

Destek Vektör Makineleri (Support Vector Machines - SVM), makine öğrenimi alanında sınıflandırma ve regresyon problemlerini çözmek için kullanılan güçlü bir algoritmadır. SVM, bir hiperdüzlem (hyperplane) ile veri noktalarını en iyi şekilde ayırarak sınıflandırma yapar veya regresyon tahminleri yapar.

SVM'nin temel amacı, veri noktalarını bir hiperdüzlem ile sınıflar arasında en geniş bir boşluk bırakacak şekilde bölmektir. Bu hiperdüzlem, sınıflar arasındaki marjinal mesafeyi (margin) maksimize eder. SVM, sınıflandırma problemlerinde lineer ve non-lineer olarak kullanılabilir.

SVM algoritması şu adımları izler:

- 1- Veri Ön İşleme: Öncelikle, veri seti üzerinde ön işleme yapılır, özellikler normalize edilir veya ölçeklendirilir.
- 2- Sınıflandırma için Hiperdüzlem Bulma: SVM, veri noktalarını sınıflandırmak için bir hiperdüzlem bulmaya çalışır. Eğer veri seti lineer olarak ayrılabilirse, SVM doğrusal bir hiperdüzlem bulmaya çalışır. Eğer veri seti doğrusal olarak ayrılamazsa, SVM çekirdek fonksiyonları kullanarak veriyi yüksek boyutta bir uzayda dönüştürür ve orada lineer olarak ayrılacak bir hiperdüzlem bulmaya çalışır.
- 3- Marjinin Maksimizasyonu: SVM, hiperdüzlem ile her bir sınıfa en yakın olan veri noktaları arasındaki mesafeyi (marjin) maksimize etmeye çalışır. Bu, sınıflar arasında en iyi ayrımı sağlar.

- 4- Destek Vektörlerin Belirlenmesi: Marjinal mesafenin maksimize edilmesi için kullanılan veri noktalarına "destek vektörler" denir. Bu noktalar, hiperdüzlem ile en yakın olan noktalardır.
- 5- Sınıflandırma veya Regresyon Tahmini: Model eğitildikten sonra, yeni veri noktaları sınıflandırmak veya tahmin yapmak için kullanılabilir.

SVM'nin avantajları arasında etkili sınıflandırma performansı, çoklu boyutta kullanılabilme yeteneği, hiperparametrelerin esnek ayarlanabilirliği ve overfitting'e karşı direnç göstermesi bulunur. Ancak, SVM'nin dezavantajları arasında hesaplama maliyetinin yüksek olması ve parametrelerin ayarlanması gerekliliği sayılabilir.

Destek Vektör Makineleri Algoritma python kodu :

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
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)

# SVM modelini oluşturma ve eğitme
classifier = SVC(kernel='linear', random_state=42) # Lineer çekirdek
kullanılıyor
```

```
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)
```