

K-En Yakın Komşu (k-NN), makine öğrenimi alanında sınıflandırma ve regresyon problemlerini çözmek için kullanılan bir algoritmadır. k-NN algoritması, veri noktalarının sınıflandırılması veya bir tahmin yapılması için komşularının etiketlerine dayanır.

Temel fikir şudur: Bir veri noktasının sınıfını tahmin etmek için, bu noktaya en yakın k adet komşusunun etiketlerine bakılır. Sınıflandırma için, bu k komşunun en yaygın olan sınıfı tahmin edilir ve veri noktası bu sınıfa atanır. Regresyon için, bu k komşunun ortalama değeri tahmin edilir ve bu değer veri noktasına atanır.

k-NN algoritması için belirleyici faktörlerden biri "k" parametresidir. Bu parametre, her bir tahmin için kaç komşunun kullanılacağını belirler. k değeri, modelin performansını ve karar sınırlarının pürüzsüzlüğünü etkileyebilir. Genellikle k için tek veya çift sayılar seçilir. Eğer $k=1$ ise, en yakın komşu yöntemi uygulanır.

k-NN algoritması şu adımları izler:

- 1- Her veri noktasının diğer tüm veri noktalarıyla olan mesafesi hesaplanır. Mesafe ölçümü genellikle Öklid mesafesi kullanılır, ancak farklı mesafe metrikleri de kullanılabilir.
- 2- Mesafeler sıralanır ve en yakın k komşu belirlenir.
- 3- Sınıflandırma için, en yakın k komşunun sınıflarının sayılması ve çoğunluğun sınıfı tahmin olarak atanır. Regresyon için, en yakın k komşunun değerlerinin ortalaması tahmin olarak atanır.
- 4- Tahmin edilen sınıf veya değer veri noktasına atanır.

k-NN algoritması basit ve anlaşılır bir yapıya sahiptir, bu nedenle sınıflandırma ve regresyon problemlerinde yaygın olarak kullanılır. Ancak, büyük veri setlerinde hesaplama maliyeti yüksek olabilir ve etkili sonuçlar elde etmek için veri setinin ölçeklenmesi ve özellik seçiminin dikkatlice yapılması gerekebilir.

K-En yakın komşu kodu :

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# Veri setini yükleme
iris = load_iris()
X = iris.data
y = iris.target

# Veriyi eğitim ve test setlerine ayırma
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Veri ön işleme: özellik ölçeklendirme
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# k-NN modelini oluşturma ve eğitme
classifier = KNeighborsClassifier(n_neighbors=5) # k = 5 olarak
ayarlanabilir
```

```
classifier.fit(X_train, y_train)
```

```
# Test seti üzerinde tahmin yapma
```

```
y_pred = classifier.predict(X_test)
```

```
# Doğruluk skorunu hesaplama
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
print("Doğruluk:", accuracy)
```