapılandırılan bir ağ üzerindeki diğer ağ bileşenlerinin(kullanıcıların) erişebileceği, kullanıma ve/veya **paylaşıma açık kaynakları barındıran**, güçlü donanım ve yazılım bileşenlerinden oluşan bilgisayar birimine denir.
- **İstemci (Client):** Bir ağ üzerinde sunucu bilgisayarlardan hizmet alan, erişim yetkileri sunucuda belirlenen **kullanıcı** bilgisayarlara denir.
- **Vekil (Proxy):** Bir ağ üzerinde sunucu ile istemci bilgisayarlar arasındaki **bilgi akışına aracı** - güvenlik duvarı, önbellekleme sistemi v.b. - olarak görev gören **ara sunucular**a vekil sunucu (proxy server) ya da kısaca vekil (proxy) denir.

- **Web madenciliğinde kullanılan veriler:** 1) İçerik verisi, 2) Yapı (harita) verisi, 3) Kullanım verisi, 4) Kullanıcı profil verisi.
- **Web Verisinin Özellikleri**
  - Web ortamındaki veri miktarı aşırı büyüklüktedir.
  - Web ortamındaki veri dağınık ve heterojen bir yapıdadır.
  - Web ortamındaki veri yapılandırılmamıştır.
  - Web ortamındaki veri dinamiktir.
- **Web Madenciliği üçe ayrılır:** 1) **İçerik Madenciliği**, 2) **Yapı Madenciliği**, 3) **Kullanım Madenciliği**.
- **Kısa metin işleme**, web sitelerinde var olan **metinsel verinin derlenmesi ve sınıflandırılması işlemi** olarak tanımlanabilir. Konuya göre dokümanların sınıflandırılmasında ve web sayfalarının alt kategorilere ayrılmasında kullanılan algoritmalar bütünüdür. Kısa metinlerin en bilindik uygulaması arama motorlarının kullanıcıya sunduğu aranılan kelimeyi tamamlayıcı nitelikte olan **"ilgili aramalar"** uygulamasıdır.
- **Web Görüş Madenciliği**, bir ürün veya hizmet hakkında yapılan **olumlu veya olumsuz görüşler analiz edilerek** kullanıcı eğilimleri tespit edilebilir, web sayfaları ona göre düzenlenebilir ve hatta sayfaya konulacak reklamların içerikleri düzenlenebilir.
- **Web Yapı Madenciliği**, web sitesinin yapısal özetini yani kendi içerisindeki sayfalarla ve diğer sitelerle olan bağlantı yapılarını elde ederek, bu yapılardan yararlı bilginin ortaya çıkarılması olarak tanımlanabilir. **(Yapıdan kasıt, bağlantı (link) yapısıdır.)**
- **Atıf analizi**, akademik olarak **yazarlar ile yayınları arasındaki ilişkiyi kurmak için** yapılan alıntıları inceleyen bir araştırma alanıdır. Bir yayın başka bir yayından alıntı yaptığında bu iki yayın arasında bir ilişki veya bağlantı kurulmuş olur. Dolayısıyla atıf analizinde de bu bağlantılar incelenerek **yayınların önem düzeyleri** ortaya konulmaya çalışılır.
- **Web topluluğu**, belirli bir konu üzerinde kaynak sağlayan web sayfaları topluluğudur.
- **Web kullanım madenciliği**, **kullanıcıdan elde edilen bilgiler aracılığı ile** kullanıcıların **internet gezinme alışkanlıklarını analiz ederek** kişiye özel modeller oluşturmayı amaçlar. Böylece ilgi alanları belirlenebilir, ilgi alanları ile ilgili öneriler sunulabilir. **(İkincil Veri Tipi)**
- **Web kullanım madenciliği üç aşamada gerçekleştirilir:** 1) **Veri ön işleme**, 2) **Örüntü keşfi**, 3) **Örüntü analizi**.
- **Web kullanım madenciliği, veri ön işleme aşamaları:**
  1) **Verinin Temizlenmesi**, 2) **Kullanıcı Bilgisinin Belirlenmesi**, 3) **Oturum Bilgisinin Belirlenmesi**, 4) **İz (Yol) Tamamlama**.
- **Web kullanım madenciliği, örüntü keşfi aşamasında;** istatistiksel analiz, ilişki kuralları, sınıflandırma analizi, kümeleme analizi ve sıralı örüntüler vb. gibi veri madenciliği teknikleri kullanılır.
- Veri madenciliği algoritmalarının sonucunda elde edilen çıktılara uygulanabilen herhangi araç veya filtreye **örüntü analiz aracı** denir. Örüntü analizi için yaygın olarak kullanılan iki araç vardır. İlki **SQL**, diğeri çevrimiçi analitik veri işlemeye imkân tanıyan **OLAP**'tır.
- **Web Kullanım Madenciliği Temel Uygulama Alanları:**
  1) Kişiselleştirme, 2) Sistem Geliştirme, 3) Web Sitesi Güncelleme, 4) İş Zekâsı, 5) Kullanım Karakteristiği.
- **Sosyal Medya'nın Sınıflandırılması:**
  1) Genel amaçlı veya arkadaş tabanlı, 2) Bilgilendirici, 3) Mesleki, 4) Eğitim, 5) Hobiler, 6) Akademik, 7) Haberler.
- **API (Application Programming Interface / Uygulama Programlama Arayüzü),** bir yazılımın başka bir yazılımda tanımlanmış fonksiyonlarını kullanabilmesi için uygulama oluşturmada kullanılan alt program, protokol ve araçlar bütünüdür.
- **Kontenjans Tablosu:**

	i. nesne			
j. nesne	Değişken	yok (-)	var (+)	Toplam
	yok (-)	a	b	a + b
	var (+)	c	d	c + d
	Toplam	a + c	b + d	p = a + b + c + d

Tablo 4.5
İki Sonuçlu İki Nesne İçin Kontenjans Tablosu

Kontenjans tablosunda; **a değeri:** i ve j nesnelerinin her ikisinde de ilgilenilen değişkenin olmadığı yani yok olduğu durum (0-0 eşleşmesi) sayısını, **b değeri:** ilgilenilen değişkenin i nesnesinde var olduğu ve j nesnesinde olmadığı durum (1-0 eşleşmesi) sayısını, **c değeri:** ilgilenilen değişkenin i nesnesinde olmadığı ve j nesnesinde var olduğu durum (0-1 eşleşmesi) sayısını, **d değeri:** i ve j nesnelerinin her ikisinde de ilgilenilen değişkenin var olduğu durum (1-1 eşleşmesi) sayısını, **p değeri:** değişken sayısını göstermektedir.

- **Basit Eşleştirme Katsayısı ve Uzaklığı:** Basit eşleştirme katsayısı, p tane değişken açısından ilgilenilen nesnelerin her ikisinde de olmama (0-0) ve olma (1-1) durum sayılarının oranını gösteren bir benzerlik ölçüsüdür.

$$s_{ij} = \frac{a+d}{a+b+c+d}$$

- Elde edilen bu benzerlik ölçüsünden yola çıkılarak **basit eşleştirme uzaklığı** aşağıdaki gibi bulunur.

$$d_{ij} = 1 - s_{ij} = \frac{b+c}{a+b+c+d}$$

- **Binary Öklid ve Binary Karesel Öklid Uzaklığı:** Binary Öklid uzaklığı, iki sonuçlu değişkenler arasındaki yakınlığın belirlenmesinde yaygın olarak kullanılan ve tutarlı bir ölçü olduğu kabul edilen bir uzaklık ölçüsüdür. Aşağıdaki gibi bulunur.

$$d_{ij} = \sqrt{b+c}$$

- **Binary Karesel Öklid uzaklığı** ise yandaki gibi bulunur: $d_{ij}^2 = b + c$