MÜHENDİSLİK MİMARLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ
MAKİNE ÖĞRENMESİ PROJE RAPORU
E-MAİL SPAM
2021

18110131032
KÜBRA KABALCI

18110131035
BURCU GENÇ

İÇİNDEKİLER

Özet

Kelime anlamı itibariyle “istenmeyen” anlamına gelen spam mail; ticari reklam amaçlı veya belli bir konuda kamuoyu oluşturmak amacıyla gönderilen toplu maillerdir.

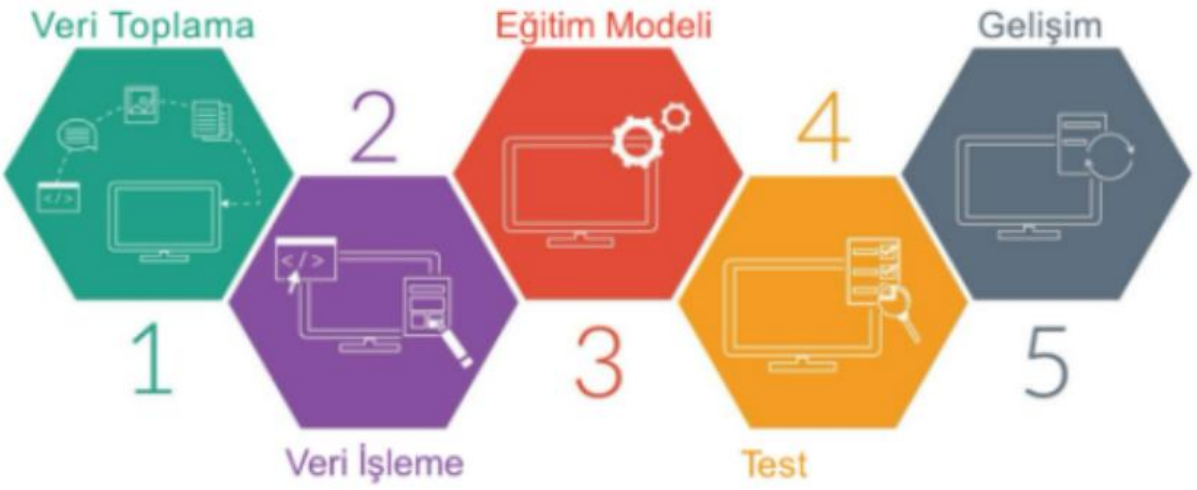
Spam e-mail, e-posta adresi sahibinin rızası dışında ve talebi olmadan gönderilen bir mail türüdür. Aynı mesajın kopyasının milyonlarca kişiye iletildiği spam e-posta, zorlayıcı nitelikte istenmeyen mesajlardır. Spam mailler, farklı amaçlarla gönderilebilir.

Bu projemizde ise gelen maillerin spam olup olmadığına karar verebilen bir makine öğrenmesi tasarladık.

Giriş

Veri setimizde spam mailler için 1 ve spam; spam olmayan mailler için 0 ve ham olarak etiketleri bulunuyor. Bu ayrım ise mailerin konularına göre belirlenmiştir. Makinemizin öğrenmesi için sınıflandırma yöntemi kullanılmıştır.

Makine Öğrenmesi Süreci



Veri Toplama

Veri setimiz için kaggle sitesini kullandık.

https://www.kaggle.com/venky73/spam-mails-dataset

In [121]:
```
#Kütüphaneleri içe aktarma

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
import re
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report , confusion_matrix , accuracy_score
from sklearn.linear_model import LogisticRegression

# label_num = 0 Spam değil demektir.
# label_num = 1 Spam demektir.
df = pd.read_csv("spam_ham_dataset.csv")
df.head()
```

Out[121]:

	Unnamed: 0	label	text	label_num
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0

Satır ve Sütun sayısını yazdırdık.

```
In [77]: df.shape
Out[77]: (5171, 4)
```

Veri Ön İşleme

Yüzdelik dilimlerle veri çerçevesinin istatistiksel açıklaması döndürüldü. Bizim veri setimizde sayısal olan "Unnamed:0 ve label_num" sütunları olduğu için bunların istatistiksel özetini görebiliyoruz.

```
In [75]: df.describe()
```

Out[75]:

	Unnamed: 0	label_num
count	5171.000000	5171.000000
mean	2585.000000	0.289886
std	1492.883452	0.453753
min	0.000000	0.000000
25%	1292.500000	0.000000
50%	2585.000000	0.000000
75%	3877.500000	1.000000
max	5170.000000	1.000000

Her sütunda eksik değer var mı yok mu ona bakıldı ve olmadığını görmüş olduk.

```
In [76]: df.isnull().sum()
Out[76]: Unnamed: 0    0
         label         0
         text          0
         label_num     0
         dtype: int64
```

Label_num içersindeki 0 ve 1 (spam=1, ham=0) olmak üzere toplam etiket sayısına bakıldı.

```
In [78]: df.value_counts('label_num')
Out[78]: label_num
         0    3672
         1    1499
         dtype: int64
```

Verisetimizdeki ham verileri listeledik.

```
In [11]: ham = df[df['label_num'] == 0] #sadece ham olanları listeledik
```

```
In [12]: ham
```

Out[12]:

	Unnamed: 0	label	text	label_num
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0
5	2949	ham	Subject: ehronline web address change\r\nthis ...	0
...	...	...	...	...
5165	2849	ham	Subject: fw : crosstex energy , driscoll ranch...	0
5166	1518	ham	Subject: put the 10 on the ft\r\nthe transport...	0
5167	404	ham	Subject: 3 / 4 / 2000 and following noms\r\nhp...	0
5168	2933	ham	Subject: calpine daily gas nomination\r\n>\r\n...	0
5169	1409	ham	Subject: industrial worksheets for august 2000...	0

3672 rows × 4 columns

Veri setimizdeki spam verileri listeledik.

```
In [13]: spam = df[df['label_num']==1] #sadece spam olanları listeler..
```

```
In [14]: spam
```

Out[14]:

	Unnamed: 0	label	text	label_num
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1
7	4185	spam	Subject: looking for medication ? we ` re the ...	1
10	4922	spam	Subject: vocable % rnd - word asceticism\r\nvc...	1
11	3799	spam	Subject: report 01405 !\r\nwffur attion brom e...	1
13	3948	spam	Subject: vic . odin n ^ ow\r\nberne hotbox car...	1
...	...	...	...	...
5159	4381	spam	Subject: pictures\r\nstreamlined denizen ajar ...	1
5161	4979	spam	Subject: penny stocks are about timing\r\nnoma...	1
5162	4162	spam	Subject: anomaly boys from 3881\r\nuosda apapr...	1
5164	4365	spam	Subject: slutty milf wants to meet you\r\ntake...	1
5170	4807	spam	Subject: important online banking alert\r\ndea...	1

1499 rows × 4 columns

Veri kümesindeki toplam spam ve ham sayısını grafikte göstermiş olup 3672 ham sayısına 1499 civarında ise spam veriyi gösterir.

```
In [123]:
df['label'].value_counts().plot.bar(color = ["b","r"])
plt.title('Veri kümesindeki toplam ham ve spam sayısı')

plt.show()
```

Veri Temizleme

Text sütunundaki konular düzenli hale getirildi. İlgili '\r' , '\n' , '#' "Subject: " kaldırıldı.

"we ' re → we are",

"they ' re → they are",

"you ' re → you are"

haline getirildi.

Text_clean ile de son haline bakıldı.

```
In [88]: #verisetinin temizlenmesi
         def preprocess(text):
             text = text.replace('\r',' ')
             text = text.replace('\n',' ')
             text = text.replace('#',' ')
             text = text.replace("we ' re","we are")
             text = text.replace("they ' re","they are")
             text = text.replace("you ' re","you are")
             text = text.replace("Subject:"," ")

             return text
```

```
In [89]: df['text_clean']=df['text'].map(preprocess)
```

```
In [90]: df.head(4993)  #sonunda metni düzenleyip dönüştürüyoruz..
```

Out[90]:

	Unnamed: 0	label	text	label_num	text_clean
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0	enron methanol ; meter : 988291 this is a...
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0	hpl nom for january 9 , 2001 (see attached...
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0	neon retreat ho ho ho , we are around to th...
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1	photoshop , windows , office . cheap . main ...
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0	re : indian springs this deal is to book th...

Eğitim Modeli

Verimizi eğitim_set ve test_set olarak ayırdık.

```
X_train , X_test , Y_train , Y_test =train_test_split(X, Y, test_size = 0.2, random_state = 42)
```

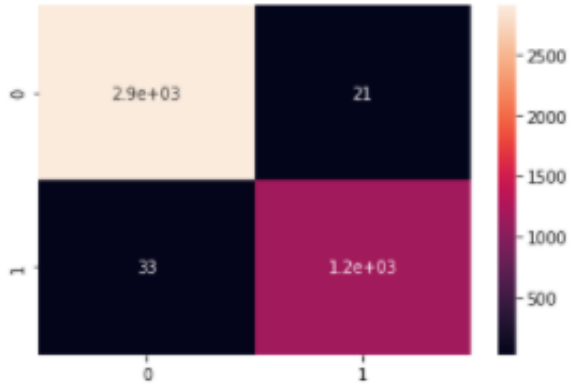
Model_selection kütüphanesinin modülü olan train_test_split()'e vereceğimiz ilk iki parametre X ve Y, yani veri kaynağı olarak ne kullanılacak onu belirtmiş oluyoruz. test_size parametresi ile test için ne kadar bir veri ayrılacak onu belirtiyoruz. Yukarıdaki 0.2 verinin %20'sini test için ayır demek. Bu parametreyi atamakla aslında train_size'ı da dolaylı olarak 0.8 yapmış oluyoruz. Yani yukarıda veri setinin %20'sini test, %80'ini eğitim olarak ayırmış bulunuyoruz.

Naive Bayes Algoritması

Koşullu olasılık kuralını kullanarak bir öğenin belirli bir kategoriye girme olasılığını hesaplayan bu algoritma, oldukça etkili bir denetimli makine öğrenimi algoritması olarak bilinir. Sınıf değişkeninin değeri göz önüne alındığında, Bayes'in teoremini verilere uygulayarak, her özellik çifti arasında saf bir koşullu bağımsızlık varsayımı ile çalışır.

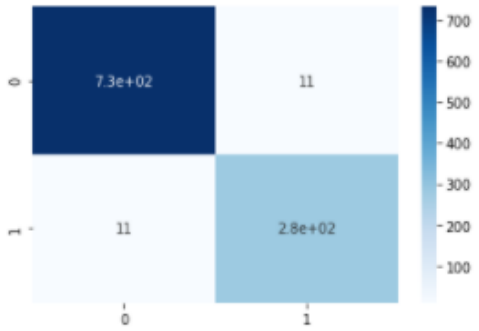
Eğitim ve test tahmin sonuçları oluşturuldu. Ardından sınıflandırma algoritmasının performans ölçümü için karışıklık matrisi kullanılır. Her bir e-postanın spam olup olamayacağına karışıklık matrisi ile bulunur.

```
In [98]: #NAIVE_BAYES algoritması ile modelin eğitilmesi
         NB_classifier =MultinomialNB() #çok terimli naive bayes sınıflandırıcısı
         NB_classifier.fit(X_train , Y_train)
Out[98]: MultinomialNB()
In [99]: #Training_set sonucunu tahmin etme
         Y_pred_train = NB_classifier.predict(X_train)
         print(Y_pred_train)
         [0 0 0 ... 1 0 0]
In [100]: #Test_set sonucunu tahmin Etme
          Y_pred_test = NB_classifier.predict(X_test)
          print(Y_pred_test)
          [0 1 0 ... 1 0 0]
In [101]: cm_test = confusion_matrix(Y_test, Y_pred_test)
          cm_train = confusion_matrix(Y_train , Y_pred_train)
In [102]: sns.heatmap(cm_train , annot = True)
Out[102]: <AxesSubplot:>
```



Ardından eğitim ve test modelinin doğruluk oranına bakılır. Elde edilen sonucun 0.978 test olduğu, 0.986 in ise eğitim olduğu gözlenir.

```
In [103]: sns.heatmap(cm_test, annot = True,cmap='Blues')
Out[103]: <AxesSubplot:>
```

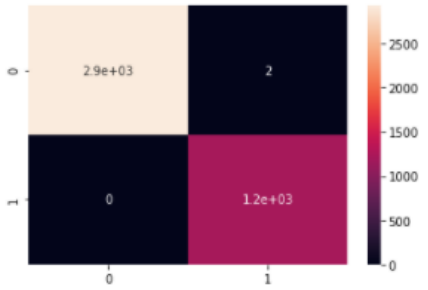



```
In [104]: #Test setinin doğruluk oranı
          accuracy_score(Y_test , Y_pred_test)
Out[104]: 0.978743961352657

In [105]: #Eğitim setinin doğruluk oranı
          accuracy_score(Y_train , Y_pred_train)
Out[105]: 0.9869439071566731
```

Gelişim

Logistic regresyon algoritmasını kullanarak daha az hata oranına ulaştık ve Doğruluk oranının Naive Bayes algoritmasına göre daha yüksek olduğunu görmüş olduk.

```
In [200]: model = LogisticRegression()

In [201]: # Lojistik Regresyon modelini eğitim verileriyle eğitmek
          model.fit(X_train, Y_train)

In [202]: # eğitim verileriyle ilgili tahmin

          prediction_on_training_data = model.predict(X_train)
          accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)

In [110]: cm_train = confusion_matrix(Y_train , prediction_on_training_data)

In [111]: sns.heatmap(cm_train , annot = True)
Out[111]: <AxesSubplot:>
```



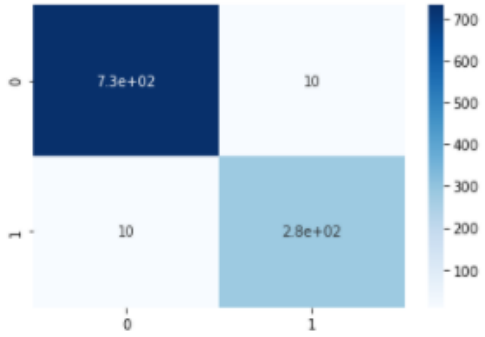

```
In [205]: print('Eğitim verilerinin doğruluğu : ', accuracy_on_training_data )
Eğitim verilerinin doğruluğu :  0.9995164410058027
```

In [113]:
```
# test verileriyle ilgili tahmin

prediction_on_test_data = model.predict(X_test)
accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)
```

In [114]:
```
cm_test = confusion_matrix(Y_test, prediction_on_test_data)
```

In [117]:
```
sns.heatmap(cm_test, annot = True,cmap='Blues')
```

Out[117]: <AxesSubplot:>



In [206]:
```
print('Test verilerinin doğruluğ : ', accuracy_on_test_data)
```

```
Test verilerinin doğruluğ :  0.9806763285024155
```