

Naive bayes

Naive Bayes

- Bayes Ağı, bir dizi değişken arasındaki ilişkilerin olasılıklarını öğrenerek çıkarım yapan bir makine öğrenmesi algoritmasıdır. Düşünce ya da oluşan kanı veya yönlendirilmiş olasılıklı bir modeldir. Örneğin, bir Bayes ağı, hastalıklar ve semptomlar arasındaki olasılık ilişkilerini temsil edebilir. Belirtiler verildiğinde, ağ çeşitli hastalıkların varlığının olasılıklarını hesaplamak için kullanılabilir.

Naive Bayes

- Konuşma sinyalleri veya protein dizileri gibi değişken dizilerini modelleyen Bayes ağlarına dinamik Bayes şebekeleri denir. Belirsizlik altında karar problemlerini temsil edebilen ve çözebilen Bayes ağlarının genellemelerine etki diyagramları denir.
- Naïve Bayes'te sınıflandırıcı, lojistik regresyon gibi ayırt edici modellere göre daha hızlı birleşir, bu nedenle daha az eğitim verisine ihtiyacınız vardır. Ana avantajı, özellikler arasındaki etkileşimleri öğrenememesidir.

Naive Bayes

- Bayesci mantık programı iki bileşenden oluşur. İlk bileşen mantıklı bir bileşendir; alanın niteliksel yapısını yakalayan bir dizi Bayes Cümlelerinden oluşur. İkinci bileşen nicelikselidir, alanla ilgili nicel bilgileri kodlar.
- Yağmur, yağmurlama sisteminin etkinleştirilip etkinleştirilmeyeceğini etkiler ve hem yağmur hem de yağmurlama sistemi çimlerin ıslak olup olmadığını etkiler. Çim ıslak ise yağmu mu yağdı yoksa yağmurlama sistemi mi çalıştı.

Naive Bayes Nasıl Çalışır?

- Bayes Teoremi

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

Bayes Teoremi

- $P(A|B)$ = B'nin doğru olduğu bilindiğinde A'nın olma olasılığı
- $P(B|A)$ = A'nın doğru olduğu bilindiğinde B'nin olma olasılığı
- $P(A)$ = A'nın olma olasılığı
- $P(B)$ = B'nin olma olasılığı
- Naive Bayes 18.yy'da Thomas Bayes'in **Bayes Teoremi** temel alarak geliştirilmiştir. Hemen bir örnekle açıklayalım.

Naive bayes örnek

- Aşağıda hava durumu(X) ve ona karşılık gelen kategorik oyunu oynama durumları(y) yer alıyor.
- Öncelikle verilerimizi frekans tablosuna çeviriyoruz ve olasılıkları hesaplıyoruz. Sonrasında teoremi uygulamaya başlıyoruz.

Hava	Oynama D.
Güneşli	Hayır
Bulutlu	Evet
Yağmurlu	Evet
Güneşli	Evet
Güneşli	Evet
Bulutlu	Evet
Yağmurlu	Hayır
Yağmurlu	Hayır
Güneşli	Evet
Yağmurlu	Evet
Güneşli	Hayır
Bulutlu	Evet
Bulutlu	Evet
Yağmurlu	Hayır

Sıklık Tablosu		
Hava	Hayır	Evet
Bulutlu	0	4
Yağmurlu	3	2
Güneşli	2	3
Toplam	5	9

Olasılık Tablosu		
Hava	Hayır	Evet
Bulutlu	0	4
Yağmurlu	3	2
Güneşli	2	3
Hepsi	5	9
	5 / 14	9 / 14
	0.36	0.64

4 / 14 0.29
5 / 14 0.36
5 / 14 0.36

- Örneğin sorumuz şu olsun:
- Hava yağmurlu olduğunda oyun oynar mıyım?
- $P(\text{Evet} \mid \text{Yağmurlu}) = P(\text{Yağmurlu} \mid \text{Evet}) * P(\text{Evet}) / P(\text{Yağmurlu})$
- $P(\text{Yağmurlu} \mid \text{Evet}) = 2 / 9 = 0.22$
- $P(\text{Evet}) = 9/14 = 0.64$
- $P(\text{Yağmurlu}) = 5/14 = 0.36$

Hava	Oynama D.
Güneşli	Hayır
Bulutlu	Evet
Yağmurlu	Evet
Güneşli	Evet
Güneşli	Evet
Bulutlu	Evet
Yağmurlu	Hayır
Yağmurlu	Hayır
Güneşli	Evet
Yağmurlu	Evet
Güneşli	Hayır
Bulutlu	Evet
Bulutlu	Evet
Yağmurlu	Hayır

Sıklık Tablosu		
Hava	Hayır	Evet
Bulutlu	0	4
Yağmurlu	3	2
Güneşli	2	3
Toplam	5	9

$$P(\text{Yağmurlu} \mid \text{Evet}) = 2/9 = 0.22$$

Olasılık Tablosu				
Hava	Hayır	Evet		
Bulutlu	0	4	4 / 14	0.29
Yağmurlu	3	2	5 / 14	0.36
Güneşli	2	3	5 / 14	0.36
Hepsi	5	9		
	5 / 14	9 / 14		
	0.36	0.64		

$$P(\text{Yağmurlu}) = 5 / 14 = 0.36$$

$$P(\text{Evet}) = 9 / 14 = 0.64$$

- Şimdi, $P(\text{Evet} \mid \text{Yağmurlu}) = 0.22 * 0.64 / 0.36 = 0.39$ olarak çıktı.
- Burada tek sütunlu bir özellik(Hava durumu) kullandık. Eğer birden fazla sütunumuz olsaydı, bu işlemlerin aynısını her sütun için yapacaktık. Yani Naive Bayes'te her özellik(X_1, X_2, \dots, X_n) birbirinden bağımsız olarak değerlendirilecekti.

$$P(y|x_1, \dots, x_n) = \frac{P(x_1|y)P(x_2|y)\dots P(x_n|y)P(y)}{P(x_1)P(x_2)\dots P(x_n)}$$

File Edit Selection View Go Run Terminal Help

EXPLORER

PHYTON

denetimli ogrenme.py
denetimsiz öğrenme.py
hiyerarşik kümeleme.py
k-means.py
karar ağaçları.py
KNN.py
naive bayes.py
pekiştirmeli öğrenme maze örneği...
phyton denemeleri.py

naive bayes.py > ...
22 nb = GaussianNB()
23 nb.fit(X_train, y_train)
24
25 # Test verisi ile tahmin yapma
26 y_pred = nb.predict(X_test)
27
28 # Sonuçları değerlendirme
29 print(f'Accuracy: {accuracy_score(y_test, y_pred)}')
30 print('Confusion Matrix:')
31 print(confusion_matrix(y_test, y_pred))
32 print('Classification Report:')
33 print(classification_report(y_test, y_pred))
34
35 # Sınıflandırma sonuçlarını görselleştirme (ilk
36 plt.figure(figsize=(10, 6))
37 plt.scatter(X_test[:, 0], X_test[:, 1], c=y_pred)
38 plt.title('Naive Bayes Sınıflandırma Sonuçları')
39 plt.xlabel(iris.feature_names[0])
40 plt.ylabel(iris.feature_names[1])
41 plt.show()
42

TERMINAL

Classification Report:
precision recall f1-score support
0 1.00 1.00 1.00 10
1 1.00 1.00 1.00 9
2 1.00 1.00 1.00 11
accuracy 1.00 30
macro avg 1.00 1.00 1.00 30
weighted avg 1.00 1.00 1.00 30

OUTLINE

