Makine Öğrenmesi Revizesi

Makine Öğrenmesi Nedir?

Makine Öğrenimi, bilgisayarları verilerden öğrenebilmeleri için programlama bilimidir (ve sanatıdır).

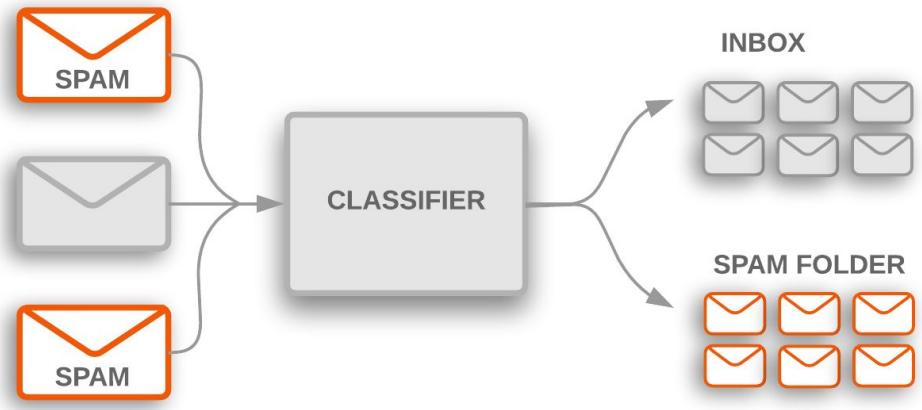
Makine Öğrenimi bilgisayarlara açıkça programlanmadan öğrenme yeteneği veren çalışma alanıdır.

—Arthur Samuel, 1959

Denetimli Öğrenme (Supervised Learning)

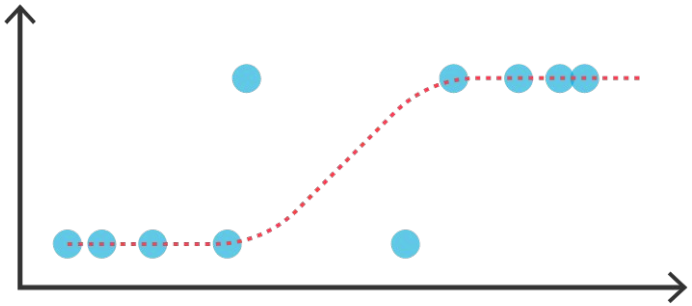
Supervised Learning; etiketlenmiş veri kümelerini bağımlı ve bağımsız değişkenler kullanarak işleme sokan makine öğrenmesi algoritmalarını içerir.

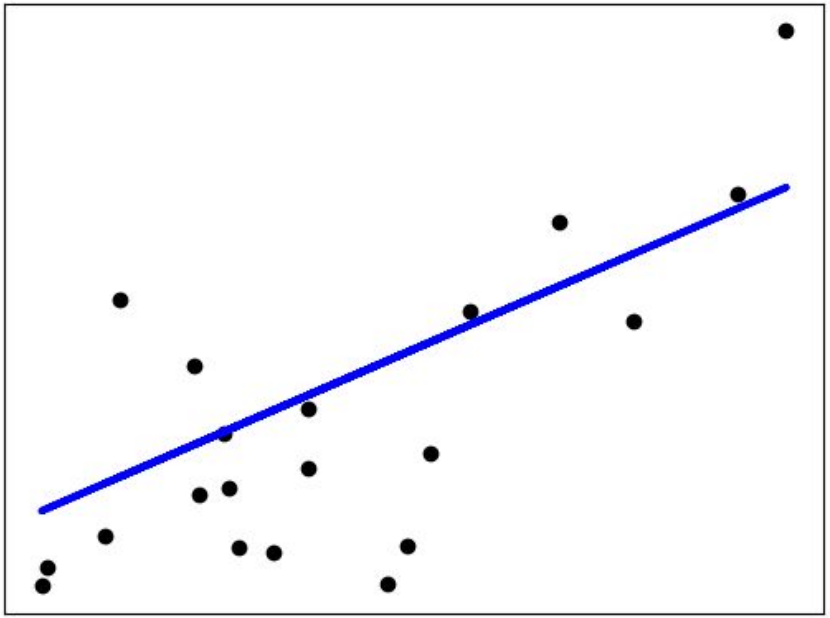
Örneğin; istenmeyen e-posta filtresi buna iyi bir örnektir: Denetimli Öğrenme sınıflarıyla birlikte birçok örnek e-posta ile eğitilmiştir ve yeni e-postaları nasıl sınıflandıracağını öğrenmelidir.



Denetimli Öğrenme Algoritma ve Mimarileri

- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural Networks

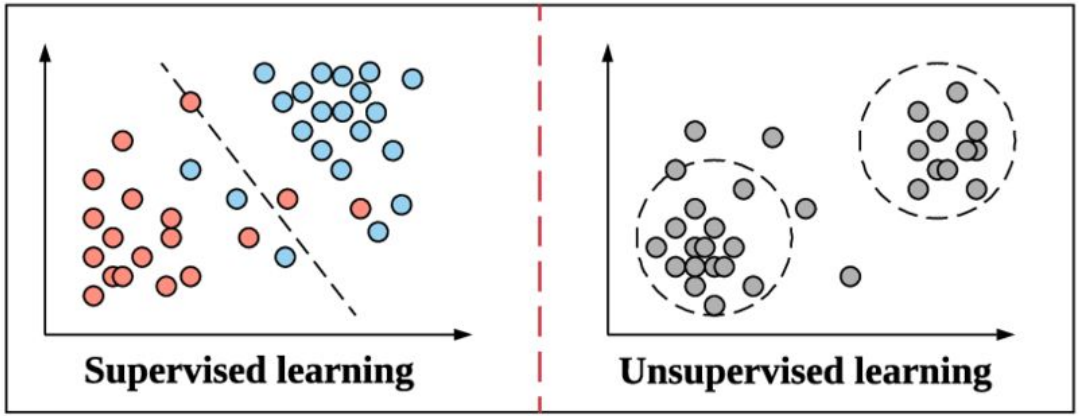




Denetimsiz Öğrenme (Unsupervised Learning)

Unsupervised Learning, etiketlenmemiş veri kümelerini analiz etmek ve kümelemek için makine öğrenimi algoritmalarını kullanır.

Bu algoritmalar, insan müdahalesine ihtiyaç duymadan gizli kalıpları veya veri gruplamalarını keşfeder.



Neden Unsupervised Learning Kullanıyoruz?

Unsupervised Learning için en yaygın görevler;

- Kümeleme
- Yoğunluk tahmini
- Temsili öğrenme

Unsupervised Learning Algoritmaları;

- Küme sayımızın bilinmediği
- Etiketli eğitim verimizin bulunmadığı

durumlarda kullanılabilir.

Örnekleri

Güvenlik:

Veri kümelerindeki olağandışı veri noktalarını tanımlandığı kümeleme anormalliği algılaması

Pazarlama:

Veri noktaları arasındaki ilişkileri bulduğu ilişki madenciliği

Unsupervised Learning Algoritmaları

Clustering

- Centroid-based
- Density-based Spatial Clustering (DBSCAN)
- Hierarchical Cluster Analysis (HCA)
- Affinity Propagation

Anomaly Detection and Novelty Detection

- One-class SVM
- Isolation Forest

Visualization and dimensionality reduction

- Principal Component Analysis (PCA)
- Kernel PCA
- Locally Linear Embedding (LLE)
- t-Distributed Stochastic Neighbor Embedding (t-SNE)

Association rule learning

- Apriori
- Eclat

Model Eğitiminde Karşılaşılabilecek Hatalar

Underfitting & Overfitting

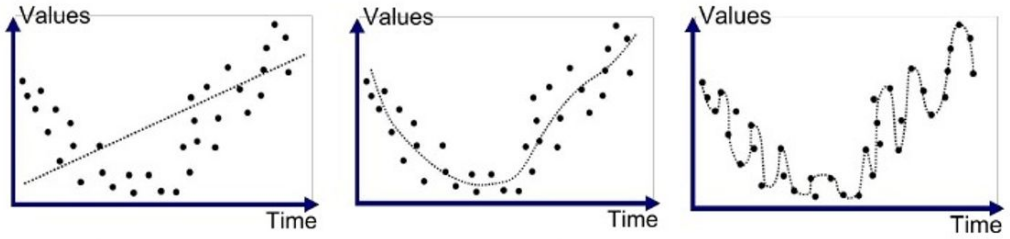
Underfitting & Overfitting

Underfitting

Makine Öğrenmesi modelinin verimizin trendini yakalayamadığında ortaya çıkan bir problemdir. Bir başka deyişle, modelimizin veriyi yeterince öğrenememesi veya veriden yeterli anlamı çıkartamamasıdır.

Overfitting

Makine Öğrenmesi modelinin, verimizin trendini yakalaması gerekenden daha çok yakaladığında ortaya çıkan bir problemdir. Bir başka deyişle, modelimiz ona verdiğimiz veriyi öğrenmekten çok ezberler ve daha önce görmediği veriler üzerinde başarılı bir çıkarım yapamaz.



Underfitted | Good Fit/Robust | Overfitted

Underfitting & Overfitting

- İstatistik ve Makine Öğreniminin önemli bir teorik sonucu, bir modelin genelleme (görmediği verileri kullanarak tahmin gerçekleştirebilme yeteneği) hatasının çok farklı üç hatanın toplamı olarak ifade edilebilmesidir:
  - Bias
  - Variance
  - İndirgenemez Hatalar
- Bias'ın fazla olması varyansın az olması anlamına gelmektedir ve bu da underfit'e örnektir
- Varyansın fazla olması bias'ın az olması anlamına gelmektedir ve bu da overfit'e örnektir
- İndirgenemez hatalar verideki noise ile alakalıdır ve çözmenin tek yolu veriyi temizlemektir

✯ Underfit olan bir model **düşük varyans, yüksek bias** değerlerine sahiptir.

✯ Overfit olan bir **model yüksek varyans, düşük bias** değerlerine sahiptir.

Bias - Variance Tradeoff

- Bias (yanlılık) ve Variance (varyans), modelin farklı training setlerinde (aynı daha büyük popülasyondan) birçok kez yeniden eğitilirse performansının nasıl olacağının ölçütleridir.
- Underfitting durumunda, model aynı popülasyondan alınmış neredeyse her training seti için çok fazla hata yapacaktır, bu da yüksek bias değerine sahip olduğu anlamına gelir. Ancak, rastgele seçilen iki training seti oldukça benzer modeller vermelidir (çünkü rastgele seçilen herhangi iki training seti oldukça benzer ortalama değerlere sahip olmalıdır). Bu yüzden de varyansının düşük olduğu söylenir.
- Overfitting durumunda, model eğitim setine mükemmel bir şekilde uymaktadır. Model, çok az hata verdiği için düşük bir bias değerine sahiptir. Ancak, herhangi iki eğitim seti overfitting durumunda muhtemelen çok farklı modellere yol açacağı için çok yüksek varyansa sahip olacaktır.

✯ Bias arttıkça varyans azalmaktadır.

✯ Varyans arttıkça bias azalmaktadır.

Bias/Variance Tradeoff

Bias

Bias, modelimizin ortalama tahmini ile tahmin etmeye çalıştığımız doğru değer arasındaki farktır. Yüksek bias'a sahip model, eğitim verilerine çok az dikkat eder ve modeli aşırı basitleştirir. Eğitim ve test verilerinde her zaman yüksek hataya yol açar. Bu underfit olarak adlandırılabilir

- ✯ Bias arttıkça varyans azalmaktadır.

Variance

Varyans, belirli bir veri noktası için model tahmininin değişkenliğidir. Karmaşıklığı yüksek olan bir modelin yüksek varyansa sahip olması ve dolayısıyla eğitim verisine fazla uyması test setindeki tahminlerde hataya yol açar bu da overfit olarak adlandırılabilir

- ✯ Varyans arttıkça bias azalmaktadır.

Engellemek İçin Neler Yapılabilir?

Underfitting

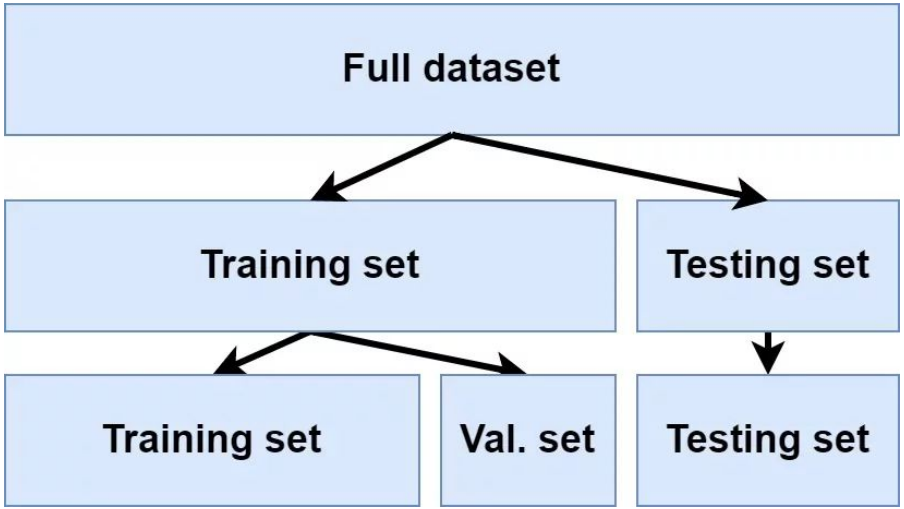
1. Yeni feature değerleri veya katmanlar eklenerek modelin karmaşıklığı arttırılabilir
2. Veriden gürültü temizlenebilir
3. Eğitim sırasında epoch sayısı arttırılabilir
4. Daha iyi feature değerleri verilebilir (Feature Engineering)
5. Modeli sınırlandıran değerler azaltılabilir (örneğin regülarizasyon parametresi)

Overfitting

1. Eğitim verisi arttırılabilir
2. Modelin karmaşıklığı azaltılabilir
3. Eğitim sırasında erken durdurma (early stopping) yapılabilir
4. Regülarizasyon uygulanabilir

Verimizi Eğitim-Test-Validasyon Olarak Ayırmak

1. Elimizdeki veri modelimize eğitim için gönderilir. Modelimiz elindeki veriyi öğrenir ve öğrendiği verinin üstünden tekrar tekrar geçerek kendisini geliştirir *(forward-backward prop.)*
2. Eğitim tamamen bittikten sonra model kendini skorlayabilmek için öğrenimde kullandığı veri ile kendini ölçer
3. Eğer skor yüksek çıkarsa modelimiz başarılı demektir
4. Modele eğitim için verilen veri arasında bulunmayan bir veri verildiğinde modelin aslında iddia ettiği kadar başarılı tahminle yapamadığını görülebilir



Veri Analizi ve Makine Öğrenmesinde Kullanılan Araçlar

Veri Analizi ve Makine Öğrenmesinde Kullanılan Araçlar

Python, genel amaçlı bir programlama dilidir.

Yorumlanan ve dinamik bir dil olan Python, esas olarak nesne tabanlı programlama yaklaşımlarını ve fonksiyonel programlamayı desteklemektedir.

- Hızlı prototipleme
- Basit syntax
- Kolay kullanım
- Geniş topluluk

Linear Regression

```
from sklearn.tree import DecisionTreeClassifier

DTC = DecisionTreeClassifier(criterion='gini',
max_features=10, max_depth=5)

DTC = DTC.fit(X_train, y_train)

y_predict = DTC.predict(X_test)
```

Veri Analizi ve Makine Öğrenmesinde Kullanılan Araçlar

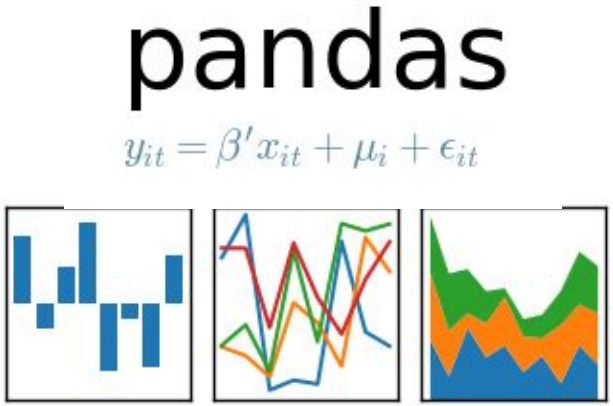
NumPy, Python'da bilimsel hesaplamalarda kullanılan temel pakettir.

- Dizi oluşturma
- Vektörleştirme ve dilimleme
- Matrisler ve basit lineer cebir
- Veri dosyaları

Veri Analizi ve Makine Öğrenmesinde Kullanılan Araçlar

Pandas, veri analizi ve veri ön işlemeyi kolaylaştıran açık kaynak kodlu bir Python kütüphanesidir.

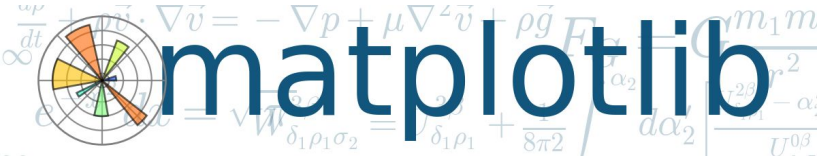
- Veri manipülasyonu için kullanışlı fonksiyonlar
- Farklı biçimler arasında veri okuma ve yazma araçları: CSV ve metin dosyaları, Microsoft Excel, SQL veritabanları
- Basit seviyede hızlı veri görselleştirme



Veri Analizi ve Makine Öğrenmesinde Kullanılan Araçlar

Matplotlib, Python programlama dili için bir veri görselleştirme ve çizim kütüphanesidir.

- Matplotlib grafik çizim paketi Python'la bilimsel programlamanın en önemli araçlarından birisidir
- Çok kuvvetli bir paket olan Matplotlib ile verileri etkileşimli olarak görselleştirebilir
- Basıma ve yayınlanmaya uygun yüksek kalitede çıktılar hazırlayabiliriz
- Hem iki boyutlu hem de üç boyutlu grafikler üretilebilir



Veri Analizi ve Makine Öğrenmesinde Kullanılan Araçlar

Scikit-learn, Python programlama dili için ücretsiz bir yazılım makinesi öğrenme kütüphanesidir.

Doğrusal regresyon, lojistik regresyon, karar ağaçları, rastgele orman gibi birçok temel yöntemi bünyesinde bulundurur.

https://scikit-learn.org/stable/

Machine Learning with Scikit-Learn

Linear Regression

```
from sklearn.tree import DecisionTreeClassifier

DTC = DecisionTreeClassifier(criterion='gini',
max_features=10, max_depth=5)

DTC = DTC.fit(X_train, y_train)

y_predict = DTC.predict(X_test)
```

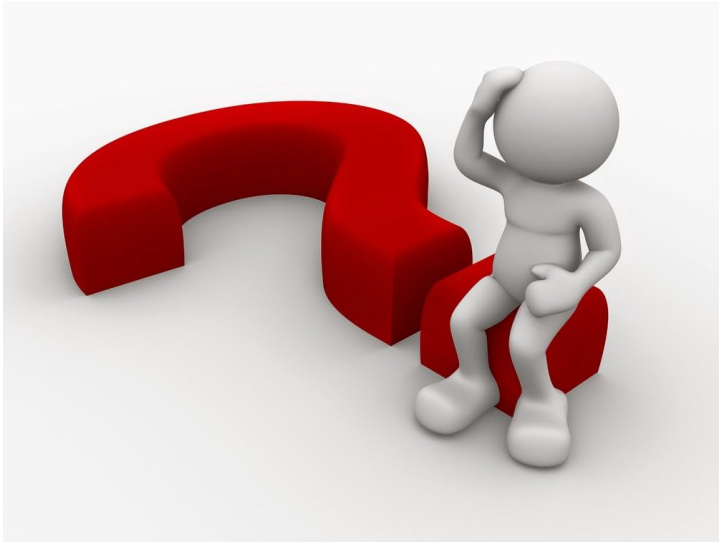
Uçtan Uca Makine Öğrenmesi Projesi

Proje Adımları

1. Problem Tanımı
2. Veriyi Toplamak
3. EDA & Ön işleme
4. Eğitim
5. ML Modellerini Kıyaslama
6. Tahmin

1. Problem Tanımı

- İlk adım problemin tanımlanmasıdır
- İkinci adım, sorunun neden çözülmesini istediğinizi veya buna ihtiyaç duyduğunuzu derinlemesine düşünmektir
- Sorun tanımının üçüncü ve son adımında, sorunu manuel olarak nasıl çözeceğinizi keşfedin



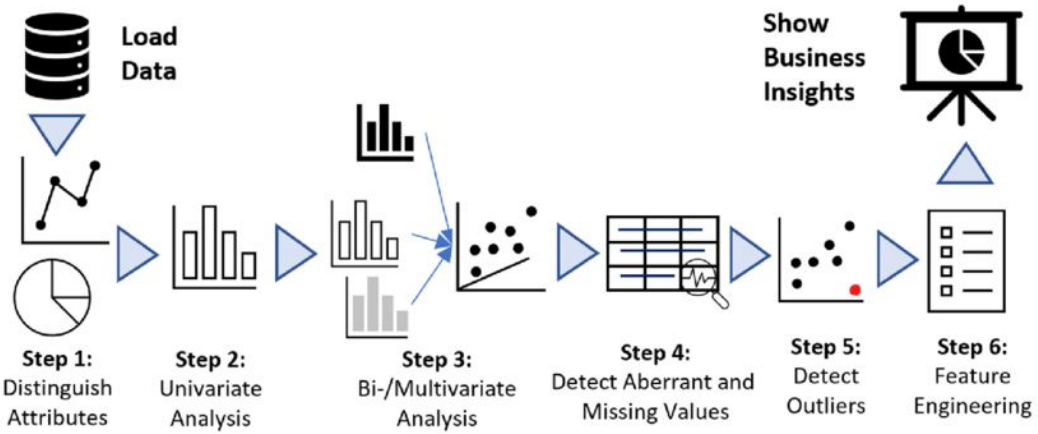
2. Veriyi Toplamak

Bu gerçek hayat örneğinde, Melbourne Konut veri setine göre bir evin fiyatını tahmin edeceğiz.

Herhangi bir fiyatı tahmin etmeden önce, ön işleme tekniklerini kullanarak verileri analiz ederek başlıyoruz.

3. EDA & Veriyi Ön İşlemek

Keşifçi Veri Analizi, kalıpları keşfetmek, anormallikleri tespit etmek, hipotezi test etmek ve özet istatistikler ve grafik gösterimler yardımıyla varsayımları kontrol etmek için veriler üzerinde ilk araştırmaları gerçekleştirmenin kritik sürecini ifade eder.



3.1 Veriyi Ön İşlemek

- Veriyi elde edin
- Tüm önemli kütüphaneleri içe aktarın
- Veri kümesini içe aktarın
- Eksik değerlerin belirleyin ve ele alın
- Kategorik verileri encode edin
- Veri kümesini bölün

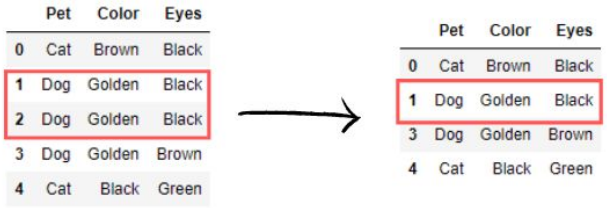
3.2 Tekrarlı Kayıtları Silmek

Duplicated ve drop_duplicates yöntemi, subset adlı bir parametre alır. Bir sütuna özgü kopyaları kontrol etmek gerekirse, kullanabilirsiniz.

name	region	sales	expense
William	East	50000	42000
William	East	50000	42000
Emma	North	52000	43000
Emma	West	52000	43000
Anika	East	65000	44000
Anika	East	72000	53000

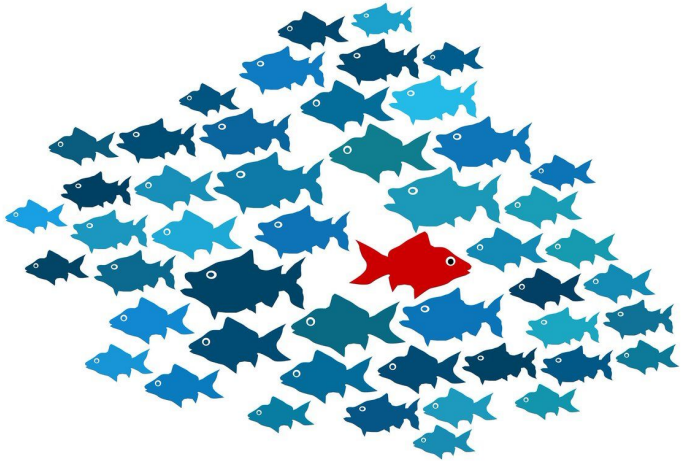
NOTE 1: **duplicated(subset=[column names])**

NOTE 2: **drop_duplicates(subset=[column names])**



3.3 Aykırı Değer Tespiti

Aykırı değerler, veriler üzerindeki diğer gözlemlerden sapan uç değerlerdir, bir ölçümdeki değişkenliği, deneysel hataları veya bir yeniliği gösterebilirler. Başka bir deyişle, aykırı değer, bir örnek üzerindeki genel bir modelden ayrılan bir gözlemdir.



Z-Skoru ile Aykırı Değerlerin Tespiti ve NA/NaN Değerlerinin Doldurulması

- Z-Skoru bir gözlemin ortalamadan kaç standart sapma uzaklıkta olduğudur
- Bir gözlemin Z-puanı sıfırdan ne kadar uzaktaysa, o kadar olağandışıdır. Veri setindeki bir değişkenin z puanı 3'ten büyük veya -3'ten küçük ise değişkenin aykırı değere sahip olduğunu söyleyebiliriz
- NA/NaN değerlerini düşürmek yerine onları doldurmak daha verimlidir

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean

σ = Standard Deviation

3.4 Label Encoder & One-Hot Encoder

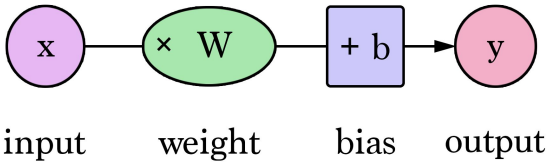
- Label Encoder, etiketleri/kelimeleri sayısal forma dönüştürmektir
- Label Encoding, veri kümesinin boyutunu etkilemez

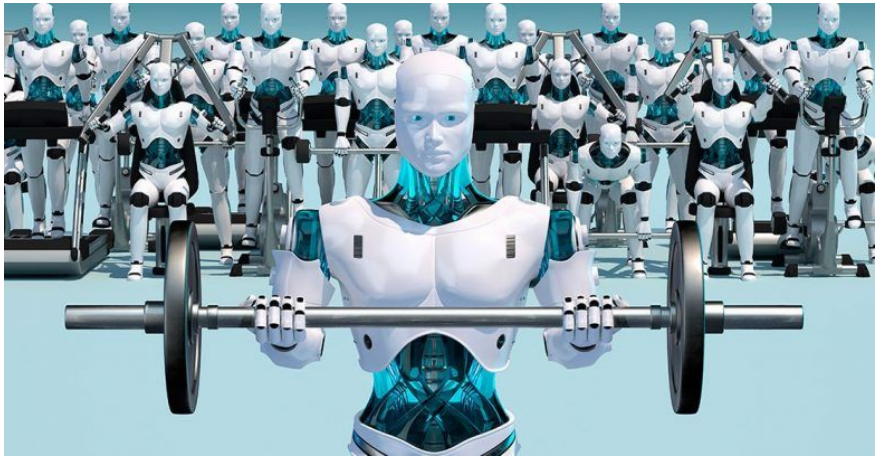
- One-hot encoding kategorik değişkenlerin ikili vektörler olarak gösterimidir.
- One-hot encoding veri kümesinin boyutunu arttırır

NOT: **Makine öğrenimi algoritmaları, bu kodlamaların nasıl çalıştırılması gerektiğine daha iyi karar verebilir.**

4. Eğitim

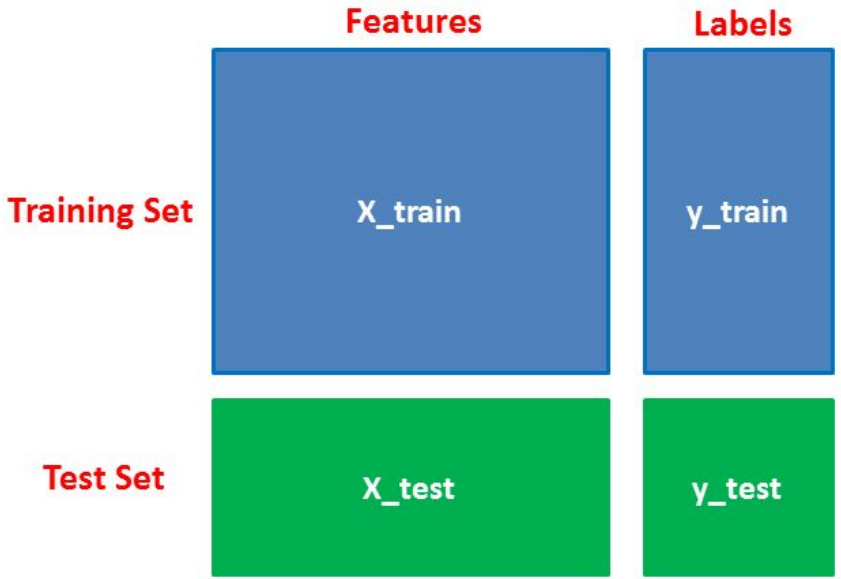
Bir modeli eğitmek, etiketlenmiş örnekleri kullanaraken iyi ağırlık ve bias değerlerini öğrenmek (belirlemek) anlamına gelir.





4.1 Verileri Özniteliklere ve Etikete Bölme

- Bir modeli eğitmek için önce verileri özellikler ve etiketler olarak ayırmak gerekir
- Modeli test ederken verileri eğitim ve test olarak bölmek faydalı olabilir, algoritmanın henüz görmediği verileri kullanmak, farklı problemleri belirlemek için faydalıdır

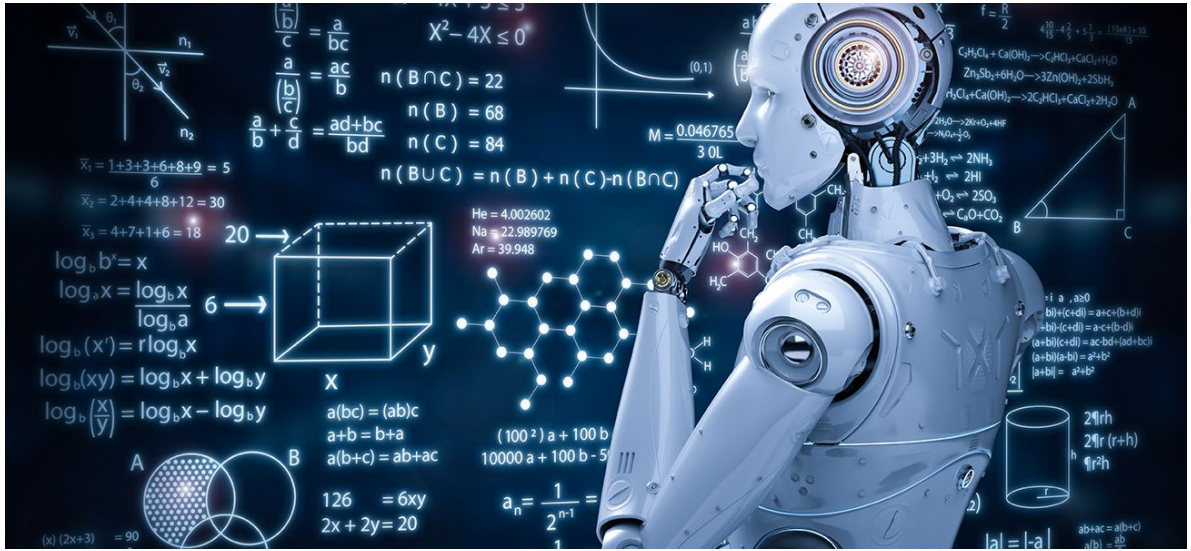


4.2 Gradient Boosting Modeli Kullanmak

Gradient Boosting algoritması, yalnızca sürekli hedef değişkeni (Regressor olarak) değil, aynı zamanda kategorik hedef değişkeni de (Sınıflandırıcı olarak) tahmin etmek için kullanılabilir. Regresör olarak kullanıldığında hata fonksiyonu Ortalama Kare Hatası (MSE) ve sınıflandırıcı olarak kullanıldığında maliyet fonksiyonu Log Loss'tur.

5. ML Modellerini Kıyaslamak

Tüm algoritmaları tek tek deneyebilir, ardından hangi makine öğrenme algoritmasının probleminize en uygun olduğuna karar vermek için puanları ve hataları karşılaştırabilirsiniz.



6. Tahmin

Tahmin, bir algoritmanın geçmiş bir veri kümesi üzerinde eğitildikten ve belirli bir sonucun olasılığını tahmin ederken yeni verilere uygulandıktan sonra çıktısını ifade eder