ISE 302 –Veri Madenciliği

DR. ÖĞR. ÜYESİ ESİN AYŞE ZAİMOĞLU

esinzaimoglu@sakarya.edu.tr

Problem

- Gerçek uygulamalarda toplanan veri kirli
- eksik: bazı nitelik değerleri bazı nesneler için girilmemiş, veri madenciliği uygulaması için gerekli bir nitelik kaydedilmemiş
  - meslek = “ ”
- gürültülü: hatalar var
  - maaş= “-10”
- tutarsız: nitelik değerleri veya nitelik isimleri uyumsuz
  - yaş= “35”, d.tarihi: “03/10/2004”
  - önceki oylama değerleri: “1,2,3”, yeni oylama değerleri: “A,B,C”
  - bir kaynakta nitelik değeri ‘ad’, diğerinde ‘isim’

Verinin Gürültülü Olma Nedenleri

- Eksik veri kayıtlarının nedenleri
  - Veri toplandığı sırada bir nitelik değerinin elde edilememesi, bilinmemesi
  - Veri toplandığı sırada bazı niteliklerin gerekliliğinin görülememesi
  - İnsan, yazılım ya da donanım problemleri
- Gürültülü (hatalı) veri kayıtlarının nedenleri
  - Hatalı veri toplama gereçleri
  - İnsan, yazılım ya da donanım problemleri
  - Veri iletimi sırasında problemler

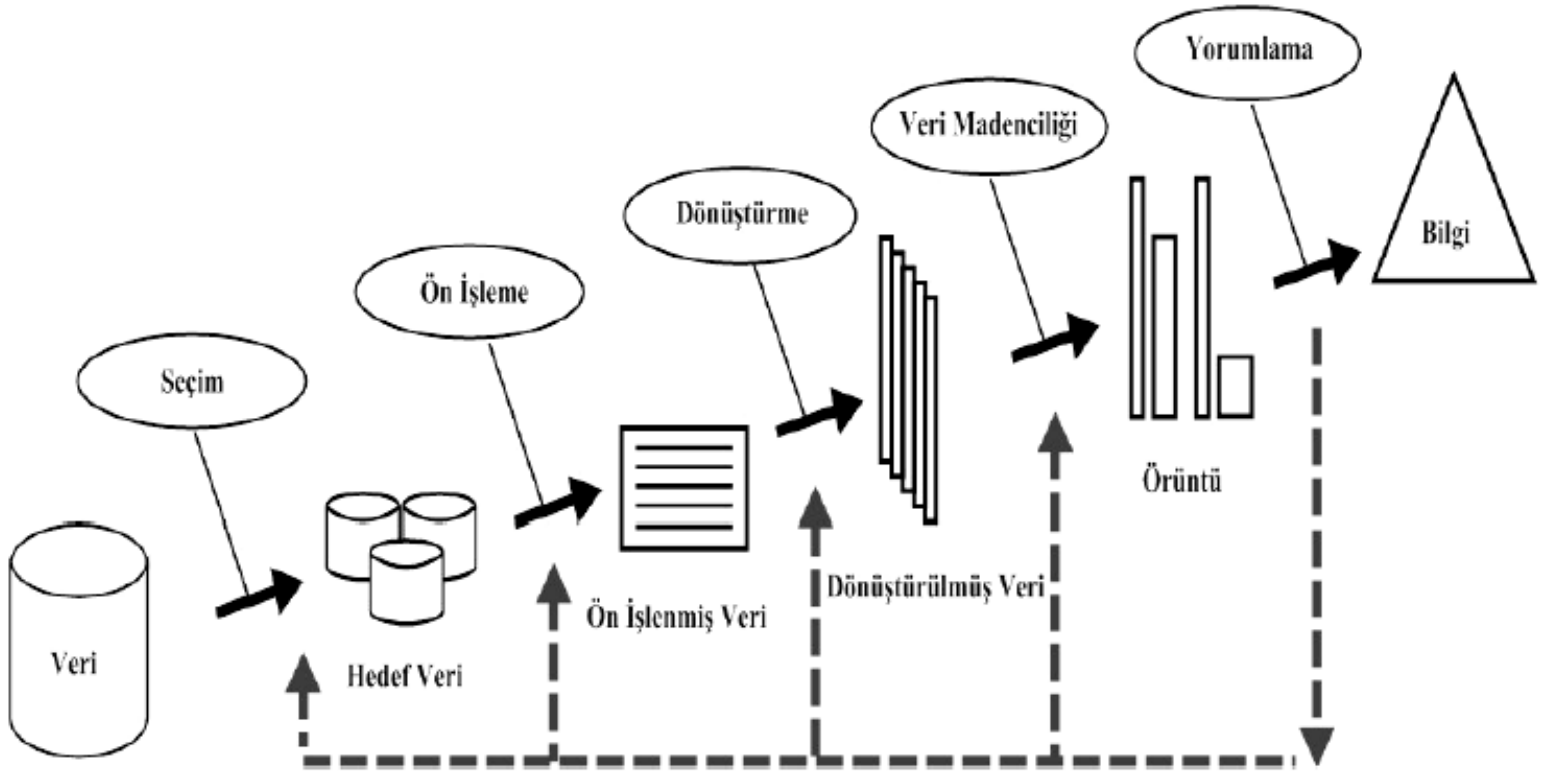
Verinin Gürültülü Olma Nedenleri

- Tutarsız veri kayıtlarının nedenleri
  - Verinin farklı veri kaynaklarında tutulması
  - İşlevsel bağımlılık kurallarına uyulmaması

Sonuç

- Veri güvenilmez
  - Veri madenciliği sonuçlarına güvenilebilir mi?
  - Kullanılabilir veri madenciliği sonuçları kaliteli veri ile elde edilebilir
- Veri kaliteli ise veri madenciliği uygulamaları ile yararlı bilgi bulma şansı daha fazla.

Veri Madenciliği Süreci

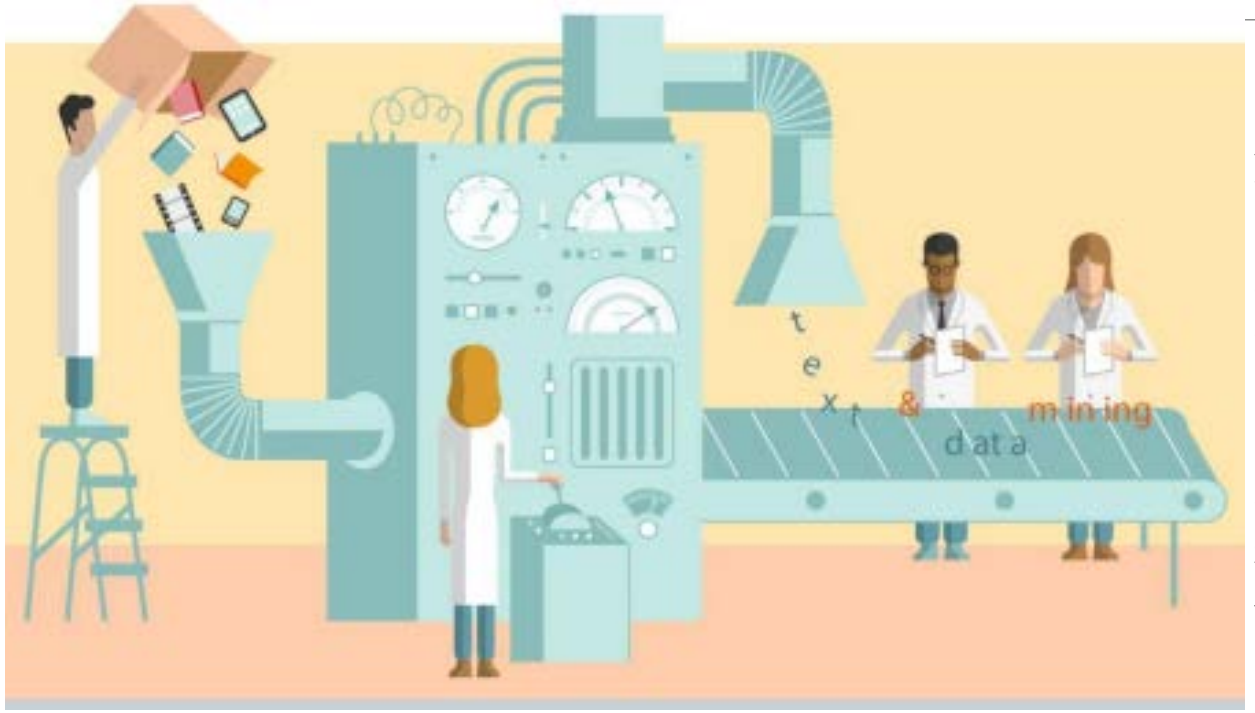


. Veri Madenciliği Süreci

1-Veri Seçimi(Data Selection)

- Veri seçimi, veri madenciliği aşamalarında en fazla zaman alan kısımlardan biridir.
- Bu aşamada bilgi sistemlerinde oluşan bilgi iyi analiz edilmelidir ve problemle ilişkilendirilmelidir.
- Büyük miktardaki verilerin tek bir veri tabanı veya veri ambarında birleştirilmesi veri madenciliği uygulaması için gereklidir.
- Bu adımdan sonra veri madenciliği süreci elde edilen ve hedef veri (target data) olarak isimlendirilen bu veri üzerinde gerçekleştirilir.

2- Veri Ön İşleme(Preprocessing)



Ön işleme aşaması veri madenciliğinin başarısı için önemlidir.

Bu aşamada veri, sonraki aşamalarda kullanılabilmesi için elverişli hale getirilir.

Başarılı bir ön işleme aşamasıyla kesin ve net sonuçlara ulaşmak mümkündür.

Veri Hazırlama/Temizleme ve Önişleme

- Veri analizi e modelleme uygulama aşamaları yanlış sonuçların önlenmesi adına her aşamada dikkat edilmesi gereken bir süreçtir.
- Veri ile çalışırken yapılan analiz sonuçlarının kaliteli ve güvenilir olması için verinin de kaliteli olması gereklidir.
- Veri yükleme, temizleme, dönüştürme gibi işlemler analiz sürecinin neredeyse %80'ini veya daha fazlasını almaktadır.
- Çoğunlukla veri elde edildiği ilk ham hali ile kullanım ve analiz için uygun değildir.
- Bu nedenle öncelikle veri hazırlanarak analiz için uygun hale getirilmelidir.
- Hazırlık işlemlerini kapsayan sürece de veri önişleme adı verilmektedir.
- Veri hazırlama aşaması, ham veriden başlayarak nihai veri dizisine erişinceye kadar gerekli tüm faaliyetleri kapsamaktadır.

Veri Hazırlama ve Önişleme Adımları

- Veri Temizleme (Eksik Verilerin Düzenlenmesi)
- Veri Birleştirme
- Veri Azaltma
- Veri Dönüştürme

→Veri Temizleme

Gerçek uygulamalarda veri eksik, gürültülü veya tutarsız olabilir.

- Veri temizleme işlemleri
- Eksik nitelik değerlerini tamamlama
- Aykırılıkların bulunması ve gürültülü verinin düzeltilmesi
- Tutarsızlıkların giderilmesi

Veri Temizleme (Eksik Verilerin Düzenlenmesi)

- Verinin eksik olması: Ör: Bir örneğin yaş değerinin girilmemiş olması
- Verinin gürültülü olması: Ör: Yaşın negatif ya da 500 gibi bir değer girilmesi
- Tutarsız veri: Ör: Doğum tarihi ile yaş verisi birbirine uygun değil
- Ya da kasıtlı olarak boş da bırakılmış olabilir.

Bu durumların çözülmesi gerekmektedir.

Veri Temizleme (Eksik Verilerin Düzenlenmesi)

Veri temizleme için ayrı ayrı yöntem ve teknikler geliştirilmiştir. Bunlar;

- İstatistiksel yöntemler
- Kümeleme
- Optimizasyon
- Regresyon

olabilir.

→Eksik Veri

Veri için bazı niteliklerin değerleri her zaman bilinemeyebilir.

Eksik veri

- diğer veri kayıtlarıyla tutarsızlığı nedeniyle silinmesi
- bazı nitelik değerleri hatalı olması dolayısıyla silinmesi
- yanlış anlama sonucu kaydedilmeme
- veri girişi sırasında bazı nitelikleri önemsiz görme

→Eksik Veriler nasıl Tamamlanır?

- Eksik nitelik değerleri olan veri kayıtlarını kullanma
- Eksik nitelik değerlerini elle doldur
- Eksik nitelik değerleri için global bir değişken kullan (Null, bilinmiyor,...)
- Eksik nitelik değerlerini o niteliğin ortalama değeri ile doldur
- Aynı sınıfa ait kayıtların nitelik değerlerinin ortalaması ile doldur
- Olasılığı en fazla olan nitelik değeriyle doldur

→Gürültülü Veri

- Grubbs'a göre sıra dışılık *"İçinde bulunduğu örnek kütlenin diğer üyelerinden önemli düzeyde farklılık gösteren gözlemdir."* (Grubb, 1974)
- Ölçülen bir değerdeki hata
- Yanlış nitelik değerleri
  - hatalı veri toplama gereçleri
  - veri girişi problemleri
  - veri iletimi problemleri
  - teknolojik kısıtlar
  - nitelik isimlerinde tutarsızlık

Gürültülü Veri nasıl Düzeltilir?

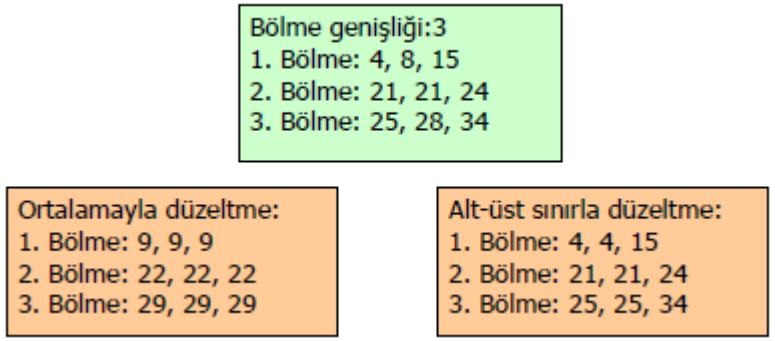
- Gürültüyü yok etme

- Bölmeleme
  - veri sıralanır, eşit genişlik veya eşit derinlik ile bölünür
- Demetleme
  - aykırılıkları belirler
- Eğri uydurma
  - veriyi bir fonksiyona uydurarak gürültüyü düzeltir

Bölmeleme

Veri sıralanır: 4, 8, 15, 21, 21, 24, 25, 28, 34

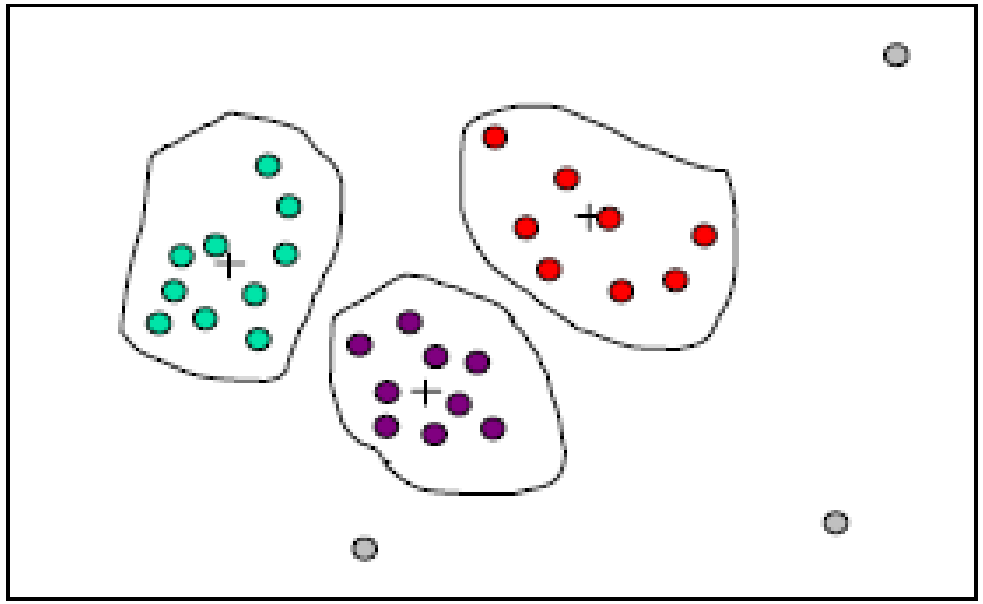
- Eşit genişlik: Bölme sayısı belirlenir. Eşit aralıklarla bölünür.
- Eşit derinlik: Her bölmede eşit sayıda örnek kalacak şekilde bölünür.
  - ✓her bölme ortalamayla ya da bölmenin en alt ve üst sınırlarıyla temsil edilir.



Demetleme

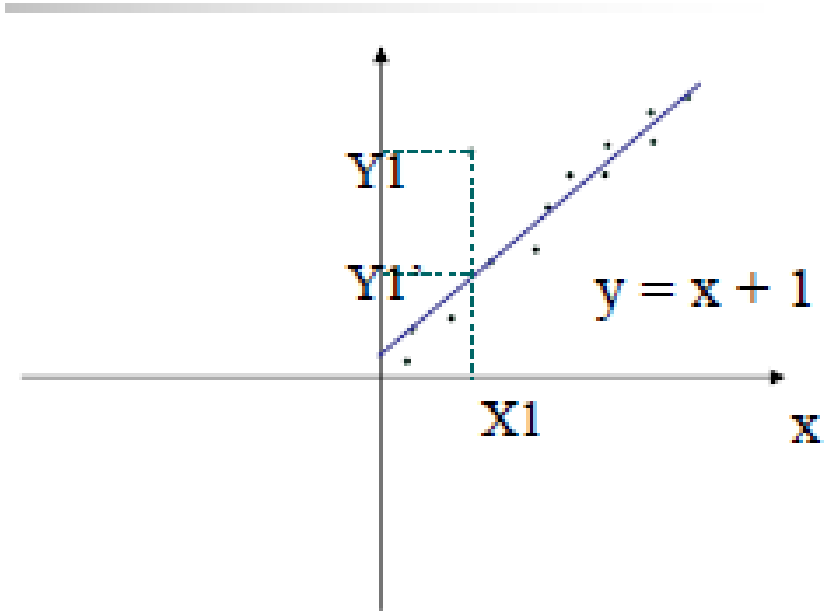
Benzer veriler aynı demette olacak şekilde gruplanır

- Bu demetlerin dışında kalan veriler aykırılık olarak belirlenir ve silinir.



Eğri Uydurma

- Veri bir fonksiyona uydurulur.
- Doğrusal eğri uydurmada, bir değişkenin değeri diğer bir değişken kullanılarak bulunabilir.



Veri Ön İşlemede Kontrol Edilecek Hususlar

- Kesinlik: Veride belirsizlik olup olmadığı, verilerin net şekilde belirtilip belirtilmediğidir
- Tamamlık: Tüm niteliklerde tüm değerlerin eksiksiz olarak girilip girilmediğidir.
- Tutarlılık: Çelişkili verilerin olup olmaması. Farklı veri kaynaklarından veriler birleştirildiğinde bu konuya dikkat edilmelidir.
- Güncellik: Veriler güncel olmalıdır.
- İnandırıcılık: Verilerde şüpheye neden olacak bir durum yani diğer verilerden ya da geçmiş verilerden çok farklı olmamalıdır.
- Yorumlanabilirlik: Veriler kolayca anlaşılabilir olmalıdır.
- Peki temel amacımız nedir? Min kayıp, max performans

Veri Birleştirme

- Bazı durumlarda farklı veri kaynaklarından ya da veri tabanlarından tek bir yerde toplanarak birleştirilebilir.
- Bu durumda veri kayıpları, yanlış eşleştirmeler olabilir. Aynı nitelik farklı kodlanmış ya da farklı metrikler kullanılmış olabilir
- Niteliklerdeki bu gibi tutarsızlıklar detaylı şekilde incelenerek düzeltilmelidir.
- Ayrıca farklı veri kaynaklarından verilerin birleştirilmesi gereksiz verilerin kaydedilmesine neden olabilir.

Veri Birleştirme

Sıcaklık	Baş Ağrısı	Mide Bulantısı	Nezle
100.2	NA	hayır	evet
100.6	evet	evet	evet
NA	hayır	hayır	hayır
99.6	evet	evet	evet

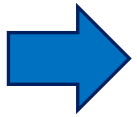
Sıcaklık	Baş Ağrısı	Mide Bulantısı	Nezle
99.8	NA	evet	hayır
96.4	evet	hayır	hayır
96.6	evet	evet	hayır
NA	evet	NA	evet

Sıcaklık	Baş Ağrısı	Mide Bulantısı	Nezle
100.2	NA	hayır	evet
100.6	evet	evet	evet
NA	hayır	hayır	hayır
99.6	evet	evet	evet
99.8	NA	evet	hayır
96.4	evet	hayır	hayır
96.6	evet	evet	hayır
NA	evet	NA	evet

Nitelik Birleştirme

id	Sıcaklık	Baş Ağrısı
1	100.2	NA
2	100.6	evet
3	NA	hayır
4	99.6	evet
5	99.8	NA
6	96.4	evet
7	96.6	evet
8	NA	evet

id	Mide Bulantısı	Nezle
1	hayır	evet
2	evet	evet
3	hayır	hayır
4	evet	evet
5	evet	hayır
6	hayır	hayır
7	evet	hayır
8	NA	evet



id	Sıcaklık	Baş Ağrısı	Mide Bulantısı	Nezle
1	100.2	NA	hayır	evet
2	100.6	evet	evet	evet
3	NA	hayır	hayır	hayır
4	99.6	evet	evet	evet
5	99.8	NA	evet	hayır
6	96.4	evet	hayır	hayır
7	96.6	evet	evet	hayır
8	NA	evet	NA	evet

Veri Azaltma

- Veri miktarı gereğinden fazla olduğunda analiz için kullanılan istatistiksel yöntemlerin ya da makine öğrenme algoritmalarının çözüm üretme süresi uzamaktadır.
- Gereksiz verileri veri setinde azaltarak temizlemek bazen başarı üzerinde de olumlu etkilere neden olmaktadır.
- Kısacası, veri setinin özet forma sokulduğu veri azaltma ve indirgeme işlemleri hem başarı hem de çalışma süresi açısından fayda sağlamaktadır. Sonucun (nerdeyse) hiç değişmemesi gerekir

Veri Azaltma

Veri azaltma

- nitelik birleştirme
- nitelik azaltma/seçme
- veri sıkıştırma
- veri ayrıştırma ve kavram oluşturma
- veri küçültme
  - eğri uydurma
  - demetleme
  - histogram
  - örnekleme

Veri Dönüşümü

- Veri setinde, analizde ya da sonrasındaki yorumlamalar aşamasında kullanım için anlamlı gözükmeyen nitelikler bulunabilir. Bu durum veri setini ham halinde de olabilir veri birleştirme gibi bir işlem sonucunda da ortaya çıkmış olabilir.
- Örneğin iki veri seti birleştirilirken bir veri setinde yaş bilgisi varken diğer veri setinde bu bilgi doğum yılı olarak girilmiş olabilir.
- Ya da bir nitelik veri tip açısından kullanılacak analiz yöntemine uygun olmayabilir.
- Bunun gibi durumlarda veriler üzerinde uygun dönüşüm işlemi yapılmalıdır.
- Eldeki verinin büyük bir kısmı, her ne kadar ön işleme aşamasından geçmiş olsa bile sonraki aşamalarda kullanılabilecek durumda değildir.

Veri Dönüşümü

- Veri entegrasyonu,
  - Verinin bir çok kaynaktan toplanması, seçilmesi ve entegre edilerek tek bir kaynakta bir araya getirilmesi işlemidir.
  - Günümüzde entegre sistemlerin kullanılması önem taşısa da, on yılların birikimi ile gelen çeşitli iç ve dış veri tabanlarının bir araya getirilmesi ve bunların özellikle *data warehouse (veri ambarı)* veya *data mart* gibi veri depolarında saklanması, veri ön işlemenin zaman ve işgücü açısından en yoğun işlemidir.
- Veri İndirgeme,
  - öz nitelik ve/veya nesne sayısının örnekleme *(sampling)*,
  - faktör analizi *(factor analysis)*,
  - boyut indirgeme *(dimension reduction)* gibi çeşitli tekniklerle azaltılması ile veri hacminin küçültülmesi için gerçekleştirilen işlemlerdir.

Veri Dönüşümü

Veri dönüştürme,

- normalleştirme *(normalization)*,
- kesimlere ayırma *(discretization)* /
- birleştirme *(aggregation)* gibi teknikler kullanılarak verinin modelleme aşamasında kullanılacak veri analizi modelleri / yöntemleri için hazırlanmasıdır.
- Başka bir deyişle daha sağlıklı sonuçların elde edilebilmesi veya verinin kullanılan algoritmalarla uyumlu olabilmesi için, verinin tanımlanan bir fonksiyona uygun olarak farklı değer veya ölçeklere dönüştürülmesi işlemidir.

Veri Dönüşümü

Veri Dönüşümü için kullanılabilecek çözümler;

- Veri düzeltme
  - Bölmeleme
  - Kümeleme
  - Eğri Uydurma

  Aykırı verilerin bulunması ve temizlenmesi
- Normalizasyon

  Verilerin 0 ve 1 arasında yeniden ölçeklenmesi
- Nitelik oluşturma

  Yeni nitelikler oluşturarak veri setine ekleme

  1. Oluşturulan yeni nitelik mevcut niteliklerden daha önemli bilgiler içermeli
  2. Analiz sonuçlarındaki başarıyı olumlu yönde etkilemeli

Normalizasyon

Veri normalleştirme / standardizasyon

- z-skor normalleştirmesi,
- [0,1] aralığında normalleştirme,
- [-1,1] aralığında normalleştirme,
- 10 tabanına göre logaritma,
- aritmetik ortalamanın 1 olduğu normalleştirme
- ve standart sapmanın 1 olduğu normalleştirme

olmak üzere altı ana başlık altında toplamak mümkündür.

Normalizasyon

- min-max normalizasyon

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- z-score normalizasyon

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

- ondalık normalizasyon

$$v' = \frac{v}{10^j}$$ j: Max($|v'|$)<1 olacak şekildeki en küçük tam sayı

Nitelik Oluşturma

Yeni nitelikler yarat

orjinal niteliklerden daha önemli bilgi içersin

- alan=boy x en

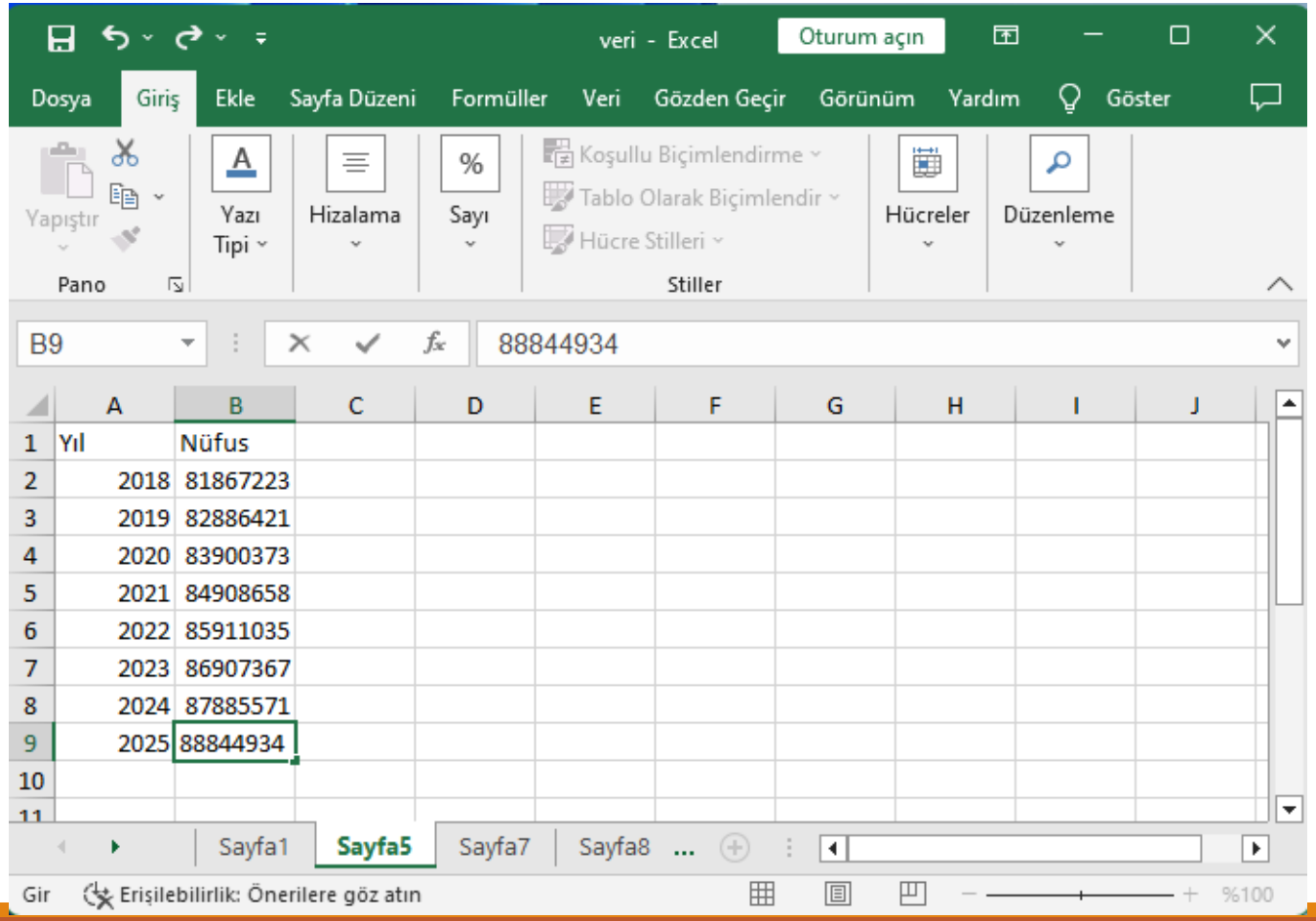
veri madenciliği algoritmalarının başarımı daha iyi olsun

4-Veri Madenciliği

- Veri madenciliği çalışmasının tam olarak kullanıldığı aşamadır.
- Veri bu aşamaya gelince doğru ve kullanılabilir haldedir.
- Çalışmanın amacına göre bu aşamada veri madenciliği yöntemlerinden biri veya birkaçı uygulanır.

‘xls’ veya ‘xlsx’ Uzantılı Dosya Girişi

veri.xlsx



‘xls’ veya ‘xlsx’ Uzantılı Dosya Girişi

Dosyanın Python ile okutulmasında aşağıdaki kod kullanılır:

```
import pandas as pd

veri=pd.read_excel("C:/Users/Hp/Desktop/veri.xlsx","Sayfa5")

veri.head()
```

Çıktı:

```
    Yıl     Nüfus
0  2018  81867223
1  2019  82886421
2  2020  83900373
3  2021  84908658
4  2022  85911035
```

Verinin Dizi Olarak Okutulması

Numpy kütüphanesini kullanarak verinin bir dizi olarak okutulmasını sağlayabiliriz:

```
import numpy as np

veri=np.loadtxt('C:/Users/Hp/Desktop/Veri2.txt', delimiter=',')

veri
```

Çıktı:

```
array([[1. , 1.4, 1.8],
       [2. , 2.8, 0.9],
       [3. , 4.5, 2.3],
       [4. , 4.2, 0.5],
       [5. , 5.3, 0.6]])
```

delimiter=‘,’ olarak tanımlanan komut asıl txt dosyasında verilerin arasında bulunan ayırma işaretini belirtir. Bu komut eklenmezse dosya yanlış okunabilir veya hata verebilir.

Verilerin Temizlenmesi ve Hazırlanması

Bu bölümde;

- eksik veriler,
- yinelenen veriler,
- diğer bazı analitik veri dönüşümleri

için araçlar ve uygulamaları gösterilecektir.

Eksik Verilerin Düzenlenmesi

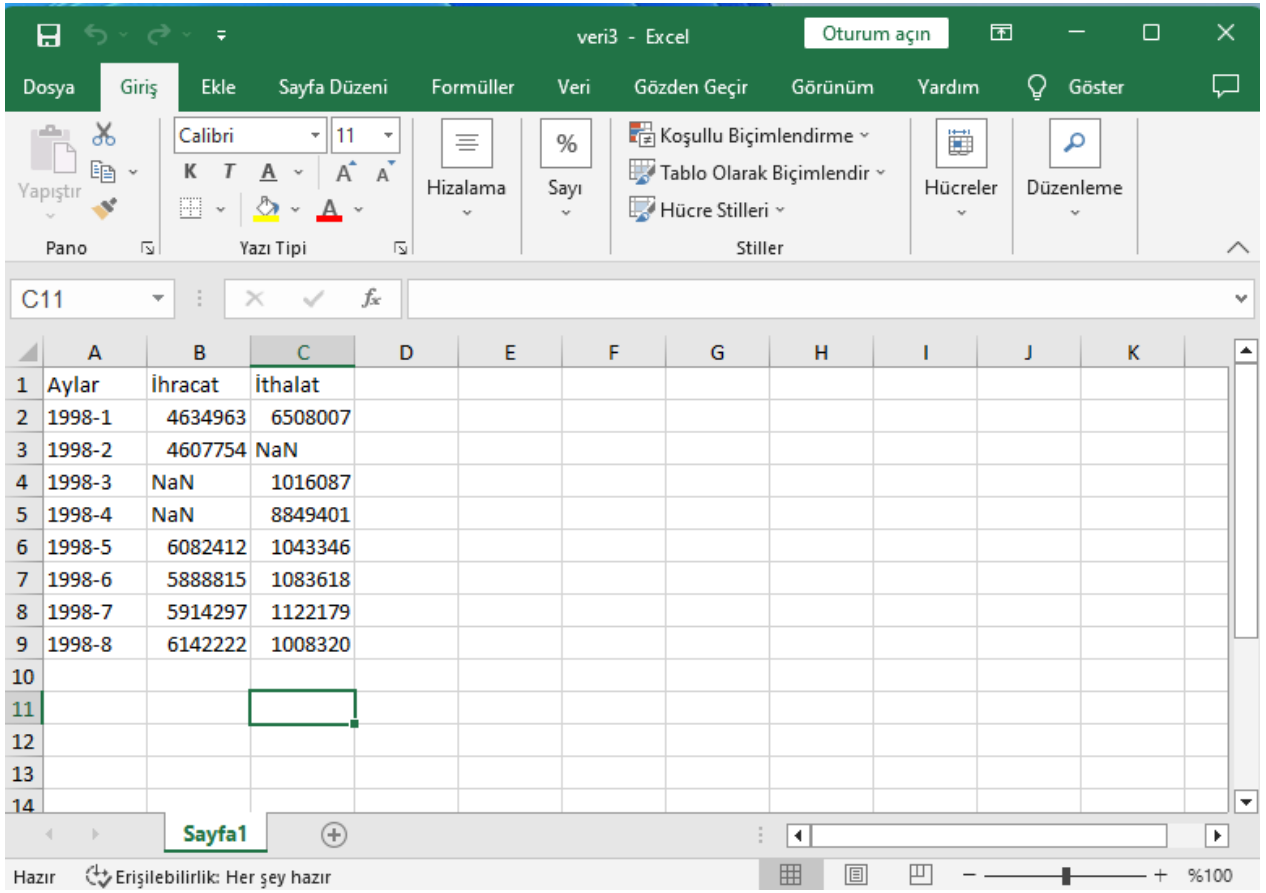
Pandas kütüphanesi açıklayıcı istatistiklerde eksik verileri hariç tutmaktadır.

Pandas çıktılarında eksik veriler ‘NaN’ olarak gösterilir.

‘dosya.isnull()’ sorgusuyla eksik olan verilerin dosyada taranması sağlanır.

Eksik Verilerin Düzenlenmesi

veri3.xlsx



	A	B	C
1	Aylar	İhracat	İthalat
2	1998-1	4634963	6508007
3	1998-2	4607754	NaN
4	1998-3	NaN	1016087
5	1998-4	NaN	8849401
6	1998-5	6082412	1043346
7	1998-6	5888815	1083618
8	1998-7	5914297	1122179
9	1998-8	6142222	1008320
10			
11			
12			
13			
14			

Eksik Verilerin Düzenlenmesi

```
import pandas as pd

veri=pd.read_excel("C:/Users/Hp/Desktop/veri3.xlsx", "Sayfa1")

veri.head()
```

Çıktı:

```
     Aylar     İhracat     İthalat
0   1998-1   4634963.0   6508007.0
1   1998-2   4607754.0         NaN
2   1998-3         NaN   1016087.0
3   1998-4         NaN   8849401.0
4   1998-5   6082412.0   1043346.0
```

Eksik Verilerin Düzenlenmesi

```
veri.isnull()
```

Çıktı:

	Aylar	İhracat	İthalat
0	False	False	False
1	False	False	True
2	False	True	False
3	False	True	False
4	False	False	False
5	False	False	False
6	False	False	False
7	False	False	False

Eksik Verilerin Düzenlenmesi

dropna: Eksik verilerin bulunduğu satırları çıkartır. Bir önceki uygulamada eksik veriler ‘dropna()’ modülü kullanılarak 1, 2 ve 3 numaralı satırlar çıkartılmıştır.

```
veri.dropna()
```

Çıktı:

```
    Aylar    İhracat    İthalat
0  1998-1  4634963.0  6508007.0
4  1998-5  6082412.0  1043346.0
5  1998-6  5888815.0  1083618.0
6  1998-7  5914297.0  1122179.0
7  1998-8  6142222.0  1008320.0
```

Eksik Verilerin Düzenlenmesi

fillna: Eksik verilerin belirtilen bir değerle doldurulmasını sağlar. ‘veri’ dosyası için eksik verilerin ‘0’ değeri ile doldurulması için kod satırı aşağıdaki gibidir:

```
veri.fillna(0)
```

Çıktı:

	Aylar	İhracat	İthalat
0	1998-1	4634963.0	6508007.0
1	1998-2	4607754.0	0.0
2	1998-3	0.0	1016087.0
3	1998-4	0.0	8849401.0
4	1998-5	6082412.0	1043346.0
5	1998-6	5888815.0	1083618.0
6	1998-7	5914297.0	1122179.0
7	1998-8	6142222.0	1008320.0

Eksik Verilerin Düzenlenmesi

Ayrıca farklı sütunlar için farklı değerler ile eksik verilerin doldurulması sağlanabilir:

```
veri.fillna({"İhracat": 0.5, "İthalat": 0})
```

Çıktı:

	Aylar	İhracat	İthalat
0	1998-1	4634963.0	6508007.0
1	1998-2	4607754.0	0.0
2	1998-3	0.5	1016087.0
3	1998-4	0.5	8849401.0
4	1998-5	6082412.0	1043346.0
5	1998-6	5888815.0	1083618.0
6	1998-7	5914297.0	1122179.0
7	1998-8	6142222.0	1008320.0

Eksik Verilerin Düzenlenmesi

Bu işlemler var olan ana dosya üzerinde herhangi bir değişim yapmamaktadır. Ancak modülde ‘inplace=True’ komutu eklenirse asıl dosya üzerinde de yapılan işlemler uygulanacaktır.

```
veri.fillna({"İhracat": 0.5, "İthalat": 0}, inplace=True)

veri
```

Çıktı:

	Aylar	İhracat	İthalat
0	1998-1	4634963.0	6508007.0
1	1998-2	4607754.0	0.0
2	1998-3	0.5	1016087.0
3	1998-4	0.5	8849401.0
4	1998-5	6082412.0	1043346.0
5	1998-6	5888815.0	1083618.0
6	1998-7	5914297.0	1122179.0
7	1998-8	6142222.0	1008320.0

Eksik Verilerin Düzenlenmesi

method='ffill': Doldurma veya ileri doldurma, gözlenen son boş olmayan değeri, boş olmayan başka bir değerle karşılaşana kadar ileriye doğru yayar.

```
import pandas as pd

veri=pd.read_excel("C:/Users/Hp/Desktop/veri3.xlsx", "Sayfa1")

veri.head()
```

Çıktı:

	Aylar	İhracat	İthalat
0	1998-1	4634963.0	6508007.0
1	1998-2	4607754.0	NaN
2	1998-3	NaN	1016087.0
3	1998-4	NaN	8849401.0
4	1998-5	6082412.0	1043346.0

Eksik Verilerin Düzenlenmesi

```
veri.fillna(method='ffill')
```

Çıktı:

	Aylar	İhracat	İthalat
0	1998-1	4634963.0	6508007.0
1	1998-2	4607754.0	6508007.0
2	1998-3	4607754.0	1016087.0
3	1998-4	4607754.0	8849401.0
4	1998-5	6082412.0	1043346.0
5	1998-6	5888815.0	1083618.0
6	1998-7	5914297.0	1122179.0
7	1998-8	6142222.0	1008320.0

Eksik Verilerin Düzenlenmesi

method='bfill': Bfill veya backward-fill, gözlenen ilk boş olmayan değeri geriye doğru başka bir boş olmayan değerle karşılaşana kadar yayar.

```
veri.fillna(method='bfill')
```

Çıktı:

	Aylar	İhracat	İthalat
0	1998-1	4634963.0	6508007.0
1	1998-2	4607754.0	1016087.0
2	1998-3	6082412.0	1016087.0
3	1998-4	6082412.0	8849401.0
4	1998-5	6082412.0	1043346.0
5	1998-6	5888815.0	1083618.0
6	1998-7	5914297.0	1122179.0
7	1998-8	6142222.0	1008320.0

Yinelemelerin Kaldırılması

Bazı durumlarda yinelenen değerlerin dosya içerisinde yer almaması gerekebilir.

Örneğin bir kümeleme işlemi sonucunda küme elemanları içerisinde yinelemeler yer almamalıdır.

‘dosya.dublicated’ modülü ile dosyada var olan yinelemeler tespit edilebilir.

Yinelemelerin Kaldırılması

```
import pandas as pd

veri = pd.read_csv ('C:/Users/ACER/küme.txt')
veri
```

Çıktı:

	k_1	k_2	v_1
0	bir	1	0
1	iki	1	1
2	bir	2	2
3	iki	3	3
4	bir	3	4
5	iki	4	6
6	bir	3	4

```
veri.duplicated()
```

Çıktı:

0	False
1	False
2	False
3	False
4	False
5	False
6	True

Yinelemelerin Kaldırılması

Son satırda yer alan verinin yinelendiği belirlenmiştir. ‘dosya.drop_duplicates()’ modülü yardımıyla yenilenen değerler kaldırılır.

```
veri. drop_duplicates ()
```

Çıktı:

	k_1	k_2	v_1
0	bir	1	0
1	iki	1	1
2	bir	2	2
3	iki	3	3
4	bir	3	4
5	iki	4	6

Verilerin Değiştirilmesi

Eksik verilerin değiştirilmesi haricinde veride var olan bazı değerler yanlış yazılmış olabilir.

Bu durumda var olan dosya üzerinde ‘dosya.replace(‘eskideğer’,’yenideğer’) modülü yardımıyla istenilen değişimler yapılabilir.

Verilerin Değiştirilmesi

```
import pandas as pd

veri=pd.read_csv('C:/Users/ACER/kitapdata/küme.txt')
veri
```

Çıktı:

	Cinsiyet	k_1	k_2	v_1
Ali	e	bir	1	0
Esra	k	bir	3	4
Ahmet	e	iki	1	1
Ayşe	k	bir	2	2
Emel	k	iki	3	3
Esra	k	bir	3	4
Veli	e	iki	4	6
Can	e	bir	3	4
Ahmet	e	iki	1	1

Verilerin Değiştirilmesi

Veride ilk önce sözel ifadeler için değişiklik yapılmıştır.

```
veri.replace ('bir', 'sekiz')
```

Çıktı:

	Cinsiyet	k_1	k_2	v_1
Ali	e	sekiz	1	0
Esra	k	sekiz	3	4
Ahmet	e	iki	1	1
Ayşe	k	sekiz	2	2
Emel	k	iki	3	3
Esra	k	sekiz	3	4
Veli	e	iki	4	6
Can	e	sekiz	3	4
Ahmet	e	iki	1	1

Verilerin Değiştirilmesi

Aynı işlem cinsiyet için erkek ve kadın verileri için sözel ifadelerden sayısala çevrilmiştir.

```
veri.replace ('e','1',inplace=True)
veri.replace ('k','2',inplace=True)
veri
```

Çıktı:

	Cinsiyet	k_1	k_2	v_1
Ali	1	sekiz	1	0
Esra	2	sekiz	3	4
Ahmet	1	iki	1	1
Ayşe	2	sekiz	2	2
Emel	2	iki	3	3
Esra	2	sekiz	3	4
Veli	1	iki	4	6
Can	1	sekiz	3	4
Ahmet	1	iki	1	1

Verilerin Değiştirilmesi

Sayısal veriler üzerinde değişim işlemi ‘k2’ sütunu için yapılmıştır. ‘k2’ sütunu için yapılmıştır.

‘inplace=True’ komutu ile veri çerçevesi üzerinde uygulama boyunca değişikliğin kalıcı olması sağlanmıştır.

```
veri.k2.replace(3,5,inplace=True)
veri
```

Çıktı:

	Cinsiyet	k_1	k_2	v_1
Ali	1	sekiz	1	0
Esra	2	sekiz	5	4
Ahmet	1	iki	1	1
Ayşe	2	sekiz	2	2
Emel	2	iki	5	3
Esra	2	sekiz	5	4
Veli	1	iki	4	6
Can	1	sekiz	5	4
Ahmet	1	iki	1	1

Satır Sütun İsimlerinin Değiştirilmesi

Uygulamanın yapıldığı dosya çıktısının üzerinde satır ve sütun isimleri değiştirilmek istenebilir. Bu durumda yeni bir veri yapısı oluşturmadan istenilen değişimler yapılabilir.

‘dosya.rename(index={‘eskiadı’:’yeniadı’}, columns={‘eskiadı’:’yeniadı’})’ modülü ile değişim gerçekleştirilir.

Eğer modüle ‘inplace=True’ eklenirse değişim dosya üzerine işlenir.

Satır Sütun İsimlerinin Değiştirilmesi

```
import pandas as pd

veri = pd.read_csv ('C:/Users/ACER/küme.txt')
veri
```

Çıktı:

	k_1	k_2	v_1
Ali	bir	1	0
Ahmet	iki	1	1
Ayşe	bir	2	2
Mehmet	iki	3	5
Kadir	bir	3	4
Veli	iki	4	6
Can	bir	3	4

```
veri.rename (index={'Ali':'Osman'},columns={'k1': 'sınav'})
```

Çıktı:

	sınav	k_2	v_1
Osman	bir	1	0
Ahmet	iki	1	1

Kukla Değişken

İstatistiksel modelleme veya makine öğrenimi uygulamalarında kategorik değişkenlerin analize uygun yapıya getirilmesi için kukla değişken tanımlanır. Pandan kütüphanesinde ‘get_dummies’ komutu ile kategorik verilere kukla değişkenler atanır.

```
import pandas as pd

veri = pd.read_csv ('küme.txt')
veri
```

Çıktı:

	Cinsiyet	k_1	k_2	v_1
Ali	e	bir	1	0
Ahmet	e	iki	1	1
Ayşe	k	bir	2	2
Mehmet	k	iki	3	5
Kadir	k	bir	3	4
Veli	e	iki	4	6
Can	e	bir	3	4

Kukla Değişken

Küme dosyasında yer alan ‘Cinsiyet’ sütununa kukla değişkenlerin ataması yapılarak yeni değerler var olan veri üzerinde sütun olarak aşağıdaki gibi belirtilir.

```
kukla=pd.get_dummies(veri['Cinsiyet'], prefix='Cinsiyet')
kukla
```

Çıktı:

	Cinsiyet_e	Cinsiyet_k
Ali	1	0
Ahmet	1	0
Ayşe	0	1
Mehmet	0	1
Kadir	0	1
Veli	1	0
Can	1	0

5-Değerlendirme (Evaluation)

- ✓ Veri üzerinde veri madenciliği uygulandıktan sonra alınan sonuçlar yorumlanır ve çalışmanın doğru sonuca ulaşıp ulaşmadığı araştırılır.
- ✓ Bu aşamada genellikle farklı yöntemler uygulanmışsa onların karşılaştırması yapılır.
- ✓ Elde edilen sonuçlar yapılmış olan diğer çalışmaların sonuçlarıyla karşılaştırılıp doğrulanır.