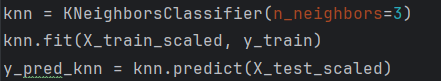
**KNN Algoritması**

KNN, gözetimli öğrenme algoritmalarından biridir ve sınıflandırma problemlerinde sıkça kullanılır. Temelde, yeni bir veri noktasını sınıflandırırken, bu noktanın en yakın "k" komşusunun etiketlerine bakarak bir tahminde bulunur. Komşular arasındaki mesafe genellikle Euclidean mesafesi ile hesaplanır.

* Veri setindeki her bir örnek bir nokta olarak kabul edilir.
* Yeni bir örnek sınıflandırılacaksa, bu örneğin etrafındaki "k" en yakın komşu noktalar bulunur.
* Bu komşuların hangi sınıftan olduğu belirlenir ve en fazla görülen sınıf, yeni örneğe atanır.

Kodda Örneği:  
****

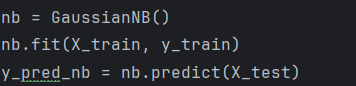
KNN modeli, KNeighborsClassifier sınıfı ile tanımlanır. Bu sınıf, eğitim verisini alır ve ardından test verisi üzerinde tahmin yapar. Burada kullanılan parametre, n\_neighbors=3 ile, her tahmin için 3 en yakın komşu dikkate alınır.

**Naive Bayes Algoritması**

Naive Bayes, olasılık temelli bir sınıflandırma algoritmasıdır. Her özellik arasındaki bağımsızlık varsayımına dayanır (yani her özellik birbiriyle bağımsızdır). Bu nedenle "naive" (naif) olarak adlandırılır. Genelde metin sınıflandırma gibi büyük veri setlerinde oldukça etkilidir.

* Naive Bayes, Bayes teoremi kullanarak her sınıfın olasılığını hesaplar.
* Eğitim setinden elde edilen her bir özelliğin sınıfla ilişkisi modelde kullanılır. Sonrasında, her sınıf için en yüksek olasılık hesaplanarak tahmin yapılır.

Kodda Örneği:

****

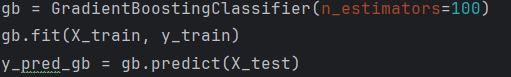
Naive Bayes, GaussianNB sınıfı ile uygulanır. Bu sınıf, verilerin Gauss (normal) dağılımına uygun olduğu varsayımıyla çalışır. Eğitim verisi üzerinden model eğitilir ve ardından test verisi ile tahmin yapılır.

**Gradient Boosting Algoritması**

Gradient Boosting, bir tür topluluk öğrenme algoritmasıdır. Bu algoritma, zayıf öğrenicilerden (genellikle karar ağaçları) oluşan bir model ailesi yaratır. Temel fikir, önceki modelin hatalarını düzeltmek için yeni modellerin oluşturulmasıdır. Bu, modelin her seferinde hataları minimize etmeye çalıştığı bir süreçtir.

* Gradient Boosting, bir dizi basit model (karar ağaçları) kurar ve her yeni model, önceki modelin hatalarını düzeltmeye çalışır.
* Her yeni modelin ağırlığı, modelin önceki hatalarına göre belirlenir.

Kodda Örneği:



GradientBoostingClassifier, bu algoritmanın Scikit-learn'deki karşılığıdır. Burada, model 100 ağaç kullanılarak eğitilmiştir. Model eğitim verisini öğrenir ve test verisi üzerinde tahmin yapar.

**Grid Search**

Grid Search, bir modelin hiperparametrelerini en iyi şekilde ayarlamak için kullanılan bir tekniktir. Bu yöntem, belirli bir hiperparametre aralığı içinde tüm kombinasyonları dener ve her birini değerlendirir. Bu şekilde, en iyi parametrelerin bulunduğu model seçilir.

* Kullanıcı, hangi hiperparametrelerin optimize edileceğini ve hangi değer aralıklarında tarama yapılacağını belirler.
* Grid Search, bu parametrelerin tüm olasılıklarını dener ve en iyi sonucu veren parametreyi seçer.

**Cross Validation**

Cross-validation, modelin performansını değerlendirmek için kullanılan bir tekniktir. Veriler, eğitim ve test kümelerine birkaç kez farklı şekilde bölünür ve model her bir bölmede eğitim alıp test edilir. Bu, modelin genelleme yeteneğini daha doğru bir şekilde ölçmeyi sağlar.

* Veri seti, genellikle "k" katmana (fold) ayrılır.
* Her katman, sırayla test seti olarak kullanılırken diğer katmanlar eğitim seti olarak kullanılır.
* Sonuçlar, her katmandan elde edilen doğrulukların ortalaması alınarak değerlendirilir.