Naive Bayes Sınıflandırıcı

Atil Samancioglu

1 Naive Bayes Sınıflandırıcı

Naive Bayes, olasılık temelli bir makine öğrenimi algoritmasıdır ve hem ikili hem de çok sınıflı sınıflandırma problemlerinde yaygın olarak kullanılır. Bu algoritmanın temeli, istatistikte önemli bir yer tutan **Bayes Teoremi**'ne ve özelliklerin birbirinden *bağımsız olduğu* varsayımına dayanır.

1.1 Temel Olasılık Bilgisi

Olasılık teorisinde olaylar ikiye ayrılır:

- **Bağımsız Olaylar:** Bir olayın gerçekleşme olasılığı, başka bir olayın gerçekleşip gerçekleşmediğinden etkilenmez. Örneğin, zar atıldığında 1 gelme olasılığı her zaman $\frac{1}{6}$'dır.
- **Bağımlı Olaylar:** Bir olayın gerçekleşme olasılığı, başka bir olayın gerçekleşip gerçekleşmediğine bağlıdır. Örneğin, bir torbadan bilye çektikten sonra yerine koymadan bir tane daha çekmek.

1.2 Koşullu Olasılık

Bir olayın, başka bir olayın gerçekleştiği bilgisi altında gerçekleşme olasılığıdır.

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

Bu formül, "B olayı gerçekleştiğinde A olayının olasılığı nedir?" sorusunu yanıtlar.

1.3 Bayes Teoremi

Bayes Teoremi, koşullu olasılıkların tersini hesaplamak için kullanılır:

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

Bu formül, Naive Bayes algoritmasının temelidir. Burada:

- $P(A|B)$: Gözlem B verildiğinde, A'nın olasılığı (posterior)

- $P(B|A)$: A verildiğinde B'nin olasılığı (likelihood)
- $P(A)$: A'nın önceden bilinen olasılığı (prior)
- $P(B)$: Gözlemlerin genel olasılığı (evidence)

1.4 Naive Bayes Algoritması

Makine öğrenimi bağlamında, Bayes Teoremi şu şekilde yeniden yazılır:

$$P(y|x_1, x_2, \ldots, x_n) = \frac{P(y) \cdot P(x_1|y) \cdot P(x_2|y) \cdots P(x_n|y)}{P(x_1, x_2, \ldots, x_n)}$$

Naive Bayes'in "naive" olarak adlandırılmasının nedeni, **özelliklerin birbirinden bağımsız** olduğu varsayımıdır. Yani:

$$P(x_1, x_2, \ldots, x_n|y) = P(x_1|y) \cdot P(x_2|y) \cdots P(x_n|y)$$

1.5 Sınıf Tahmini

Sınıflandırma işlemi için şu hesaplama yapılır:

$$\text{class}_{\text{predict}} = \arg\max_y \left[P(y) \prod_{i=1}^{n} P(x_i|y) \right]$$

Buradaki hedef, hangi sınıfın (örneğin "evet" veya "hayır") en yüksek olasılığa sahip olduğunu belirlemektir.

1.6 Uygulamalı Örnek: E-posta Sınıflandırma (Spam / Ham)

Bu örnekte basit bir e-posta sınıflandırma problemi ele alınacaktır. Amaç, gelen bir e-postanın **spam** mi yoksa **normal (ham)** mı olduğunu belirlemektir. Veri kümesinde yalnızca iki özellik (feature) bulunmaktadır: `Free` ve `Win`, yani bu kelimelerin e-posta metninde bulunup bulunmadığı.

Veri Kümesi

Free	Win	Label (Sınıf)
Yes	Yes	Spam
Yes	No	Spam
No	Yes	Ham
Yes	Yes	Spam
No	No	Ham

Amaç: `Free = Yes`, `Win = No` olan yeni bir e-postanın sınıfını tahmin etmek.

Adım 1: Sınıf Olasılıklarını Hesapla (Prior)

Toplam 5 e-posta örneği vardır. Bunlardan 3'ü `Spam`, 2'si `Ham`:

$$P(\text{Spam}) = \frac{3}{5}, \quad P(\text{Ham}) = \frac{2}{5}$$

Adım 2: Koşullu Olasılıkları Hesapla (Likelihood)

Verilen sınıf altında, her özelliğin değerine göre olasılık:

- $P(\text{Free=Yes}|\text{Spam}) = \frac{3}{3} = 1.0$
- $P(\text{Free=Yes}|\text{Ham}) = \frac{0}{2} = 0.0$
- $P(\text{Win=No}|\text{Ham}) = \frac{1}{2}$

Adım 3: Tahmin Edilecek Olasılıkları Hesapla

Yeni bir e-posta geldi: `Free = Yes, Win = No`

Spam olma olasılığı:

$$P(\text{Spam}|\text{Free=Yes}, \text{Win=No}) \propto P(\text{Spam}) \cdot P(\text{Free=Yes}|\text{Spam}) \cdot P(\text{Win=No}|\text{Spam})$$

$$= \frac{3}{5} \cdot 1.0 \cdot \frac{1}{3} = \frac{1}{5}$$

Ham olma olasılığı:

$$P(\text{Ham}|\text{Free=Yes}, \text{Win=No}) \propto \frac{2}{5} \cdot 0.0 \cdot \frac{1}{2} = 0$$

Sonuç:

$$P(\text{Spam}|\text{Free=Yes}, \text{Win=No}) > P(\text{Ham}|\text{Free=Yes}, \text{Win=No})$$

Bu nedenle, bu e-posta **Spam** olarak sınıflandırılır.

Not: Sıfır Olasılık Sorunu ve Laplace Düzeltmesi

Yukarıdaki örnekte, $P(\text{Free=Yes}|\text{Ham}) = 0$ olduğu için ilgili olasılık çarpımı sıfırlandı. Bu tür durumlarda **Laplace düzeltmesi** uygulanarak her kategoriye küçük bir sabit (genellikle 1) eklenir.

Laplace düzeltmesi uygulandığında:

$$P(\text{Free=Yes}|\text{Ham}) = \frac{0+1}{2+2} = \frac{1}{4}$$

Tüm olasılık hesapları tekrar yapılabilir.

1.7 Naive Bayes Varyantları: Bernoulli, Multinomial ve Gaussian

Naive Bayes algoritmasının farklı veri türleri için kullanılan üç ana varyantı vardır. Bunlar:

1. Bernoulli Naive Bayes
2. Multinomial Naive Bayes
3. Gaussian Naive Bayes

Bu üç model, Bayes Teoremi'ne dayanır, ancak veri türüne göre farklı dağılım varsayımları yaparlar. Hangi modelin ne zaman kullanılması gerektiğini anlamak, modelin başarısı açısından kritik öneme sahiptir.

Bernoulli Naive Bayes

Bernoulli Naive Bayes, bağımsız değişkenlerin (*features*) yalnızca iki farklı değer alabildiği durumlar için uygundur. Bu iki değer genellikle `0-1`, `Evet-Hayır`, `Var-Yok`, `Başarılı-Başarısız` gibi ikili (binary) durumlardır. Özellikler Bernoulli dağılımını izler.

- **Veri Türü:** Binary (0/1)
- **Kullanım Alanı:** Tıklanma tahmini, e-posta açılıp açılmama, kullanıcı davranışları (örneğin bir özelliği kullanmış mı?)

Örnek:

Free Offer	Clicked	Purchased
Yes	No	No
Yes	Yes	Yes
No	No	Yes
Yes	No	No
No	Yes	Yes

Yukarıdaki örnekte her bir özellik yalnızca `Yes/No` (1/0) biçimindedir, bu nedenle Bernoulli Naive Bayes kullanılabilir.

Multinomial Naive Bayes

Multinomial Naive Bayes özellikle **metin verisi** (text data) ile çalışmak için uygundur. Bu model, bir özelliğin (örneğin bir kelimenin) bir örnekte kaç defa geçtiğine (frekansına) göre çalışır. Bu nedenle, en çok kullanılan doğal dil işleme (NLP) görevlerinde tercih edilir.

- **Veri Türü:** Sayma (count-based), kelime frekansı

- **Kullanım Alanı:** E-posta spam tespiti, haber kategorilendirme, duygu analizi

Örnek:
Aşağıdaki e-posta içeriklerinin **spam** olup olmadığını belirlemek istiyoruz:

- E-posta 1: "Win $1000 now!" → Spam
- E-posta 2: "Hi Atil, how was meeting yesterday?" → Ham

Özellik Matrisi (Bag of Words):

	win	now	meeting	yesterday
Spam	1	1	0	0
Ham	0	0	1	1

Burada kelimelerin sayıları özellik olarak modele verilir. Multinomial Naive Bayes, bu frekanslara göre sınıf olasılıklarını hesaplar.

Gaussian Naive Bayes

Gaussian Naive Bayes, özelliklerin sürekli sayısal değerler aldığı durumlarda kullanılır. Eğer veriniz ortalama etrafında simetrik bir dağılım (çan eğrisi – normal dağılım) gösteriyorsa bu algoritma uygundur. Özelliklerin Gaussian (normal) dağılım gösterdiği varsayılır.

- **Veri Türü:** Sürekli (continuous)
- **Kullanım Alanı:** Yaş, boy, kilo gibi fiziksel ölçümlerle yapılan sınıflandırmalar, medikal veriler

Örnek:

Age	**Weight**	**Height**	**Obese?**
25	80	170	Yes
30	90	165	Yes
22	55	178	No
28	60	180	No

Bu verideki yaş, kilo ve boy değişkenleri sürekli (continuous) olduğundan Gaussian Naive Bayes kullanılabilir. Her bir özellik için, sınıfa göre ayrı ayrı ortalama (μ) ve standart sapma (σ) hesaplanır. Olasılıklar aşağıdaki normal dağılım fonksiyonu ile hesaplanır:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$$

Gaussian Naive Bayes: Adım Adım Örnek

Elimizde şu veri seti var:

Age	Weight	Height	Obese?
25	80	170	Yes
30	90	165	Yes
22	55	178	No
28	60	180	No

Yeni bir kişi için tahmin yapmak istiyoruz:

- Age: 27
- Weight: 85
- Height: 175

Adım 1: Ortalama (μ) ve Standart Sapma (σ) Hesapla

- **Obese = Yes**
  - Age: $\mu = 27.5$, $\sigma = 2.5$
  - Weight: $\mu = 85$, $\sigma = 5$
  - Height: $\mu = 167.5$, $\sigma = 2.5$
- **Obese = No**
  - Age: $\mu = 25$, $\sigma = 3$
  - Weight: $\mu = 57.5$, $\sigma = 2.5$
  - Height: $\mu = 179$, $\sigma = 1$

Adım 2: Gaussian Olasılıklarını Hesapla Normal dağılım formülü:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right)$$

Obese = Yes için:

- Age = 27:

$$P = \frac{1}{\sqrt{2\pi(2.5^2)}} \exp\left(-\frac{(27 - 27.5)^2}{2(2.5^2)}\right) \approx 0.156$$

- Weight = 85:

$$P \approx 0.0798$$

- Height = 175:

$$P \approx 0.0017$$

Obese = No için:

- Age = 27:
$$P \approx 0.133$$
- Weight = 85:
$$P \approx 0 \quad \text{(çok düşük)}$$
- Height = 175:
$$P \approx 0.0001$$

Adım 3: Olasılıkları Çarp ve Karşılaştır

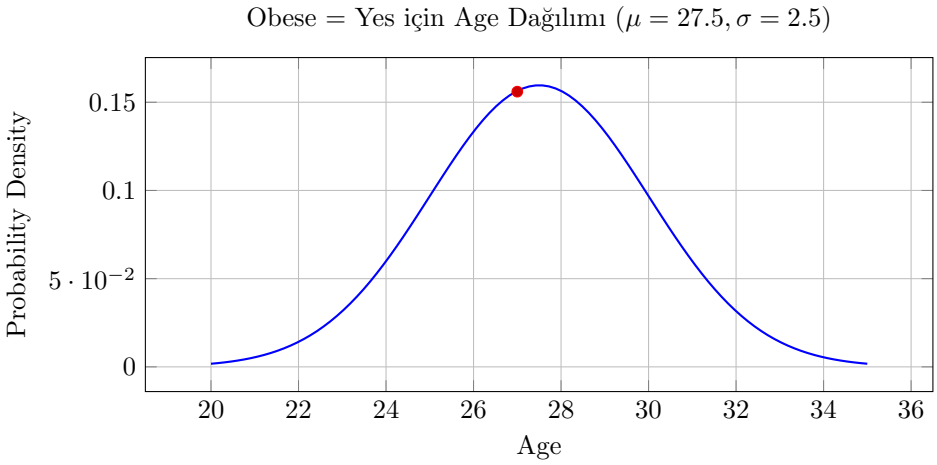
$$\textbf{Obese = Yes:} \quad 0.156 \times 0.0798 \times 0.0017 \approx 0.000021$$

$$\textbf{Obese = No:} \quad 0.133 \times 0 \times 0.0001 \approx 0$$

Karar

Bu kişinin **Obese = Yes** olması daha yüksek olasılıklıdır.

Görsel: Normal Dağılım Eğrisi Örneği Aşağıdaki şekilde, Gaussian Naive Bayes'in her özellik için nasıl bir çan eğrisi (normal dağılım) oluşturduğunu görebilirsiniz. Test verisi (örneğin, Age = 27) bu eğrinin üzerinde bir noktaya denk gelir ve ilgili olasılık bu noktadan hesaplanır.



Kırmızı nokta, test verisinin Age=27 için olasılığını göstermektedir.

Hangi Varyantı Ne Zaman Kullanmalıyım?

- Özellikler **0/1** ise ⇒ Bernoulli
- Özellikler **kelime sayısı** veya frekans ise ⇒ Multinomial
- Özellikler **sürekli sayı** ise ve normal dağılıma yakın dağılımlar (örneğin yaş, boy, gelir) ⇒ Gaussian
- Gaussian için özellikler sürekli ama normal dağılıma uymuyorsa, transformasyon fonksiyonları ile normal dağılıma yakınlaştırılabilir

Not: Karışık Özellikler Durumu

Eğer veri setiniz hem sürekli hem de kategorik özellikler içeriyorsa:

- Özelliklerin büyük kısmı binary ise Bernoulli tercih edilir
- Özelliklerin büyük kısmı sürekli ise Gaussian tercih edilir
- Alternatif olarak, kategorik özellikleri sayısal değerlere dönüştürüp Gaussian kullanmak da mümkündür

1.8 Naive Bayes'in Avantajları

- Hızlı ve hesaplama açısından verimli
- Yüksek boyutlu verilerle iyi çalışır
- Eksik verilerle bile çalışabilir

1.9 Sınırlamaları

- Özelliklerin birbirinden bağımsız olduğu varsayımı her zaman gerçekçi değildir.
- Sayısal verilerle çalışmak için genellikle dağılım varsayımları gerekir (örneğin Gaussian Naive Bayes).