

Naive Bayes Algoritması

Anas Alhardi- 218264001011
Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilişim Teknolojileri Mühendisliği Anabilim Dalı
Konya / Selçuklu / Bosna Hersek Mh
anas.rajah1@gmail.com

Özet– Artan istenmeyen e-mail hacmi aynı zamanda (spam olarak da bilinen), güvenilir anti-spam filtrelerine ihtiyaç duyulmasına yol açmıştır. Spam e-postayı çok başarılı bir oranla otomatik olarak filtrelemek için makine öğrenimi teknikleri kullanılmaktadır. Bu makalede Naive Bayes algoritmasını incelenmiştir. Naive Bayes Algoritması, sıklıkla sınıflandırma görevleri için kullanılan bir denetimli makine öğrenimi algoritmasıdır. Bayes' teoremi kullanarak, belirli bir kanıt varlığında bir olayın olasılığını tahmin etme fikrinden yola çıkar. Algoritma, verilen bir veri noktasında özelliklerin değerlerine dayalı bir sınıf etiketinin olasılığını temsil eden bir model oluşturarak çalışır. Bu, her bir özelliğin belirli bir sınıfa ait olasılığını tahmin ederek, sınıfın kendisinin olasılığını hesaplamak suretiyle yapılır. Naive Bayes algoritmasının anahtar varsayımı, özelliklerin birbirlerinden bağımsız olduğudur. Bu, belirli bir özelliğin varlığının veya yokluğunun diğer özelliklerin olasılığını etkilemediği anlamına gelir. Özet olarak, Naive Bayes algoritması, sınıflandırma görevleri için güçlü bir araç olan, her bir özelliğin belirli bir sınıfa ait olasılığını tahmin ederek tahminler yapma. Bu makalenin ilk bölümünde Naive Bayes algoritmasının gelişim tarihi, argümanı, türevleri ve bu algoritmanın özelliklerinden bahsedilmiştir. İkinci bölümde ise spam e-posta probleminin çözümü için Python programı kullanılarak bu algoritmaya basit bir program uygulanmıştır. Son bölümde sonuçlar tartışılmıştır.

Anahtar Kelimeler – Makine Öğrenmesi, Naive Bayes, Sınıflandırma, Spam, e-mail sınıflandırması.

I. GİRİŞ

Naive Bayes algoritması, ilk kez 18. yüzyılda Thomas Bayes tarafından geliştirilen Bayes olasılık teoremine dayanan bir makine öğrenme algoritmasıdır. Bayes Teoremi, bir olayın meydana gelme olasılığı ile olayla ilgili ön bilgiler arasındaki ilişkiyi açıklayan matematiksel bir formüldür. Bayes Teoremini sınıflandırma amacıyla kullanma fikri ilk olarak 18. yüzyılda Fransız matematikçi Pierre-Simon Laplace tarafından önerildi. Laplace, Bayes Teoremini olayla ilgili önceki bilgilere dayanarak bir olayın meydana gelme olasılığını hesaplamak için kullandı. Ancak, bugün bildiğimiz şekliyle Naive Bayes algoritması 20. yüzyıla kadar geliştirilmedi. 1950'lerde, Naive Bayes algoritması ilk olarak bilgisayar bilimcisi ve mühendis Oliver Self tarafından spam filtreleme için kullanıldı. Self, e-postalarda görünen kelimelere göre e-postaları spam veya spam olmayan olarak sınıflandırmak için algoritmayı kullandı [1, 2].

1980'lerde Naive Bayes algoritması, onu metin sınıflandırma görevleri için kullanan bilgisayar bilimciler John Langford ve Robert Schapire tarafından daha da geliştirildi ve popüler hale getirildi. 1990'larda, algoritma daha da geliştirildi ve tıbbi teşhis ve kredi dolandırıcılığı tespiti dahil olmak üzere çeşitli uygulamalarda yaygın olarak kullanılmaya başlandı. Bugün, Naive Bayes algoritması çeşitli uygulamalarda hala yaygın olarak kullanılmaktadır ve makine öğrenimi alanında bir kıyaslama algoritması olarak kabul edilmektedir. Basit ve verimli olduğu için genellikle temel model olarak kullanılır ve genellikle daha karmaşık makine öğrenimi modelleri için bir başlangıç noktası olarak kullanılır. Yaygın kullanımına rağmen, Naive Bayes algoritmasının bazı sınırlamaları vardır. Veri kümesindeki tüm özelliklerin birbirinden bağımsız olduğu varsayımı, gerçek dünya verilerinde genellikle doğru değildir ve bu da daha az doğru tahminlere yol açabilir. Bununla birlikte, Naive Bayes algoritması, çeşitli uygulamalarda yaygın olarak kullanılan güçlü bir araçtır. Naive Bayes algoritmasının en önemli avantajlarından biri basitliği ve verimliliğidir. Minimum veri ön işlemesi gerektiren hızlı ve uygulaması kolay bir algoritmadır, bu da onu büyük veri kümelerini işlemek için çok uygun hale getirir. Ayrıca eksik verileri işleyebilir ve hem sürekli hem de ayrık verilerle çalışabilir. Sadeliği ve verimliliğine ek olarak, Naive Bayes algoritmasının bir takım başka faydaları da vardır. Gürültüye karşı dayanıklıdır ve yüksek boyutlu verileri işleyebilir, bu da onu çeşitli uygulamalar için çok uygun hale getirir. Ayrıca, birçok gerçek dünya veri setinde yaygın olan dengesiz verileri de işleyebilir. Sınırlamalarına rağmen, Naive Bayes algoritmasının çeşitli uygulamalar için güçlü ve etkili bir araç olduğu kanıtlanmıştır. Doğal dil işleme, görüntü sınıflandırma ve tıbbi teşhis gibi alanlarda yaygın olarak kullanılır ve genellikle daha karmaşık makine öğrenimi modelleri için temel model olarak kullanılır. Genel olarak, Naive Bayes algoritmasının geçmişi, kökleri 18. yüzyıla kadar uzanan ve günümüze kadar gelişmeye ve gelişmeye devam eden uzun ve hikâyesidir. Zamanın testinden geçen ve makine öğrenimi alanında yaygın olarak kullanılan ve saygı duyulan bir araç olmaya devam eden basit ama güçlü bir algoritmadır [3, 4].

Bayes Teoremi

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

$P(A|B)$ = 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

Bayes Yasasının Türetilmesi

$$P(A, B) = P(A, B) \quad (2)$$

$$P(A|B) \times P(B) = P(B|A) \times P(A) \quad (3)$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (4)$$

Naive Bayes algoritması türleri. Üç tür Naive Bayes algoritması vardır [5].

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

Gaussian Naive Bayes algoritması

Sürekli öznitelik değerlerine sahip olduğumuzda, her sınıfla ilişkili değerlerin Gauss veya Normal dağılıma göre dağıtıldığı varsayımında bulunduk. Örneğin, eğitim verilerinin sürekli bir x özniteliği içerdiğini varsayalım. Önce verileri sınıfa göre bölümlere ayırırız ve ardından her sınıftaki x'in ortalamasını ve varyansını hesaplarız. μ_i değerlerin ortalaması olsun ve σ_i^2 i'inci sınıfla ilişkili değerlerin varyansı olsun. Diyelim ki bir gözlem değerimiz var x_i . Daha sonra, bir sınıfa verilen x_i 'nin olasılık dağılımı aşağıdaki denklemle hesaplanabilir –

$$P(x_i|y_i) = \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}} \quad (5)$$

Multinomial Naive Bayes algoritması

Multinomial Naive Bayes modelinde, örnekler (özellik vektörleri), belirli olayların birçok terimli (p_1, \dots, p_n) tarafından üretildiği frekansları temsil eder; burada p_i , i olayının meydana gelme olasılığıdır. Çok terimli Naive Bayes algoritması, çok terimli olarak dağıtılan veriler üzerinde kullanmak için tercih edilir. Metin sınıflandırma sınıflandırmasında kullanılan standart algoritmalarından biridir.

Bernoulli Naive Bayes algoritması

Çok değişkenli Bernoulli modelinde, özellikler girdileri tanımlayan bağımsız boole değişkenleridir (ikili değişkenler). Tıpkı Multinomial model gibi, bu model de terim frekansları yerine ikili terim oluşum özelliklerinin kullanıldığı belge sınıflandırma görevleri için popülerdir.

Naive Bayes algoritmasının uygulamaları

Naive Bayes, en basit ve hızlı sınıflandırma algoritmalarından biridir. Büyük hacimli veriler için çok uygundur. Aşağıdakiler gibi çeşitli uygulamalarda başarıyla kullanılır:

1. Spam filtreleme
2. Metin sınıflandırması
3. Duygu analizi
4. Tavsiye sistemleri

Bilinmeyen sınıfın tahmini için Bayes olasılık teoremini kullanır [6].

Naive Bayes algoritmasının avantajları ve dezavantajları

Naive Bayes algoritmasının avantajları aşağıdaki gibidir [7]:

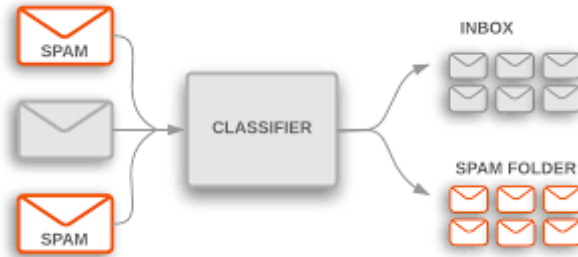
1. Naive Bayes hızlı, basit ve uygulaması kolay bir algoritmadır. Bu nedenle, sınırlı miktarda veriye sahip olduğumuzda daha karmaşık modellerden daha iyi performans gösterebilir.
2. Naive Bayes, ikili ve çok sınıflı sınıflandırma için kullanılabilir.
3. Metin sınıflandırma problemleri için mükemmel bir seçimdir. Spam e-posta sınıflandırması için popüler bir seçimdir.
4. Küçük veri kümesi üzerinde kolayca eğitilebilir.
5. Naive Bayes, sayısal ve kategorik verilerle iyi çalışır. Naive Bayes kullanarak regresyon gerçekleştirmek için de kullanılabilir.

Naive Bayes algoritmasının dezavantajları aşağıdaki gibidir [7]:

1. Naive Bayes algoritması, Bayes teoreminin kullanımını içerir. Bu nedenle, belirli eksik değerlerimiz veya eksik değer kombinasyonumuz olduğunda iyi çalışmaz.
2. Naive Bayes algoritması, basit kategorilerimiz olduğunda iyi çalışır. Ancak, kelimeler arasındaki ilişki önemli olduğunda iyi çalışmaz.

II. MATERYAL VE METOT

Spam e-mail, İnternet üzerinden büyük bir sorun haline gelir. İstenmeyen posta; zaman, depolama alanı ve iletişim bant genişliği kaybına neden olur. Spam e-mail sorunu yıllardır daha da artıyor. Son istatistiklerde, tüm e-postaların %40'ı istenmeyen e-postalardır ve bu da internet kullanıcılarının yılda yaklaşık 355 milyon dolara mal olmaktadır. Bilgi mühendisliği yaklaşımında, e-posta filtrelemede kullanılan iki genel yaklaşım vardır. Bilgi mühendisliği yaklaşımında hangi e-postaların istenmeyen posta ya da jambon olarak sınıflandırıldığına bağlı olarak bir dizi kural belirtilmelidir. Makine öğrenimi yaklaşımı, bilgi mühendisliği yaklaşımından daha verimlidir, herhangi bir kural belirtilmesini gerektirmez. Bunun yerine, bir dizi eğitim numunesi, bu örnekler önceden sınıflandırılmış e-posta mesajları kümesidir. Daha sonra bu e-posta mesajlarından sınıflandırma kurallarını öğrenmek için belirli bir algoritma kullanılır. Makine öğrenimi yaklaşımı yaygın olarak araştırılmıştır ve e-posta filtrelemede kullanılabilecek birçok algoritma vardır. Bunlar arasında Naive Bayes, taşıyıcı vektör makineleri, Nöral Ağlar, K'ya en yakın komşu, Rough setleri ve yapay bağışıklık sistemi yer alır [8].



Şekil. 1 Spam sınıflandırması [9]

Bu projede, spam'i tahmin etmek ve ham veya spam olarak sınıflandırmak için bir Naive Bayes sınıflandırıcısı oluşturdum. Python ve Scikit-Learn kullanarak Naive Bayes sınıflandırmasını uyguladım. Veri setimiz, 747'i spam olarak etiketlenmiş ve 4,827'i ham olarak etiketlenmiş 5574 İngilizce posta mesajından oluşmaktadır.

TABLE I: Sınıflandırma için kullanılan veriler

Mesajlar	Tutar	
	Mesajlar	Yüzde (%)
Hamlar	4,827	86,60
Spamlar	747	13,40
Total	5,574	100,00



Şekil. 1 sınıflandırma aşamaları

Naive Bayes algoritmasının aşamaları [10]

- Adım 1- Temel kitaplıkları içe aktarın
- Adım 2- Veri kümesini içe aktarma
- Adım 3- Veri ön işleme
- Adım 4- Modeli eğitme
- Adım 5- Modelin test edilmesi ve değerlendirilmesi

Spam sınıflandırmasında kullanılan birçok model, kitaplık ve program vardır. Açıklama çok uzun olmasın diye en kolay yolu seçtim.

Pandas kütüphanesini import edeceğiz. Bu, bir ayırıcı tanımlamamıza ve sütunları buna göre yeniden adlandırmamıza olanak tanır:

```
In [1]: import pandas as pd
```

Dosyayı okuyoruz ve ilk 5 satırı gösteriyoruz:

```
In [2]: df = pd.read_csv("spam.csv")
df.head()
```

```
Out[2]:
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Verileri gruplar halinde gösteriyoruz. Ham ve spam:

```
In [3]: df.groupby('Category').describe()
```

```
Out[3]:
```

	count	unique	Message
Category			top freq
ham	4825	4516	Sorry, I'll call later 30
spam	747	641	Please call our customer service representativ... 4

Yeni bir alan ekliyoruz. Spam ve ham e-mail mesajlarını ayırt etmek için:

```
In [4]: df['spam']=df['Category'].apply(lambda x: 1 if x=='spam' else 0)
df.head()
```

```
Out[4]:
```

	Category	Message	spam
0	ham	Go until jurong point, crazy.. Available only ...	0
1	ham	Ok lar... Joking wif u oni...	0
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	1
3	ham	U dun say so early hor... U c already then say...	0
4	ham	Nah I don't think he goes to usf, he lives aro...	0

Veri kümemizi test ve eğitim/öğrenme kümeleri olarak ikiye ayıralım ;(Test-Train

```
In [7]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df.Message,df.spam)
```

İşlemleri Naive Bayes modeline göre eğitiyoruz:

```
In [23]: from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(X_train_count,y_train)
```

```
Out[23]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

```
In [31]: from sklearn.feature_extraction.text import CountVectorizer
v = CountVectorizer()
X_train_count = v.fit_transform(X_train.values)
X_train_count.toarray()[:2]
```

```
Out[31]: array([[0, 0, 0, ..., 0, 0, 0],
               [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

Model testi yapıyoruz:

```
In [37]: emails = [
        'Hey mohan, can we get together to watch football game tomorrow?',
        'Upto 20% discount on parking, exclusive offer just for you. Dont miss this reward!'
    ]
emails_count = v.transform(emails)
model.predict(emails_count)
```

```
Out[37]: array([0, 1], dtype=int64)
```

Modelin doğruluğunu gösteriyoruz:

```
In [38]: X_test_count = v.transform(X_test)
model.score(X_test_count, y_test)
```

```
Out[38]: 0.9827709978463748
```

III. TARTIŞMA/SONUÇ

Naive Bayes algoritmasını incelemiş olup bu algoritmanın gelişim tarihine genel bir bakıştan bahsedilmiştir. Bunun üzerine spam sınıflandırma problemi olan pratik bir uygulama geliştirilmeye çalışılmıştır. Makine öğreniminde bu sorunu çözmek için çalışan birçok algoritma vardır, ancak Naive Bayes algoritması spam sınıflandırma probleminde en ünlü ve en yaygın makine öğrenimi yöntemlerinden biridir. Geliştirdiğimiz bu algoritmayı, 747'si spam olarak sınıflandırılan ve 4.827'si ham olan 5.574 İngilizce postadan oluşan bir dizi veri içeren bir Excel dosyasında test edilmiştir. Test sonucunda beklendiği gibi modelin doğruluğu 0.982770099'a ulaştığı için iyi bir sonuç ortaya çıkmıştır. Bu algoritma birçok metin sınıflandırma probleminde kullanılabilir ayrıca farklı alanlarda birçok uygulaması vardır. Naive Bayes algoritması basittir, yalnızca birkaç eğitim veri seti gerektirir bu da onu benzersiz kılar ve gelecekte birçok çözüme katkıda bulunabilir.

KAYNAKLAR

- [1] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.
- [2] P. Pedamkar. "Discover the Naive Bayes Algorithm - eduCBA." <https://www.educba.com/naive-bayes-algorithm/> (accessed 2022 December 31).
- [3] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009.
- [4] HolyPython. "Naive Bayes Classifier History." <https://holypython.com/nbc/naive-bayes-classifier-history/> (accessed 2022 December 31).
- [5] G. Sarah, "Introduction to machine learning with python," 2022.
- [6] K. Eremenko, "Hadelin de Ponteves, SuperDataScience Support, et al," *Machine Learning AZ™: Hands-On Python & R In Data Science*. URL: <https://www.udemy.com/course/machinelearning/>. Accessed, pp. 03-01, 2022.
- [7] R. Saxena. "How The Naive Bayes Classifier Works In Machine Learning." <https://dataaspirant.com/naive-bayes-classifier-machine-learning/> (accessed 10 January 2023).
- [8] W. A. Awad and S. ELseuofi, "Machine learning methods for spam e-mail classification," *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 3, no. 1, pp. 173-184, 2011.
- [9] G. Developers. "Text Classification." <https://developers.google.com/machine-learning/guides/text-classification?hl=en> (accessed 10 January 2023).
- [10] Turing. "An Introduction to Naive Bayes Algorithm for Beginners." <https://www.turing.com/kb/an-introduction-to-naive-bayes-algorithm-for-beginners> (accessed 11 January 2023).