

1. Regresyon Nedir?

Regresyon, **sürekli (numerik) bir değeri tahmin etmek** için kullanılan bir makine öğrenmesi teknigidir.

- Bağımsız değişkenler (özellikler) ile bağımlı değişken (hedef) arasındaki ilişkileri öğrenir.
- Çıkış değişkeni **sayı (kesintisiz değer)** olmalıdır.
- Kullanım alanları: Fiyat tahmini, hava sıcaklığı tahmini, satış miktarı tahmini vb.

Örnek Regresyon Problemleri:

- Bir evin **fiyatını** tahmin etmek (Metrekare, oda sayısı gibi değişkenlere bağlı olarak).
- Önümüzdeki ay bir markette **kaç ürün satılacağını** tahmin etmek.
- Bir çalışanın **maasını** deneyim yılına göre tahmin etmek.

Yaygın Regresyon Algoritmaları:

- **Linear Regression (Doğrusal Regresyon)**
- **Polynomial Regression (Polinom Regresyon)**
- **Ridge Regression, Lasso Regression**
- **Decision Tree Regressor, Random Forest Regressor**
- **XGBoost, LightGBM (Gelişmiş ağaç tabanlı algoritmalar)**

Linear Regression (Doğrusal Regresyon)

Tanım:

Linear Regression, **bağımsız değişkenler (X) ile bağımlı değişken (Y) arasında doğrusal bir ilişki olduğunu varsayan ve sürekli (numerik) bir değer tahmin eden bir makine öğrenmesi modelidir**

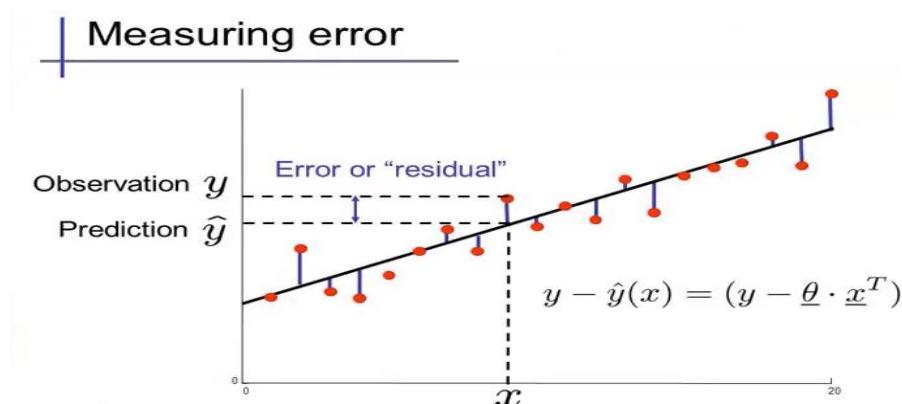
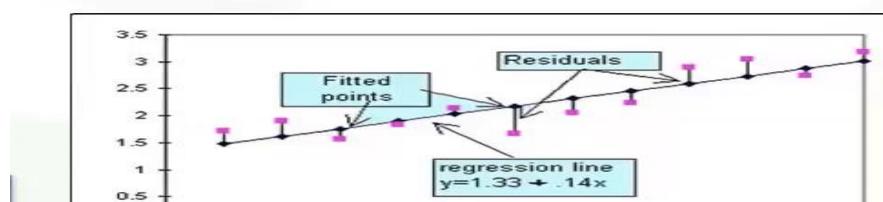
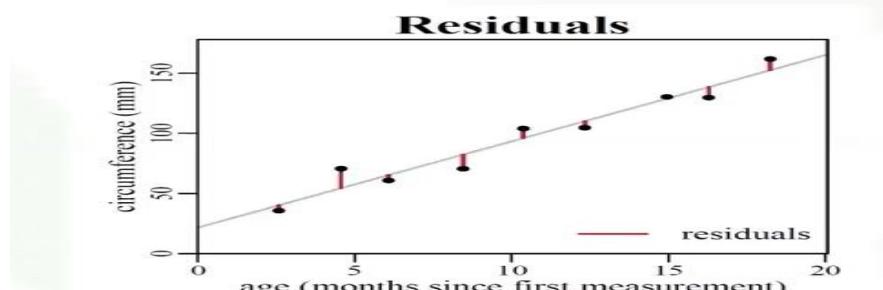
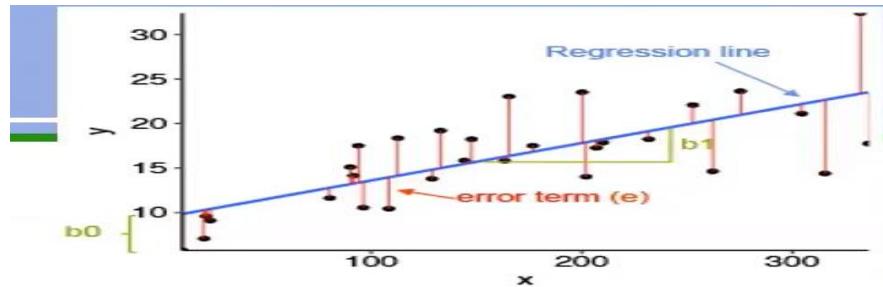
❖ Örnek 2: Maaş Tahmini

- **Problem:** Çalışanların maaşlarını, deneyim yılına göre tahmin etmek istiyoruz.
- **Bağımsız Değişkenler (X): Deneyim yılı (1, 2, 3, 5, 10 yıl...)**
- **Bağımlı Değişken (Y): Maaş (TL) (₺10,000, ₺15,000, ₺25,000...)**

Model şöyle olabilir:

$$Maas | c=5,000 + 2,500 \times (\text{Deneyim Yılı}) \\ Maas = 5,000 + 2,500 \times (\text{Deneyim Yılı}) \\ Maas = 5,000 + 2,500 \times (\text{Deneyim Yılı})$$

- **5 yıl deneyimi olan biri için:** $Maas | c=5,000 + 2,500(5) = 17,500 \text{ TL}$ $Maas = 5,000 + 2,500(5) = 17,500 \text{ TL}$



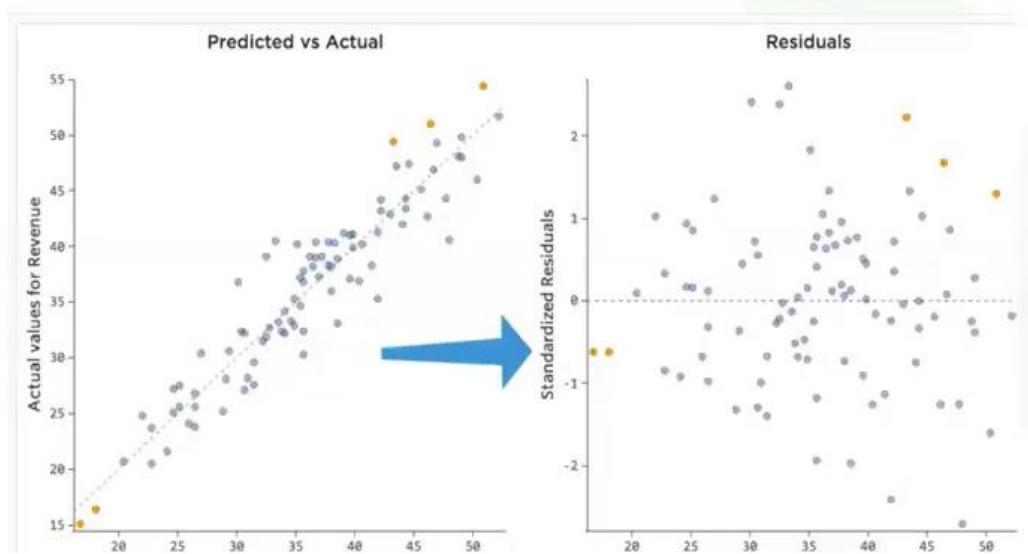
Residuals: gerçek noktaya doğru üzerinde denk geldiği nokta arasındaki fark

Error de denilebilir

Amacımız bu residuals'ları minimize etmek

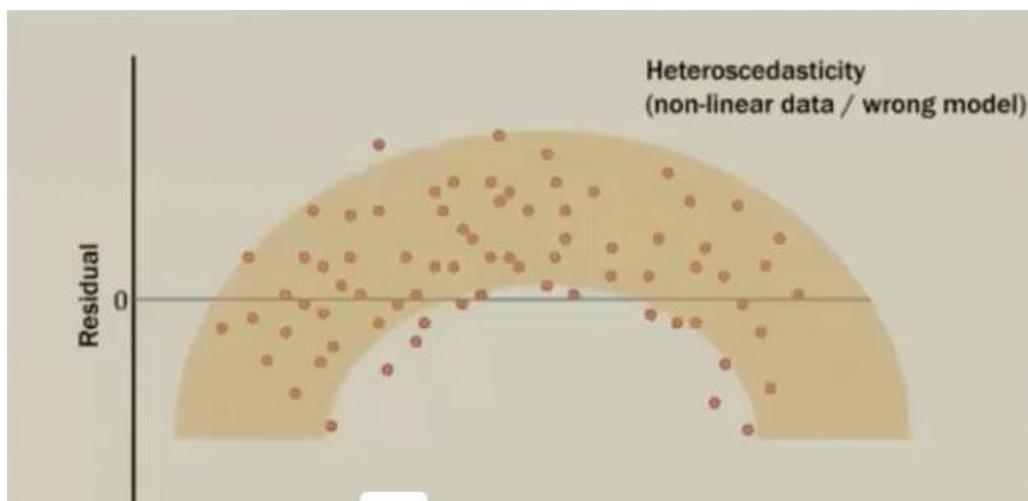
Bilgi kaybına sebep olurlar

Standardize edersek dağılım bu şekilde olur



pattern sergilememeli

Bu şekilde olursa nonlinear data olduğunu gösterir



Simple Linear Regression - Multiple Linear Regression

Basit vs. Çoklu Doğrusal Regresyon Farkı

Özellik	Basit Doğrusal Regresyon	Çoklu Doğrusal Regresyon
Bağımsız Değişken Sayısı	1	2 veya daha fazla
Denklem Yapısı	$Y = b_0 + b_1 X + \epsilon$	$Y = b_0 + b_1 X_1 + b_2 X_2 + \dots + b_n X_n + \epsilon$
Ömek	Maaş = 3000 + 1500 × Deneyim	Maaş = 2500 + 1200 × Deneyim + 3000 × Eğitim Seviyesi
Kullanım Alanı	Tek bir faktörün etkisini ölçmek	Birden fazla faktörün etkisini analiz etmek

Polynomial Regression

Polinom regresyon, **bağımlı değişken (Y) ile bağımsız değişken (X) arasındaki ilişki doğrusal olmadığından** kullanılan bir regresyon modelidir.

Polinom regresyonda bağımsız değişkenin üstleri (karesi, küpü vb.) modele eklenir:

$$Y = b_0 + b_1 X + b_2 X^2 + b_3 X^3 + \dots + b_n X^n + \epsilon$$

Burada:

- X^2, X^3, \dots, X^n terimleri, bağımsız değişkenin polinom derecelerini temsil eder.
- n değeri, modelin derecesini belirler.
- Daha yüksek dereceli polinomlar, **daha karmaşık eğrileri** yakalamaya yardımcı olur.

Örnek 1: Araç Yakıt Tüketimi

Bir arabanın hızının **yakıt tüketimi üzerindeki etkisini** inceleyelim.

Düşük hızlarda (0-50 km/saat) yakıt tüketimi düşüktür.

Orta hızlarda (50-100 km/saat) yakıt tüketimi daha verimlidir.

Çok yüksek hızlarda (100+ km/saat) yakıt tüketimi hızla artar.

Bu durumda **doğrusal regresyon uygun olmaz**, çünkü hız ve yakıt tüketimi arasındaki ilişki **doğrusal değildir**. Bunun yerine **2. dereceden polinom regresyon** kullanabiliriz:

$$Yakıt = 8 + 0.2 \times Hız - 0.001 \times Hız^2$$

$$Hız^2 Yakıt = 8 + 0.2 \times Hız - 0.001 \times Hız^2$$

Eğer hız 80 km/saat ise:

$$\begin{aligned} Yakıt &= 8 + (0.2 \times 80) - (0.001 \times 80^2) \\ Yakıt &= 8 + (0.2 \times 80) - (0.001 \times 80^2) = 8 + 16 - 6.4 = 17.6 \\ L/100km &= 8 + 16 - 6.4 = 17.6 \text{ L/100km} \end{aligned}$$

Eğer hız 120 km/saat ise:

$$\begin{aligned} Yakıt &= 8 + (0.2 \times 120) - (0.001 \times 120^2) \\ Yakıt &= 8 + (0.2 \times 120) - (0.001 \times 120^2) = 8 + 24 - 14.4 = 17.6 \\ L/100km &= 8 + 24 - 14.4 = 17.6 \text{ L/100km} \end{aligned}$$

Bu model, düşük hızlarda ve çok yüksek hızlarda yakıt tüketiminin arttığını, ancak orta hızlarda daha verimli olduğunu gösterir.

Basit, Çoklu ve Polinom Regresyon Karşılaştırması

Model Türü	Denklem Yapısı	Kullanım Durumu
Basit Doğrusal Regresyon	$Y = b_0 + b_1 X$	X ile Y arasında doğrusal bir ilişki varsa kullanılır.
Çoklu Doğrusal Regresyon	$Y = b_0 + b_1 X_1 + b_2 X_2 + \dots$	Birden fazla bağımsız değişkenin Y üzerindeki etkisini incelemek için kullanılır.
Polinom Regresyon	$Y = b_0 + b_1 X + b_2 X^2 + b_3 X^3 + \dots$	X ile Y arasında doğrusal olmayan (eğrisel) bir ilişki varsa kullanılır.

- **Eğer iki değişken arasındaki ilişki doğrusal ise**, basit veya çoklu doğrusal regresyon kullanılabilir.
- **Eğer ilişki eğrisel ise (U şeklinde, çan eğrisi gibi)**, polinom regresyon daha iyi sonuç verebilir.
- **Dereceyi (n) fazla artırmak** modele aşırı uyum (overfitting) riski getirebilir.

2. Correlation

Korelasyon, **iki değişken arasındaki ilişkinin yönünü ve gücünü ölçen** istatistiksel bir kavramdır.

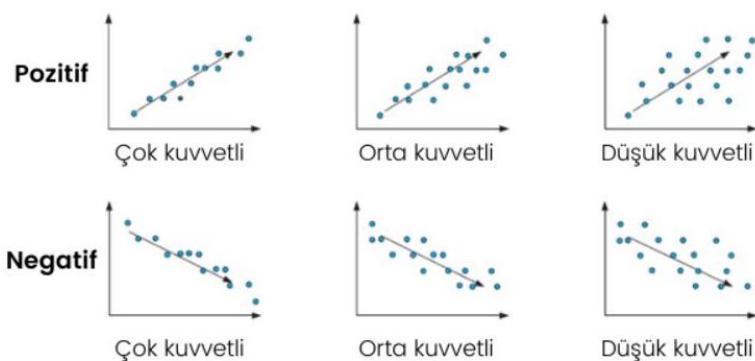
- Korelasyon **pozitif, negatif veya sıfır** olabilir.
- Korelasyon **neden-sonuç ilişkisi (nedensellik)** anlamına gelmez, sadece değişkenler arasındaki birlikte değişimi gösterir.

Korelasyon Türleri

1. **Pozitif Korelasyon (+):** Bir değişken artarken diğer değişken de artıyorsa.
 - a. Örnek: Çalışma süresi $\uparrow \rightarrow$ Sınav notu \uparrow
2. **Negatif Korelasyon (-):** Bir değişken artarken diğer değişken azalıyorsa.
 - a. Örnek: Telefon kullanımı $\uparrow \rightarrow$ Uyku süresi \downarrow
3. **Sıfır (Yok) Korelasyon (0):** İki değişken arasında bir ilişki yoksa.
 - a. Örnek: Ayakkabı numarası ile sınav notu

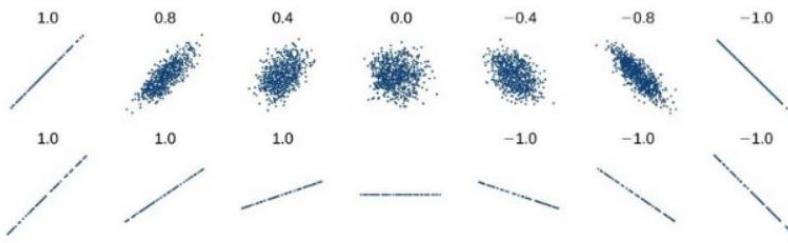
Korelasyon, genellikle **Pearson Korelasyon Katsayısı (r_{Pearson})** ile ölçülür

Doğrusal (Lineer) Korelasyon



Korelasyon katsayı değeri	Yorumu
0.9 ile 1 veya -0.9 ile -1 arası	Çok kuvvetli doğrusal ilişki
0.7 ile 0.9 veya -0.7 ile -0.9 arası	Kuvvetli doğrusal ilişki
0.5 ile 0.7 veya -0.5 ile -0.7 arası	Orta kuvvetli doğrusal ilişki
0.3 ile 0.5 veya -0.3 ile -0.5 arası	Düşük kuvvetli doğrusal ilişki
0.0 ile 0.3 veya 0.0 ile -0.3 arası	Hic ilişki yok veya çok düşük kuvvetli doğrusal ilişki

Farklı Kuvvetlere Sahip Doğrusal Korelasyon



Örnek 1: Maaş ve Deneyim Arasındaki Korelasyon

Bir çalışanın **deneyim yılı (X)** ve **maası (Y)** arasındaki ilişkiyi inceleyelim:

Deneyim (Yıl)	Maaş (TL)
1	5000
2	5500
3	6000
4	6700
5	7500

Bu verilerle hesaplanan r_{xy} değeri yaklaşık **0.95** çıkarsa:

- Çok güçlü pozitif korelasyon** → Deneyim arttıkça maaş da artıyor.

Özellik Seçimi (Feature Selection)

- Machine Learning modellerinde **çok yüksek korelasyona sahip değişkenleri** birlikte kullanmak gereksiz olabilir.
- Multicollinearity (Çoklu Bağlantı)** → Çok korelasyonlu değişkenler, regresyon modellerini bozabilir.

Veri Ön İşleme

- Korelasyon analizi ile **hangi değişkenlerin etkili olduğunu** belirleyebiliriz.
- Gereksiz (sıfır korelasyonlu) değişkenleri çıkararak modelin performansını artırabiliriz.

Tahmin Modellerinde Kullanımı

- Regresyon modellerinde** bağımlı değişkenle yüksek korelasyonu olan bağımsız değişkenler tercih edilir.
- Zayıf korelasyonlu değişkenler** modele pek katkı sağlamaz.

3. Multicollinearity

Multicollinearity, bir regresyon modelinde bağımsız değişkenlerin birbirleriyle yüksek derecede korelasyonlu olması durumudur.

- Yani **bir bağımsız değişken başka bir bağımsız değişken tarafından tahmin edilebiliyorsa**, multicollinearity problemi vardır.
- Bu durum, **regresyon modellerinin güvenilirliğini düşürür** ve tahmin değişkenlerinin etkilerini yorumlamayı zorlaştırır.

Örnek 1: Ev Fiyat Tahmini

Bir evin fiyatını tahmin etmeye çalışırken şu değişkenleri ele alalım:

- Ev Alanı (m²)**
- Oda Sayısı**
- Salon Sayısı**

Burada **Ev Alanı (m²)** ile **Oda Sayısı** arasında **yüksek bir korelasyon** olabilir. Çünkü büyük evler genellikle daha fazla odalıdır.

Aynı şekilde **Salon Sayısı da genellikle Oda Sayısı ile bağlantılıdır**.

Bu durumda **Oda Sayısı ve Salon Sayısını aynı anda modele eklemek** multicollinearity oluşturabilir.

Örnek 2: Çalışan Maaşı Tahmini

Bir çalışanın maaşını tahmin ederken şu değişkenleri düşünelim:

- Deneyim (Yıl)**
- Çalıştığı Süre (Ay olarak)**
- Şirket içindeki Kıdem**

- **Deneyim ve Çalıştığı Süre arasında yüksek korelasyon vardır** çünkü birinin artışı diğerinin artışına sebep olur.
- **Deneyim ve Kıdem de ilişkili olabilir**, çünkü kıdem genellikle deneyime bağlıdır.

Bu durumda **hem deneyimi hem de çalıştığı süreyi aynı modele koymak gereksiz olabilir** ve modelin doğruluğunu bozabilir.

Regresyon Katsayılarının (Coefficients) Güvenilirliğini Azaltır

- Bağımsız değişkenler yüksek korelasyona sahipse, **hangi değişkenin hedef değişken üzerinde daha fazla etkili olduğu belirsizleşir.**
- Örneğin, **Deneyim (Yıl)** ve **Çalıştığı Süre (Ay)** aynı anda modele girildiğinde, model **hangi değişkenin gerçekten maaşı etkilediğini anlamakta zorlanır.**

Özellik Seçimini (Feature Selection) Zorlaştırır

- Eğer bir veri setinde **çok fazla korelasyonlu değişken varsa, bazı değişkenleri modelden çıkarmak** daha iyi bir tahmin gücü sağlayabilir.

Overfitting (Aşırı Öğrenme) Riskini Artırır

- Çoklu bağıntılı değişkenler modele gereksiz bilgi yükleyerek **genelleştirme performansını düşürebilir.**

Nasıl Tespit Edilir?

Korelasyon Matrisi Kullanımı

Bir veri setindeki değişkenler arasındaki korelasyonu hesaplayarak çoklu bağıntıyı tespit edebiliriz.

```
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')  
plt.show()
```

Mesela ; Deneyim_Yıl ve Çalışma_Süresi_Ay arasında 0.99 gibi çok yüksek bir korelasyon var.



Bu durumda birini modelden çıkarmak mantıklı olabilir.

4.Error Metrics

- Regresyon Metrikleri:** MAE, MSE, RMSE, R²
- Sınıflandırma Metrikleri:** Accuracy, Precision, Recall, F1 Score
- Hangi metrik seçileceği modele ve probleme bağlıdır.**

Modelin ne kadar iyi tahmin yaptığıni veya hatalarının boyutunu belirlemek için bu metriklerden faydalanzıız.

Hata metrikleri, **süervizyonlu öğrenme (Supervised Learning)** algoritmalarında iki ana gruba ayrırlı:

1. **Regresyon Hata Metrikleri** (Sürekli değerleri tahmin eden modeller için)
2. **Sınıflandırma Hata Metrikleri** (Kategorik sınıfları tahmin eden modeller için)

1. Regresyon İçin Hata Metrikleri

(örneğin, ev fiyatı tahmini, hava sıcaklığı tahmini).

1. Mean Absolute Error (MAE) - Ortalama Mutlak Hata

- Tahmin edilen değer ile gerçek değer arasındaki farkların **mutlak ortalamasıdır**.
- ✓ Avantajı:** Kolay yorumlanabilir, **birim hassasiyetinde hata ölçer**.
✗ Dezavantajı: Büyük hataları kücümseyebilir.

KOD:

```
from sklearn.metrics import mean_absolute_error  
  
gercek = [100, 200, 300, 400, 500] tahmin = [110, 190, 310, 390, 480]  
  
mae = mean_absolute_error(gercek, tahmin) print("MAE:", mae)
```

2. Mean Squared Error (MSE) - Ortalama Kare Hata

- Hata karelerinin ortalamasıdır.
- Büyük hatalara daha fazla ceza verir.

- ✓ Avantajı:** Büyük hataları cezalandırarak modelin daha dikkatli öğrenmesini sağlar.
✗ Dezavantajı: Sonuç **verinin birim karesinde** olduğu için yorumlamak zor olabilir.

KOD:

```
from sklearn.metrics import mean_squared_error  
  
mse = mean_squared_error(gercek, tahmin)  
  
print("MSE:", mse)
```

3. Root Mean Squared Error (RMSE) - Kök Ortalama Kare Hata

- MSE'nin karekökü alınarak bulunur, böylece veriyle aynı birimde olur.
- Büyük hataları cezalandırır.

KOD:

```
import numpy as np  
rmse = np.sqrt(mse)  
print("RMSE:", rmse)
```

0'a yakın olması modelimizin gerçek satış değerlerine yakın olduğunu gösterir.

4. R² Score (Determination Coefficient - R Kare Skoru)

- Modelin bağımsız değişkenleri ne kadar iyi açıkladığını ölçer.
- 1'e yakınsa model iyidir, 0'a yakınsa model kötü tahmin yapmaktadır.

Simple regression da kullanılır

İlerleyen derslerde scikit learn kütüphanesi ile elde edeceğiz

KOD:

```
from sklearn.metrics import r2_score  
r2 = r2_score(gercek, tahmin)  
print("R^2 Skoru:", r2)
```

Avantajı: Modelin başarı oranını gösterir.

Dezavantajı: Çok fazla değişken içeren modellerde **overfitting** riski olabilir.

2. Sınıflandırma İçin Hata Metrikleri

(örneğin, e-postanın spam olup olmadığını belirleme, hastanın hasta olup olmadığını tahmin etme).

1. Accuracy (Doğruluk Oranı)

- Tüm doğru tahminlerin toplam tahminlere oranıdır.
- Formül: $Accuracy = \frac{\text{Dogru Tahminler}}{\text{Toplam Tahminler}}$

```
R2_score = corr ** 2
```

KOD:

```
from sklearn.metrics import accuracy_score
```

```
gercek = [1, 0, 1, 1, 0, 1, 0, 1]
```

```
tahmin = [1, 0, 0, 1, 0, 1, 1, 1]
```

```
accuracy = accuracy_score(gercek, tahmin)
```

```
print("Accuracy:", accuracy)
```

✓ **Avantajı:** Genel performansı ölçer.

✗ **Dezavantajı:** Veri dengesizse yaniltıcı olabilir.

2. Precision (Kesinlik)

Pozitif tahminlerin ne kadar doğru olduğunu gösterir.

KOD:

```
from sklearn.metrics import precision_score  
precision = precision_score(gercek, tahmin)  
print("Precision:", precision)
```

3. Recall (Duyarlılık)

- **Gerçek pozitifleri ne kadar iyi tahmin ettiğimizi gösterir.**

KOD:

```
from sklearn.metrics import recall_score
```

```
recall = recall_score(gercek, tahmin)  
print("Recall:", recall)
```

5. Overfitting Underfitting

- ✓ **İyi Bir Model** → Hem eğitim hem de test verisinde **iyi genelleştirme** yapar.
- ✗ **Overfitting (Aşırı Uyum)** → Model, **eğitim verisine çok fazla uyum sağlar ve genelleştirme yapamaz.**
- ✗ **Underfitting (Eksik Uyum)** → Model, **veriyi iyi öğrenemez, hem eğitim hem de test verisinde kötü performans gösterir.**

1 Overfitting (Aşırı Uyum) Nedir?

Bir modelin eğitim verilerine çok iyi uyum sağlaması, ancak yeni ve görülmemiş verilere karşı zayıf performans göstermesidir. Model, eğitim verisindeki gürültü ve rastgele dalgalanmaları öğrenir.

Overfitting, modelin **eğitim verisine aşırı uyum sağlaması** durumudur.

Model eğitim verisine çok iyi uyar, fakat yeni verilere karşı zayıf.

✗ **Eğitim verisinde yüksek doğruluk, ancak test verisinde düşük doğruluk** görülür.

✓ Overfitting'i Önleme Yöntemleri:

- **Daha fazla veri toplamak** (modelin genellemesini artırır).
- **Özellik seçimi yapmak** (önemsiz değişkenleri çıkarmak).
- **Daha basit modeller kullanmak**

Örnek:

- Bir model, bir veri kümesindeki her noktayı tam olarak tahmin ediyorsa (örneğin, bir polinom modelinin yüksek dereceli olması), bu genellikle overfitting'e işaret eder.
- Örneğin, bir ev fiyatı tahmin modeli, eğitim verisindeki her evin fiyatını doğru tahmin edebilir, ancak yeni evler için tahminleri çok yanlıştır.

◆ Gerçek Hayattan Örnek:

Bir öğrencinin **sadece geçmiş sınav sorularını ezberlemesi** ancak farklı sorular geldiğinde başarısız olması overfitting'e benzer.

2 Underfitting (Yetersiz Uyum) Nedir?

Bir modelin eğitim verilerine yeterince uyum sağlamaması ve bu nedenle hem eğitim hem de test verilerinde kötü performans göstermesidir. Model, verinin temel yapısını öğrenememiştir.

Model, verinin temel yapısını öğrenemez ve her iki veri setinde de kötü performans gösterir.

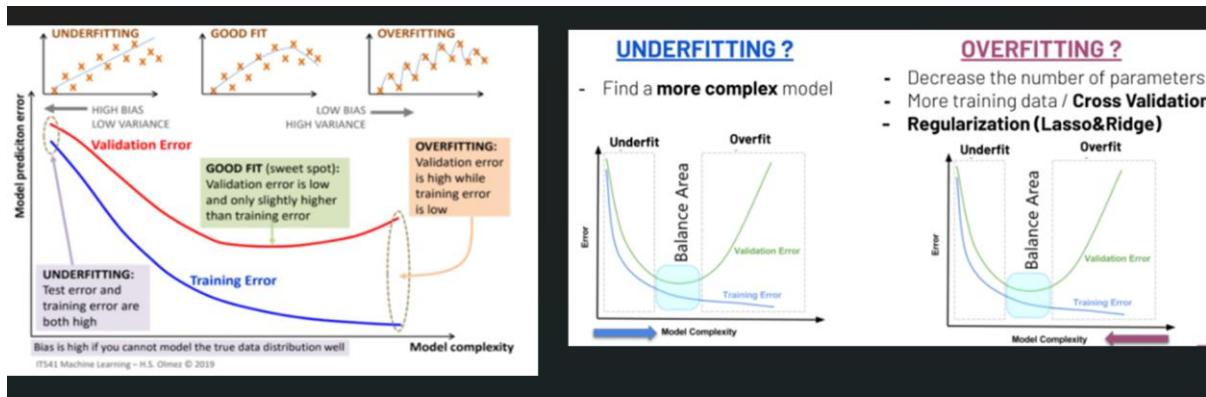
Overfitting'i Önleme Yöntemleri:

- Daha karmaşık modeller kullanmak (polinom regresyonu gibi)
- Modelin daha fazla özellik kullanmasını sağlamak

Örnek:

- Bir model, verilerin karmaşıklığını yeterince yakalayamıyorsa (örneğin, lineer bir modelin karmaşık bir ilişkiyi açıklamaya çalışması), bu underfitting'e işaret eder.
- Örneğin, bir ev fiyatı tahmin modeli, tüm evleri ortalama bir fiyatla tahmin ediyorsa, bu yetersiz bir modeldir.





6. Train Test Split

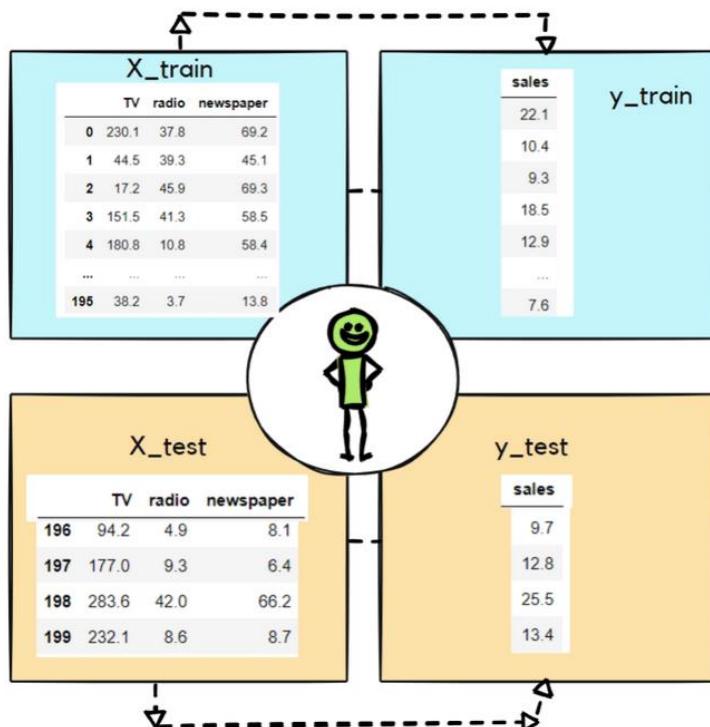
Bir modelin performansını değerlendirmek için verilerin iki ayrı küme olarak bölünmesini ifade eder. Bu yöntem, modelin genel performansını ölçmek ve aşırı öğrenmeyi (overfitting) önlemek için kullanılır.

- Datamızın genelleştirme yeteneğini ölçmek
- Overfitting durumlarını kontrol etmek

- Data arttıkça train artar
- Data seti azaldıkça test miktarını arttırmalıyız ki overfitting oluşmasın
- Train de 0.92 olan test de 0.55 düştü ne olur? Overfitting
- Train R2 ile test R2 arasında fark varsa overfitting

Terim	Açıklama	Örnek
`X_train`	Modeli eğitmek için kullanılan girdi veri seti. Arabaların teknik özellikleri, yaşı, kilometresi, markası vb.	Eğitim setindeki arabaların kilometre, yaş, marka bilgileri
`y_train`	`X_train` veri setine karşılık gelen gerçek satış fiyatları	Eğitim setindeki arabaların gerçek satış fiyatları
`X_test`	Modelin genelleme yeteneğini test etmek için kullanılan girdi veri seti. Arabaların teknik özellikleri, yaşı, kilometresi, markası vb.	Test setindeki arabaların kilometre, yaş, marka bilgileri
`y_test`	`X_test` veri setine karşılık gelen gerçek satış fiyatları	Test setindeki arabaların gerçek satış fiyatları

X_train, X, test, Y_train, y_test



7. Cross Validation

Cross Validation (Çapraz Doğrulama), **makine öğrenmesi modellerinin genellemeye performansını değerlendirmek için kullanılan bir tekniktir.**

- Modelin **overfitting (aşırı öğrenme) yapıp yapmadığını** kontrol etmek için veriyi **farklı bölmelere (folds)** ayırarak eğitim ve test işlemlerini tekrarlar.
- **Gerçek dünyada modelin nasıl performans göstereceğini daha doğru tahmin etmek için kullanılır.**

Neden Cross Validation Kullanılır?

- ❖ **Problemler:**
- ✓ **Eğer tüm veriyi eğitim için kullanırsak, model ezberleyebilir (overfitting) ve yeni verilerde başarısız olabilir.**

- Eğer tek bir test seti kullanırsak, model **yanıltıcı sonuçlar verebilir** (veri seti şans eseri çok iyi veya çok kötü olabilir).

📌 **Çözüm: Cross Validation!**

- Veriyi farklı eğitim ve test parçalarına böler.
- Modelin her veri bölmesiyle çalışmasını sağlayarak **daha adil ve güvenilir** bir doğruluk ölçümü yapar.

Cross Validation Türleri

1 Holdout Validation (Basit Veri Bölme)

Veri şu şekilde ayrılır:

- %80 **Eğitim (Train) Seti**
- %20 **Test Seti**

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Avantajı: Hızlıdır.

Dezavantajı: Veri setine bağlı olarak hatalı genellemelere yol açabilir.

2 K-Fold Cross Validation

3 Stratified K-Fold Cross Validation

4 Leave-One-Out Cross Validation (LOOCV)

Makine Öğrenmesinde Cross Validation Kullanımı

1 Overfitting'i Önler:

- Tek bir test seti yerine farklı bölmelerde test yapıldığı için modelin **aşırı öğrenmesini (overfitting) azaltır.**

2 Model Seçiminde Kullanılır:

- Farklı algoritmaların performansını ölçmek için cross-validation sonuçları karşılaştırılabilir.
- Örneğin, bir **Random Forest** mı yoksa **Logistic Regression** mi daha iyi?

3 Hiperparametre Optimizasyonu ile Kullanılır:

- **Grid Search ve Random Search gibi tekniklerle** cross-validation kullanılarak en iyi parametreler bulunur.

Özet

- ✓ Cross Validation, **modelin genelleme performansını ölçmek için veriyi farklı bölmelere ayırır.**
 - ✓ **K-Fold Cross Validation** en yaygın kullanılan yöntemdir.
 - ✓ **Stratified K-Fold**, dengesiz veri setlerinde kullanılır.
 - ✓ **LOOCV**, en güvenilir ama en yavaş yöntemdir.
 - ✓ **Model seçiminde, overfitting kontrolünde ve hiperparametre optimizasyonunda kullanılır.**

8.Encoding

Encoding (kodlama), **makine öğrenmesi modellerinin kullanabilmesi için kategorik verileri sayısal formata dönüştürme işlemidir.**

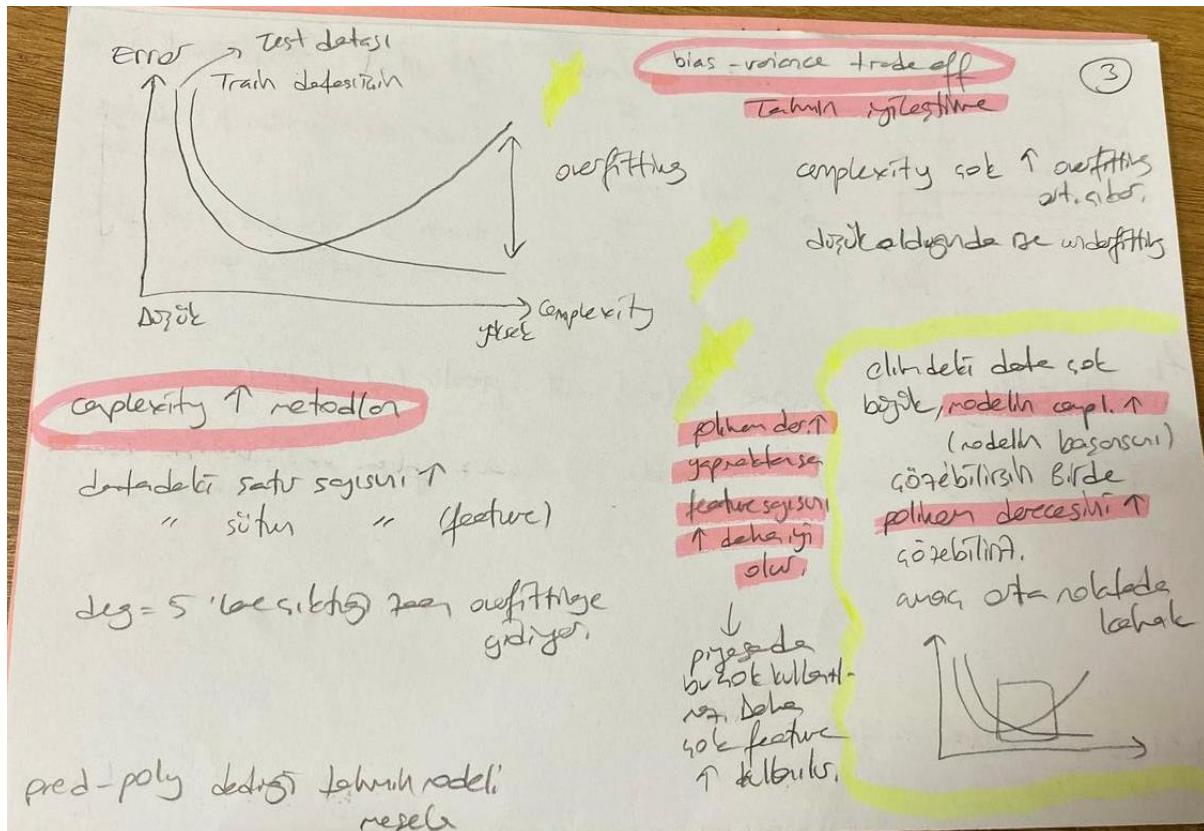
- Kategorik değişkenleri **doğru formatta encode etmek**, modelin **daha iyi öğrenmesini** sağlar.
- **Yanlış encoding yöntemi**, modelin hatalı öğrenmesine neden olabilir.

Algoritma	Kullanılabilecek Encoding
Linear Regression	One-Hot Encoding, Ordinal Encoding
Decision Tree	Label Encoding, One-Hot Encoding
Random Forest	Label Encoding, One-Hot Encoding
Logistic Regression	One-Hot Encoding, Ordinal Encoding
Deep Learning (NN)	One-Hot Encoding, Label Encoding

One-Hot Encoding sıralaması olmayan kategoriler için uygundur.

Label Encoding, kategorik değerleri sayılaraya çevirir ama sıralama problemi yaratabilir.

9. Bias Variance Trade-Off



Makine öğreniminde, bias (önyargı) ve variance (varyans) trade-off, modelin performansını etkileyen iki önemli kavramdır. Bu kavramlar, modelin öğrenme yeteneği ile genel performans arasındaki dengeyi ifade eder.

1. Bias (Önyargı)

- **Tanım:** Modelin gerçek değerleri tahmin etme yeteneği. Yüksek bias, modelin veriyi yeterince iyi öğrenemediğini gösterir.
- **Sonuç:** Aşırı basit bir model, verinin karmaşıklığını yeterince temsil edemez ve bu da hatalı tahminlere yol açar (örneğin, düz bir çizgiyle karmaşık bir veri kümesini modellemeye çalışmak).

2. Variance (Varyans)

- **Tanım:** Modelin eğitim verisine ne kadar duyarlı olduğunu ifade eder. Yüksek varyans, modelin eğitim verisine çok iyi uyum sağladığını, ancak yeni verilere kötü performans gösterdiğini gösterir.

- *Sonuç:* Aşırı karmaşık bir model, eğitim verisindeki gürültüyü öğrenir ve bu da test verisi üzerinde kötü sonuçlar doğurur.

3. Bias-Variance Trade-off

Bias ve varyans, modelin performansını etkileyen iki zıt güçtür. Genellikle, bias ve varyans arasında bir denge kurmak gereklidir:

- *Yüksek Bias, Düşük Varyans:* Basit modeller (örneğin, lineer regresyon) genellikle yüksek bias ve düşük varyansa sahiptir. Bu modeller, veriyi yeterince iyi öğrenemez.
- *Düşük Bias, Yüksek Varyans:* Karmaşık modeller (örneğin, derin öğrenme modelleri) genellikle düşük bias ve yüksek varyansa sahiptir. Bu modeller, eğitim verisine çok iyi uyum sağlar, ancak genel performansları düşer.

Model Seçimi: Farklı model türleri arasında seçim yaparken bias-variance trade-off dikkate alınmalıdır.

Sonuç

Bias-variance trade-off, makine öğreniminde model performansını optimize etmek için kritik bir kavramdır. Doğru dengeyi sağlamak, modelin hem eğitim verisi hem de test verisi üzerinde iyi performans göstermesini sağlar.

10. Regularization