ÖDEV 2
Naive Bayes ile Spam Mail Detection

Selen Erdoğan
selenerdogan2019@gtu.edu.tr
Elektronik Mühendisliği Bölümü, GTÜ

I. GİRİŞ

E-posta, günümüz dünyasında en önemli iletişim araçlarından biridir. E-posta kullanımı dünya genelinde önemli ölçüde artmıştır. 2015 yılında gönderilen ve alınan e-posta sayısı günde 205 milyarı aşmıştır. Bu nedenle, bu spam e-postaları filtreleme, dünya genelindeki e-posta kullanıcıları için acil bir ihtiyaç haline gelmiştir. Naive Bayes, sınıflandırma modelleri oluşturmak için basit bir yöntem sunmaktadır. Bu modeller, özellik değerlerinin vektörleriyle temsil edilen problem durumlarına sınıf etiketleri atar ve sınıf etiketleri sınırlı bir kümeden seçilir. Bu tür sınıflandırıcıları eğitmek için kullanılan algoritma ailesi, sınıf değişkeni verildiğinde bir özelliğin değerinin diğer tüm özelliklerin değerinden bağımsız olduğu öncülüne dayanır.
Naive Bayes modellerinde parametre tahmini genellikle maksimum olabilirlik yöntemini içerir. Bu nedenle, saf Bayes modeliyle çalışırken Bayes olasılığını benimsemeden veya herhangi bir Bayes tekniğini uygulamadan önce, parametre tahmini yapmak mümkündür.

II. ANALİZ

Bayes teoremi, e-posta filtreleri oluşturmak için uygulanır. Bayes teoremi, spam mail tanımlama konusu ile ilgili olarak kullanılan basit seviye ayrıştırıcıdır. Naive Bayes tekniği, Bayes teoremini kullanarak spam e-posta olasılıklarını belirlemek için kullanır. Bazı kelimelerin spam e-postalarda veya spam olmayan e-postalarda belirli olasılıkları vardır. Örneğin, Free kelimesinin hiçbir zaman spam olmayan bir e-postada görünmeyeceğini tam olarak biliyorsak, bu kelimeyi içeren bir ileti gördüğümüzde kesinlikle spam e-posta olduğunu söyleyebiliriz.Bu nedenle, e-postanın spam veya spam olmadığı olasılığını hesaplamak için Naive Bayes tekniği, Şekil 1'de görülen formülde gösterildiği gibi Bayes teoremini kullanır.

$$P(spam \mid word) = \frac{\mathrm{P(spam).\,P(word|spam)}}{\mathrm{P(spam).\,P(word|spam) + P(non-spam).\,P(word|non-spam)}}$$

Şekil 1

(i) P(spamword), bir e-postanın spam olduğu bilindiğinde belirli bir kelimenin bulunma olasılığıdır.
(ii) P(spam), herhangi bir verilen iletinin spam olma olasılığıdır.
(iii) P(wordspam), belirli bir kelimenin spam iletilerde görünme olasılığıdır.
(iv) P(non − spam), belirli bir kelimenin spam olmama olasılığıdır.
(v) P(wordnon − spam), belirli bir kelimenin spam olmayan iletilerde görünme olasılığıdır.

III. YÖNTEM

Sorunu çözmek amacıyla yazılan kodda, bir e-posta veri setini yüklendi, eğitim ve test setlerine bölündü, veriler temizlendi ve ardından Naive Bayes sınıflandırma modeli ile eğitim setini kullanarak spam ve ham e-postaları ayırmak için bir algoritma oluşturuldu. Daha sonra bu modeli kullanarak test setindeki e-postalar sınıflandırıldı ve gerçek etiketlerle karşılaştırarak modelin doğruluğunu değerlendirildi.

A. Veri Temizleme

```
training_set['content'] = training_set['content'].str.replace('\W', ' ', regex=True)
training_set['content'] = training_set['content'].str.lower()
training_set['content'] = training_set['content'].str.replace('subject', '', regex=False)
training_set['content'] = training_set['content'].str.replace('and', '', regex=False)
training_set['content'] = training_set['content'].str.replace('or', '', regex=False)
training_set['content'] = training_set['content'].str.replace('an', '', regex=False)
training_set['content'] = training_set['content'].str.replace('a', '', regex=False)
training_set['content'] = training_set['content'].str.replace('to', '', regex=False)
training_set['content'] = training_set['content'].str.replace('from', '', regex=False)
training_set['content'] = training_set['content'].str.replace('in', '', regex=False)
training_set['content'] = training_set['content'].str.replace('for', '', regex=False)
training_set['content'] = training_set['content'].str.replace('on', '', regex=False)
training_set['content'] = training_set['content'].str.replace('at', '', regex=False)

training_set
```

Şekil 2

Şekil 2'de görülecek olan yöntem ile metin içerisindeki tüm özel karakterler boşlukla, metin analizi sırasında büyük/küçük harf duyarlılığını ortadan kaldırmak amacıyla tüm metni küçük harfe ve sık kullanılan ama bir anlam taşımayan kelimeleri metinden çıkararak veri temizliği yapmak amaçlanmıştır. Bu işlemler, metin verilerini daha işlenebilir hale getirmek ve ardından bu temizlenmiş verileri kullanarak bir model oluşturmak veya metin madenciliği uygulamak için önemlidir. Özellikle spam filtreleme veya duygu analizi gibi uygulamalarda, metin verilerinin temiz ve standart hale getirilmesi, modelin daha iyi performans göstermesine yardımcı olabilir.

.

B. *Kelime Dağarcığı Oluşturma*

Her bir metin kelimelere ayırarak bir liste oluşturulur. Yani, her bir metin şimdi bir liste içindeki kelimelerden oluşan bir dizi haline gelir. Bu adım, her bir metni daha küçük bileşenlere (kelimelere) ayırarak analizi daha kolay hale getirir. Metini sayısallaştırmak amacıyla bu işlem yapılmıştır.

Hesaplanan kelime sayılarını içeren yapı DataFrame yapısına dönüştürülür ve training_set DataFrame yapısı ile birleştirilir. Bu birleştirme işlemi, her bir metni temsil eden orijinal özelliklere ek olarak, her kelimenin metinlerdeki frekansını içeren yeni özellikleri de içeren genişletilmiş bir veri çerçevesi oluşturur. Bu sayede, her bir metni temsil eden vektörler elde edilir ve bu vektörler, makine öğrenimi modellerine girdi olarak sağlanabilir.

C. *Olaasılık Hesaplamaları Yapma*

Eğitim veri setindeki "spam" (istenmeyen posta) ve "ham" (istenilen posta) mesajları izole edilerek, olasılık hesaplamaları yapıldı. 'labels' sütununda 'spam' ve 'ham' etiketlerine sahip olan mesajları ayırıldı, iki ayrı DataFrame olan spam_mail ve ham_maili oluşturuldu. Toplam eğitim veri setine göre spam ve ham mesajların olasılıklarını hesaplandı. Gaussian pürüzsüzleştirmesi için kullanılacak olan alfa değerini tanımlandı. Bu değer, olasılıkları sıfıra yaklaştırmaktan kaçınmak için kullanılır. Bu hesaplamalar, bir spam filtresi veya doğal dil işleme modeli oluştururken kullanılan temel olasılık değerlerini sağlamak için yapıldı.

D. *Test Veri Setine Modelin Uygulanması*

Bir test setindeki her bir e-postayı sınıflandırmak için kullanılan fonksiyon yazıldı. Gelen e-poste metni işlendi. Oluşturulan döngü ile, e-postadaki her bir kelimenin, spam ve ham mesajlardaki olasılıklarıyla çarpılmasını sağlandı. Eğer bir kelime, eğitim setindeki spam mesajlarda bulunuyorsa, spam olasılığını güncellendi. Aynı şekilde, eğer kelime ham mesajlarda bulunuyorsa, ham olasılığını güncellendi. Son olarak, güncellenmiş olasılıklara göre e-postanın "ham" veya "spam" olarak sınıflandırılmasını gerçekleştirildi. Test setindeki her bir e-postayı sınıflandırıldı ve sınıflandırılmış sonuçları içeren bir sütun eklendi ('predicted') Şekil 3. Test seti üzerinde bu sınıflandırma işlemini gerçekleştirmek için apply fonksiyonu kullanıldı.

	labels	content	b_labels	predicted
0	ham	Subject: enron / hpl actuals - nov . 7 , 2000\...	0	ham
1	spam	Subject: exclusive positions in montanayoocwo\...	1	spam
2	ham	Subject: natural gas nomination for december 2...	0	ham
3	ham	Subject: imperial sugar ' s volumes will be 14...	0	ham
4	ham	Subject: re : first delivery - wagner oil\r\nv...	0	ham
...	...	...	...	...
1288	ham	Subject: put the 10 on the ft\r\nthe transport...	0	ham
1289	ham	Subject: 3 / 4 / 2000 and following noms\r\nhp...	0	ham
1290	ham	Subject: calpine daily gas nomination\r\n>\r\n...	0	ham
1291	ham	Subject: industrial worksheets for august 2000...	0	ham
1292	spam	Subject: important online banking alert\r\ndea...	1	spam

Şekil 3

E. *Başarı Metrikleri Hesaplanması*

Makine öğrenmesinde başarı metrikleri, bir modelin performansını değerlendirmek için kullanılan ölçümlerdir. Yazılan kod ile, precision, recall ve F1 score'u hesaplamak için gerekli olan true positives, false positives ve false negatives değerleri kullanıldı. Elde edilen sonuçlar Şekil 4'te gösterilmiştir.

```
Correct: 1164
Incorrect: 129
Accuracy: 0.9002320185614849
Precision: 0.9965753424657534
Recall: 0.9030256012412723
F1 Score: 0.9474969474969475
```

Şekil 4

Accuracy (Doğruluk):Bu, doğru tahmin edilen toplam örnek sayısının toplam örnek sayısına oranıdır. Yani, doğru tahmin edilenlerin toplam örneklere oranıdır. Bu durumda, doğruluk oranı yaklaşık %90'dır. Eğer sınıflar arasında dengesiz bir dağılım varsa (yani, spam ve ham e-posta sayıları çok farklı), doğruluk tek başına yeterli bir metrik olmayabilir.

Precision (Hassasiyet):Bu, spam olarak tahmin edilen e-postaların gerçekten spam olan e-postaların oranını ölçer. Yani, yanlış pozitiflerin toplam pozitiflere oranıdır. Bu durumda, spam olarak tahmin edilen e-postaların %99.66'sı gerçekten spamdır.

Recall (Duyarlılık):Bu, gerçekten spam olan e-postaların ne kadarının doğru bir şekilde spam olarak tahmin edildiğini ölçer. Yani, yanlış negatiflerin toplam pozitiflere oranıdır. Bu durumda, gerçekten spam olan e-postaların %90.30'u doğru bir şekilde spam olarak tahmin edilmiştir.

F1 Score: Bu, hassasiyet ve duyarlılık arasındaki dengeyi sağlamak için kullanılan bir metriktir. Hassasiyet ve duyarlılık arasındaki harmonik ortalamadır. Bu durumda, F1 skoru yaklaşık %94.75'tir.

IV. ANALİZ

Bu çalışmada, Naive Bayes sınıflandırma algoritması kullanılarak e-posta spam filtreleme modeli oluşturuldu. Veri seti, ham ve spam e-postaları içeren bir etiketli veri setinden elde edildi. Eğitim ve test setleri oluşturulduktan sonra, kelimelerin sayısal temsili elde edildi. Naive Bayes'in parametre tahmini gerçekleştirildi ve elde edilen model test seti üzerinde değerlendirildi. Test sonuçları, modelin yüksek doğruluk, hassasiyet ve F1 skoru elde ettiğini gösterdi. Bu çalışma, Naive Bayes'in e-posta spam filtreleme uygulamalarında etkili bir yöntem olduğunu doğrulamaktadır.

KAYNAKÇA

[1] Nurul Fitriah Rusland, Analysis of Naïve Bayes Algorithm for Email Spam Filtering across Multiple Datasets.2017.

,

ELM472 Makine Öğrenmesinin Temelleri