



# **Yapay Zeka I: Veri Bilimi ve Makine Öğrenmesine Giriş Sertifika Programı**

Doç. Dr. Taner Arsan

H. Fuat Alsan, PhD(c)

Sena Kılınç, PhD(c)

# Train/Test Split

- Veri seti iki alt gruba ayrılmıştır: eğitim seti ve test seti
- Eğitim seti (Training set):
  - Makine öğrenimi modelini eğitmek için kullanılır
  - Model bu küme içindeki desenleri (pattern) ve ilişkileri öğrenir
- Test seti (Testing set):
  - Eğitim sırasında kullanılmaz
  - Modelin performansını yeni, görülmemiş veriler üzerinde test etmek için kullanılır
- **Genelleme (Generalization): Modelin yeni, görülmemiş verilere ne kadar iyi sonuçlar verdiği**
- K-Fold Cross-Validation: çoklu train/test setleri

# Underfitting, Overfitting

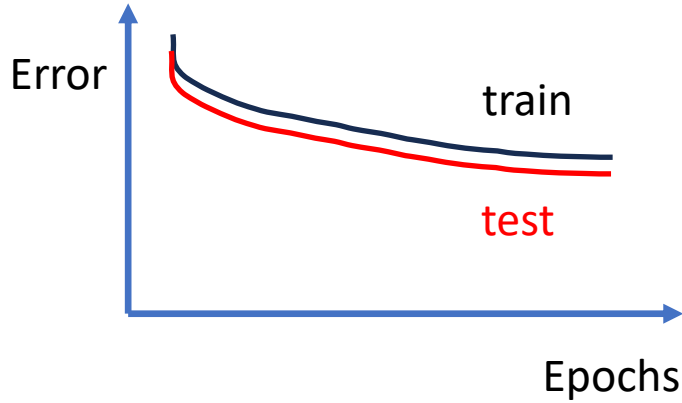
- **Overfitting:**

- Model eğitim verilerinde son derece iyi performans gösterir ancak yeni, görülmemiş verilerde zayıf performans gösterir (**genelleştirilmemiş**)
- Model gereğinden fazla karmaşıktır ve fazla parametre içerir

- **Underfitting:**

- Model çok basit kalır ve verideki desenleri (pattern) öğrenemez
- Model hem eğitim hem test kümesinde zayıf performans gösterir
- Seçilen model veriyi öğrenmek için fazlaca basittir

# Underfitting, Overfitting (Görsel)

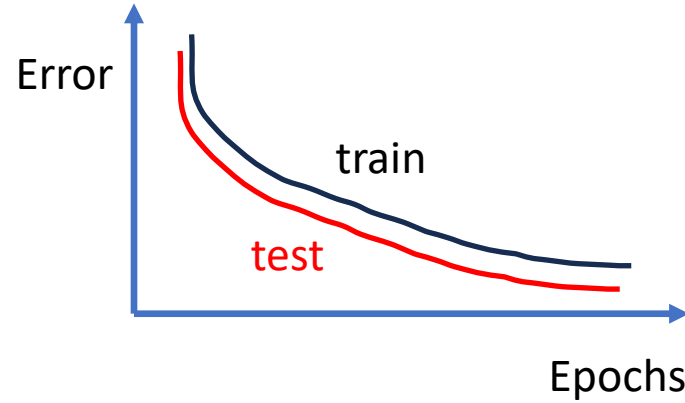


## Underfitting

Hem eğitim hem de test hatası yüksektir

Model, verileri öğrenemeyecek kadar basit

Daha gelişmiş bir model gereklidir

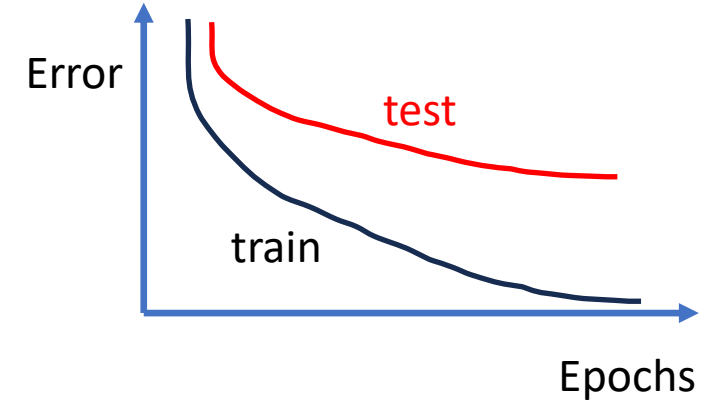


## Optimal Fitting

Hem eğitim hem de test hatası düşüktür

Model verileri iyi öğrenir ve genelleme yapabilir

En iyi sonuçtur



## Overfitting

Eğitim hatası düşük ama test hatası yüksektir

Model verileri iyi öğreniyor ancak genelleştiremiyor

Model çok karmaşık veya genelleme yapmak için yeterli veri yok

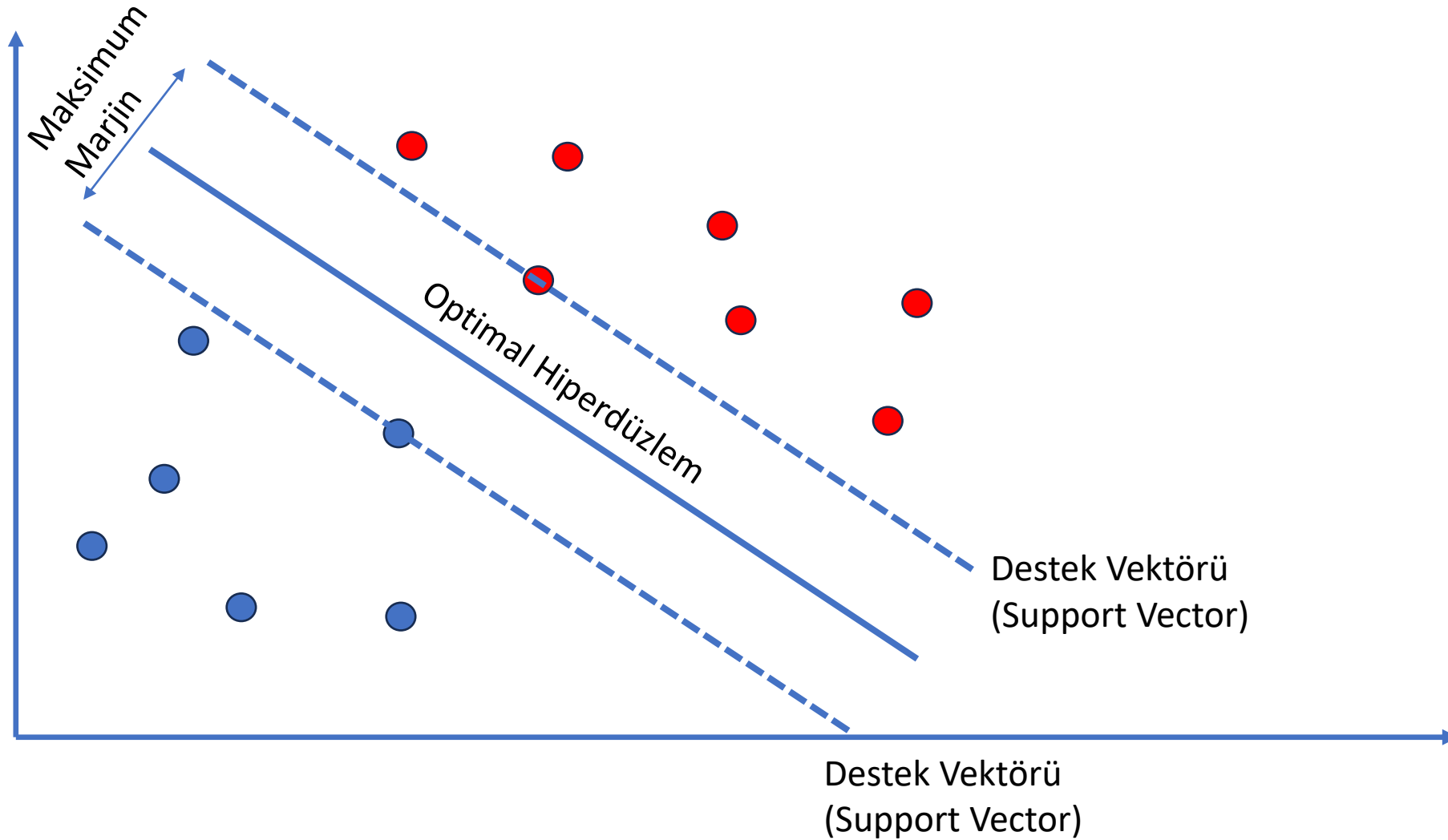
# Naïve Bayes Classifier

- Sınıflandırma için kullanılır
- Bayes Teoremi:  $P(y|x) = \frac{P(x|y)P(y)}{P(x)}$ 
  - $P(y|x)$ :  $x$  özellikleri verildiğinde  $y$  sınıfı olasılığı
  - $P(x|y)$ :  $y$  sınıfı verildiğinde  $x$  özellikleri olasılığı
  - $P(y)$ :  $y$  sınıfının olasılığı
  - $P(x)$ :  $x$  özelliklerinin olasılığı
- Naïve varsayım: özellikler birbirinden bağımsızdır
  - $P(y|x_1, x_2, x_3, \dots, x_n) = P(y|x_1)P(y|x_2) \dots P(y|x_n)$
- Sklearn:
  - GaussianNB

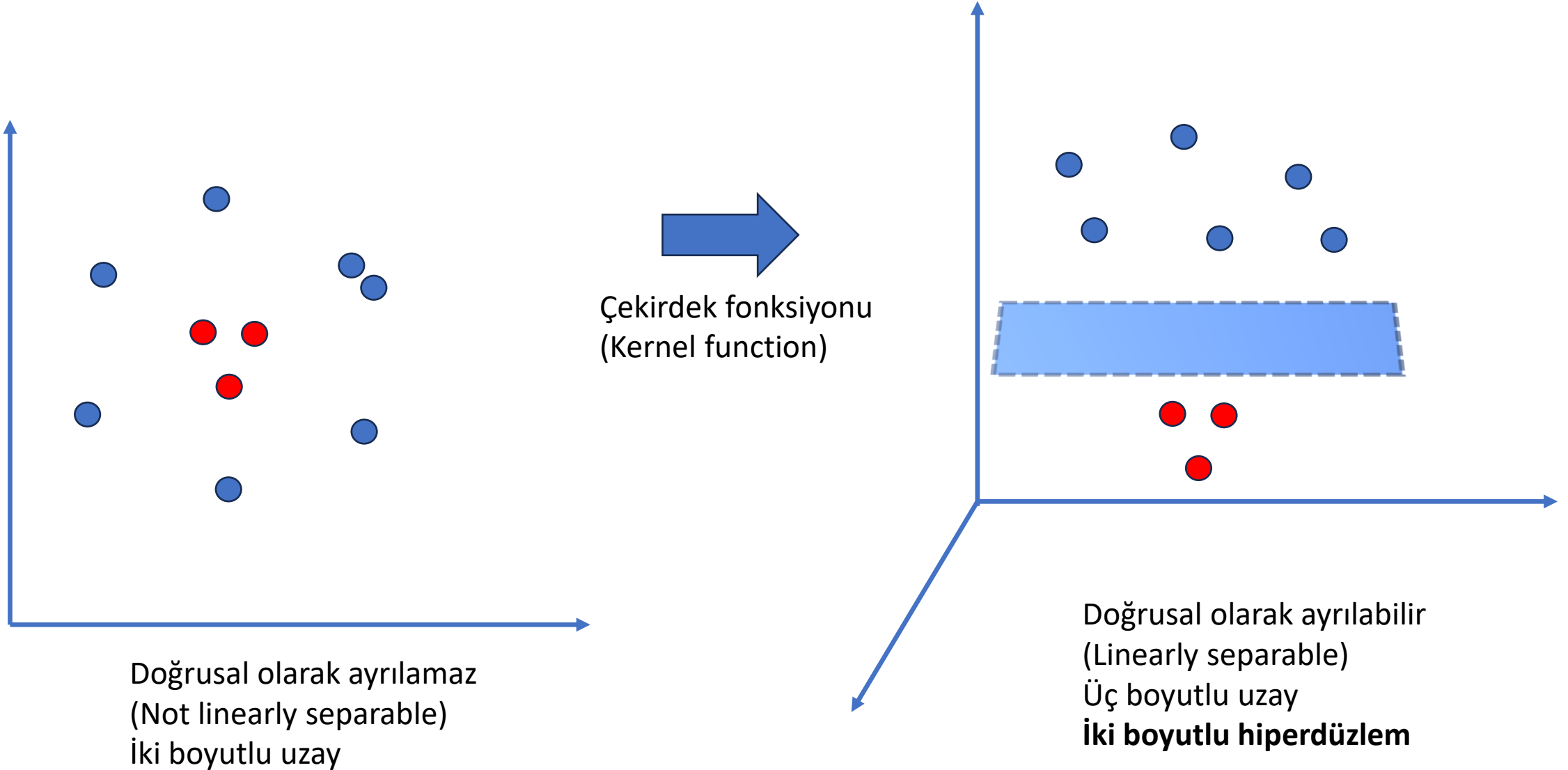
# Support Vector Machine (SVM)

- SVM, hedefin bir hiperdüzlem (hyperplane) kullanarak veri noktalarını iki sınıfa ayırmak olduğu ikili sınıflandırma görevleri için icat edilmiştir.
- **SVM, hiperdüzlem ile her sınıfın en yakın veri noktaları arasındaki mesafe olan maksimum uzaklığa (maximum margin) hesaplar**
- Kernel Trick: Bir çekirdek fonksiyonu (kernel function) kullanarak girdi özelliklerini dönüştürerek doğrusal olmayan karar sınırlarını idare edebilir
  - polynomial, radial basis function (RBF), sigmoid, vb.
- Sklearn:
  - SVC (Support Vector Classifier)
  - SVR (Support Vector Regressor)

# SVM (Görsel)



# SVM Kernel Trick

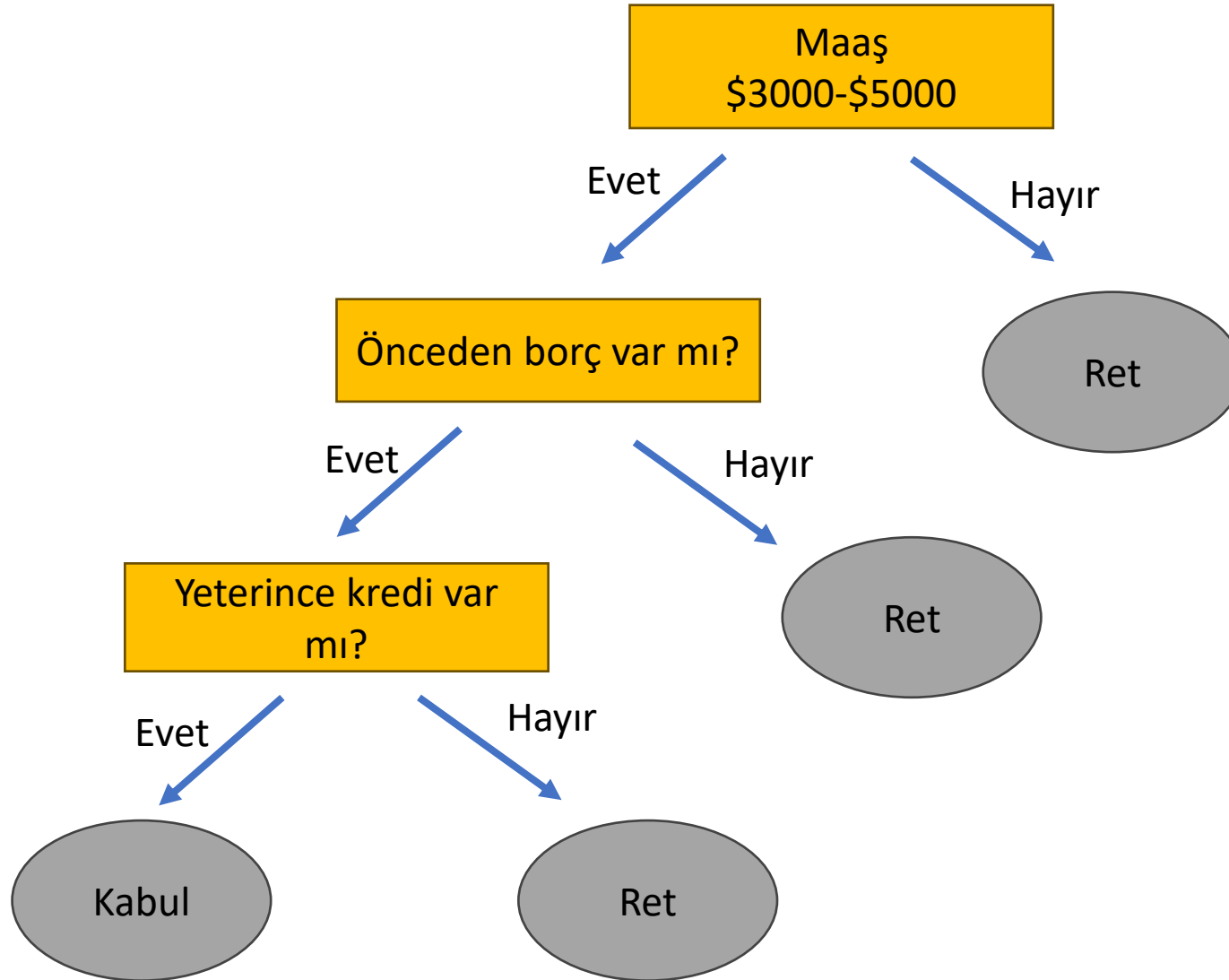




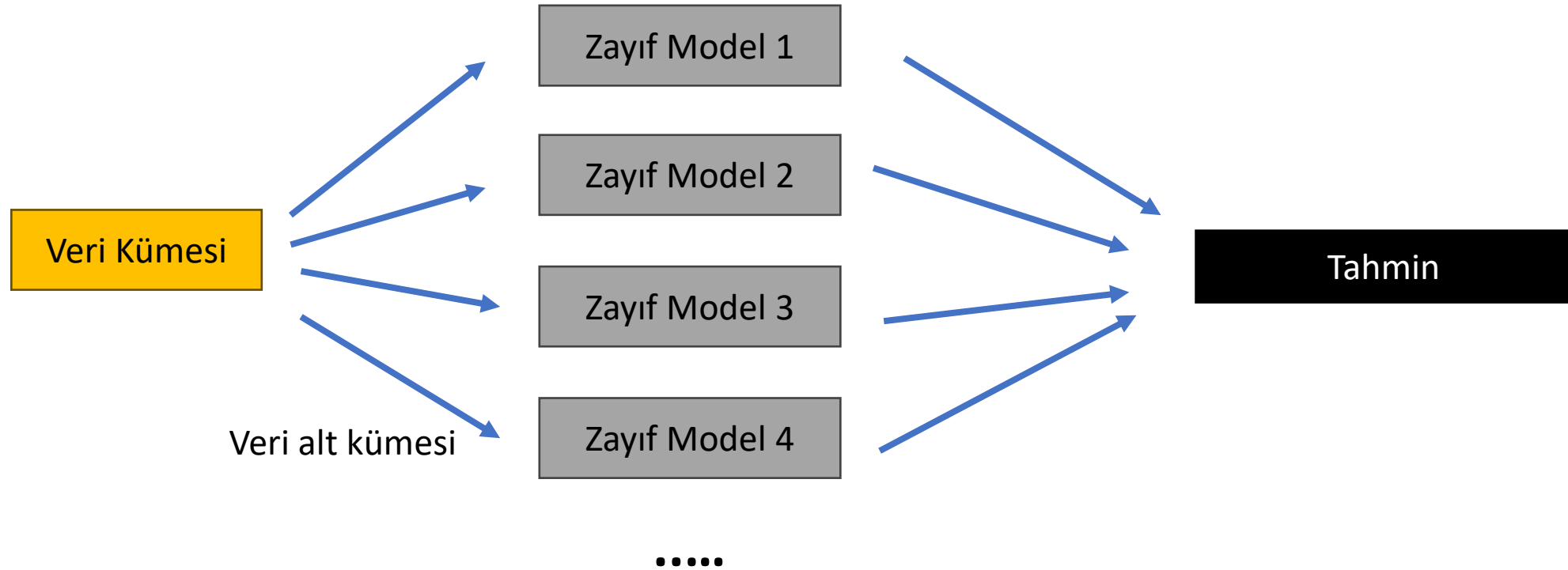
# Decision Trees

- Karar Ağaçları (Decision Trees), düğümlerin (nodes) kararları veya test koşullarını temsil ettiği ve dalların olası sonuçları temsil ettiği hiyerarşik yapıları içerir.
- Her düğümde veriyi bölmek için en iyi özellik ve eşik değerini belirlemek için bir bölme kriteri kullanır.
  - Gini impurity (sınıflandırma)
  - mean squared error (regresyon)
- Sklearn:
  - DecisionTreeClassifier
  - DecisionTreeRegressor

# Decision Trees (Görsel)

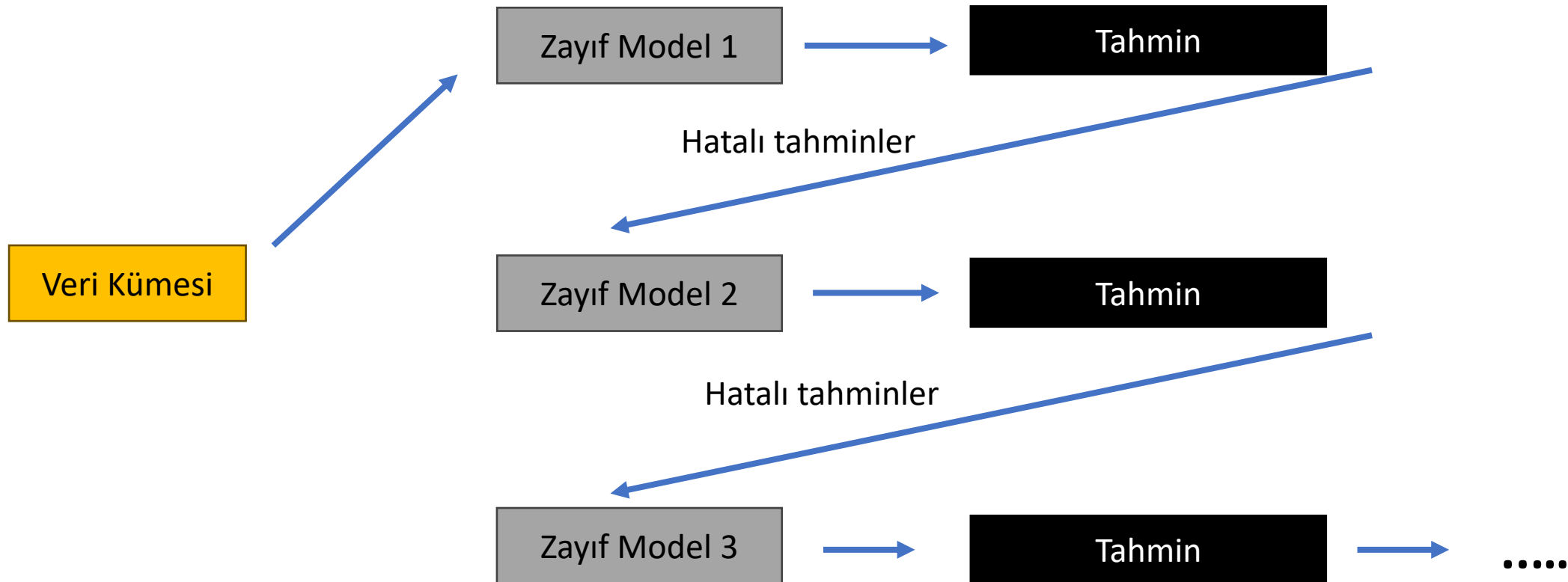


# Ensemble Modelleri



**Bagging (Bootstrap Aggregating)**

# Ensemble Modelleri

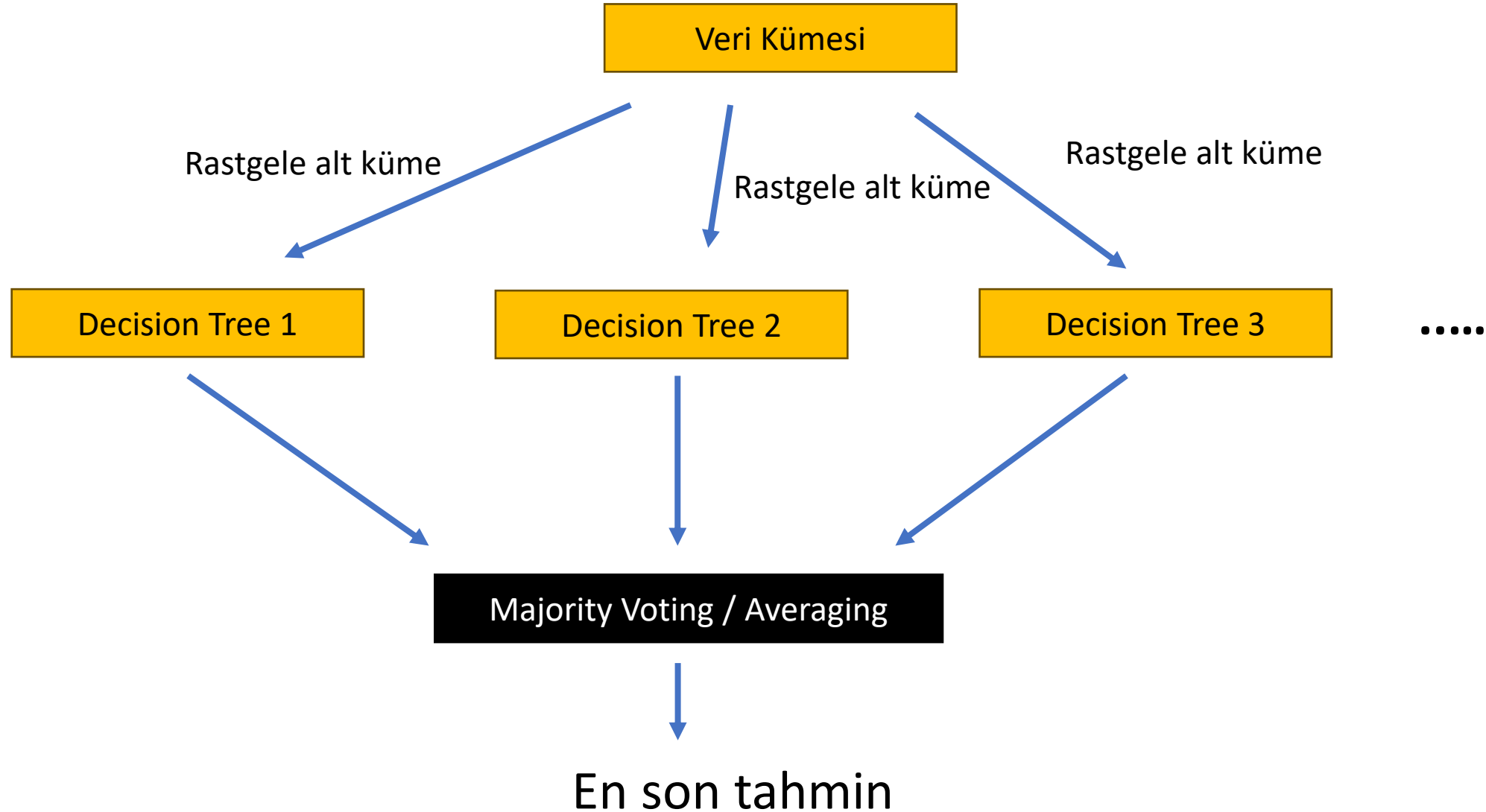


**Boosting**

# Random Forest

- Rastgele Orman (Random Forest), birden fazla karar ağacı (decision tree) oluşturan bir ensemble öğrenme tekniğidir
- Her karar ağacı, bootstrap örnekleme kullanılarak verinin rastgele bir alt kümesi üzerinde bağımsız olarak eğitilir (bagging ensemble model)
- feature importance ile özellik seçimi (feature selection) yapabilir
- Sklearn:
  - RandomForestClassifier
  - RandomForestRegressor

# Random Forest



# İkili Sınıflandırma (Binary Classification)

- İkili sınıflandırmada, etiketler 0 ve 1'dir. Gerçek değerler veri kümesinden gelirken, tahmin değerleri modelden gelir.
- **TP (true positive)**: Gerçek sınıf 1 ve biz 1 olarak tahmin ediyoruz
- **TN (true negative)**: Gerçek sınıf 0 ve biz 0 olarak tahmin ediyoruz
- **FP (false positive)**: Gerçek sınıf 0 fakat biz 1 olarak tahmin ediyoruz
  - Type I Error
- **FN (false negative)**: Gerçek sınıf 1 fakat biz 0 olarak tahmin ediyoruz
  - Type II Error

# Sınıflandırma Metrikleri

- $precision = \frac{TP}{TP + FP}$

Positive Predictive Value (PPV)

- $recall = sensitivity = \frac{TP}{TP + FN}$

True Positive Rate (TPR)

- $F1\ score = 2 * \frac{precision * recall}{precision + recall}$

Harmonic Mean of PPV and TPR

- $specificity = \frac{TN}{TN + FP}$

True Negative Rate (TNR)

- $accuracy = \frac{TP + TN}{TP + TN + FP + FN}$



# Karmaşıklık Matrisi (Confusion Matrix)

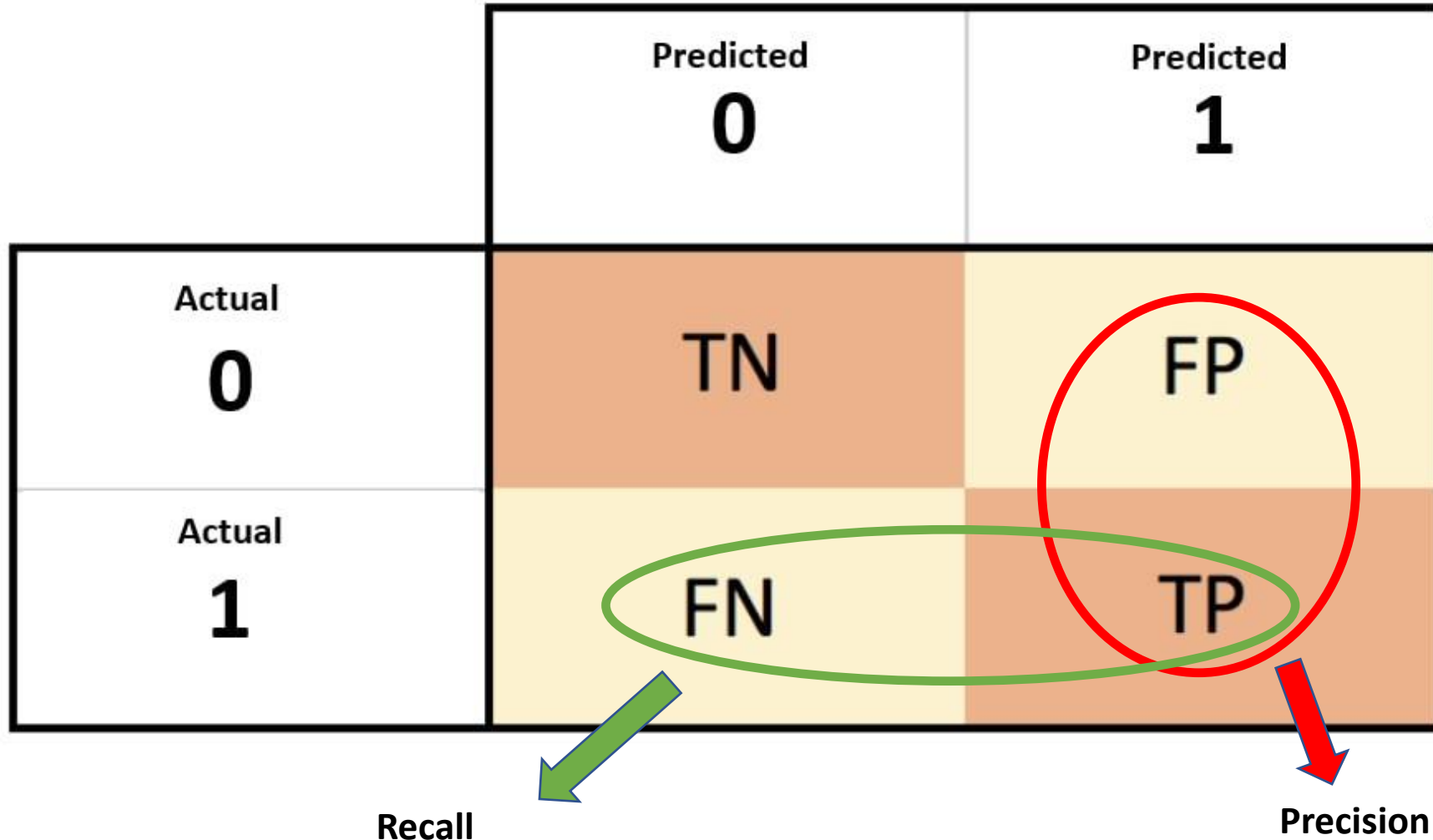
		Predicted <b>0</b>	Predicted <b>1</b>
Actual <b>0</b>	TN	FP	
Actual <b>1</b>	FN	TP	

# Karmaşıklık Matrisi (Confusion Matrix)

		Predicted <b>0</b>	Predicted <b>1</b>
Actual <b>0</b>	TN	FP	
Actual <b>1</b>	FN	TP	

Recall

Precision



# Veri Bilimi ve Makine Öğrenmesi için Genel İş Akışı

- Veri topla/oluştur
- Veriyi oku ve görselleştir (EDA - Keşifsel Veri Analizi)
- Veriyi ön işle (ölçeklendirme, eksik/dengesiz veri düzenleme vb.)
- Özellikleri seç (boyut indirgeme vb.)
- Modeli seç (Doğrusal regresyon, Rastgele Orman, SVM vb.)
- Modelin için en iyi hiperparametreleri seç (Grid search vb.)
- Görevin için değerlendirme metriklerini seç (F1 vb.)
- Eğitim verileri ile modeli eğit (fit), test verilerinde tahmin yap
- Sonuçları değerlendir ve yorum yap