spam-email-analysis

October 28, 2023

[1]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df = pd.read_csv('spam.csv', encoding = 'ISO-8859-1')
df.head()
```

[2]:
```
     v1                                                 v2 Unnamed: 2  \
0   ham  Go until jurong point, crazy.. Available only …        NaN
1   ham                      Ok lar… Joking wif u oni…       NaN
2  spam  Free entry in 2 a wkly comp to win FA Cup fina…        NaN
3   ham  U dun say so early hor… U c already then say…       NaN
4   ham  Nah I don't think he goes to usf, he lives aro…        NaN

  Unnamed: 3 Unnamed: 4
0        NaN        NaN
1        NaN        NaN
2        NaN        NaN
3        NaN        NaN
4        NaN        NaN
```

[3]:
```
df.describe()
```

[3]:
```
          v1                      v2  \
count   5572                    5572
unique     2                    5169
top      ham  Sorry, I'll call later
freq    4825                      30

                                              Unnamed: 2  \
count                                                 50
unique                                                43
top      bt not his girlfrnd… G o o d n i g h t . . .@"
freq                                                   3
```

```
                   Unnamed: 3 Unnamed: 4
count                      12          6
unique                     10          5
top      MK17 92H. 450Ppw 16"    GNT:-)"
freq                        2          2
```

```
[4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

It makes no sense to keep last 3 groups which are almost empty

```
[5]: df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'], inplace=True)
```

```
[6]: df
```

```
[6]:      v1                                                 v2
     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…
     …     …                                                …
     5567  spam  This is the 2nd time we have tried 2 contact u…
     5568   ham              Will Ì_ b going to esplanade fr home?
     5569   ham  Pity, * was in mood for that. So…any other s…
     5570   ham  The guy did some bitching but I acted like i'd…
     5571   ham                         Rofl. Its true to its name

     [5572 rows x 2 columns]
```

```
[7]: # renaming the columns so that it becomes easier for us to understand their␣
      ↪purpose
```

```
[8]: df = df.rename(columns={'v1':'Target', 'v2':'Message'})
```

[9]:
```
# checking for null values
df.isnull().sum()
```

[9]:
```
Target     0
Message    0
dtype: int64
```

[10]:
```
df.duplicated().sum()
```

[10]:
```
403
```

Using scikit learn to encode the textual data to numbers

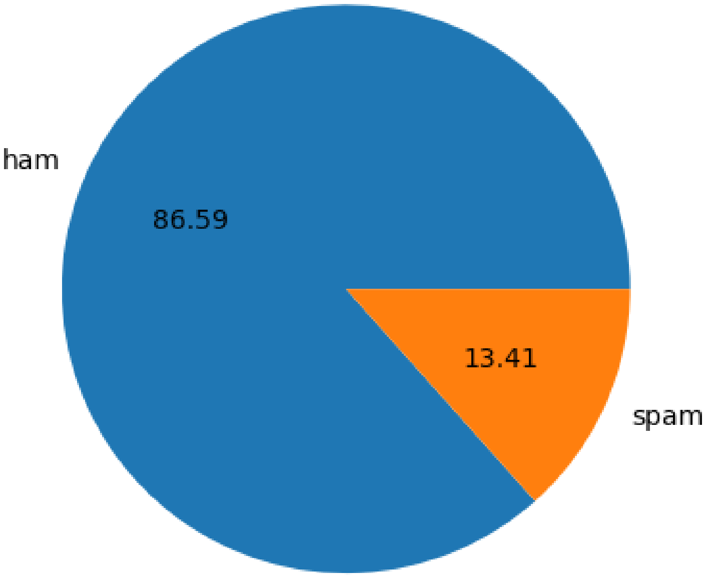
[11]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()

df['Target'] = encoder.fit_transform(df['Target'])
df['Target']
```

[11]:
```
0       0
1       0
2       1
3       0
4       0
       ..
5567    1
5568    0
5569    0
5570    0
5571    0
Name: Target, Length: 5572, dtype: int32
```

[12]:
```
df.head()
```

[12]:
```
   Target                                            Message
0       0  Go until jurong point, crazy.. Available only …
1       0                      Ok lar… Joking wif u oni…
2       1  Free entry in 2 a wkly comp to win FA Cup fina…
3       0  U dun say so early hor… U c already then say…
4       0  Nah I don't think he goes to usf, he lives aro…
```

[13]:
```
plt.pie(df['Target'].value_counts(), labels=['ham','spam'], autopct="%0.2f")
plt.show()
```



1 Getting the data ready to be split and predicted

```
[14]: X = df['Message']
      y = df['Target']
```

```
[23]: X
```

```
[23]: 0       Go until jurong point, crazy.. Available only …
      1                           Ok lar… Joking wif u oni…
      2       Free entry in 2 a wkly comp to win FA Cup fina…
      3       U dun say so early hor… U c already then say…
      4       Nah I don't think he goes to usf, he lives aro…
                                    …
      5567    This is the 2nd time we have tried 2 contact u…
      5568                Will Ì_ b going to esplanade fr home?
      5569    Pity, * was in mood for that. So…any other s…
      5570    The guy did some bitching but I acted like i'd…
      5571                               Rofl. Its true to its name
      Name: Message, Length: 5572, dtype: object
```

```
[15]: y
```

[15]:

```
0       0
1       0
2       1
3       0
4       0
       ..
5567    1
5568    0
5569    0
5570    0
5571    0
Name: Target, Length: 5572, dtype: int32
```

[16]:

```
from sklearn.model_selection import train_test_split
```

[17]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=101)
```

[18]:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import svm
```

[19]:

```
cv = CountVectorizer()
```

[20]:

```
X_train_cv = cv.fit_transform(X_train)
X_test_cv = cv.fit_transform(X_test)
```

[21]:

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
```

[24]:

```
lr.fit(X_train_cv, y_train)
prediction_train = lr.predict(X_train_cv)
```

[25]:

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_train, prediction_train)*100)
```

```
99.798070450976
```

[ ]:

```

```