Aslı Kurt/18030411057 & 18.05.2021

Erdicem Yücesan/15030411052

The Report Of the Spam Classification App ML

For the begining of this report our goal is briefly consist of explanation of the Machine Learning (ML) and how can machine learning be used for the security side with an Open Source example.

So, What is Machine Learning?

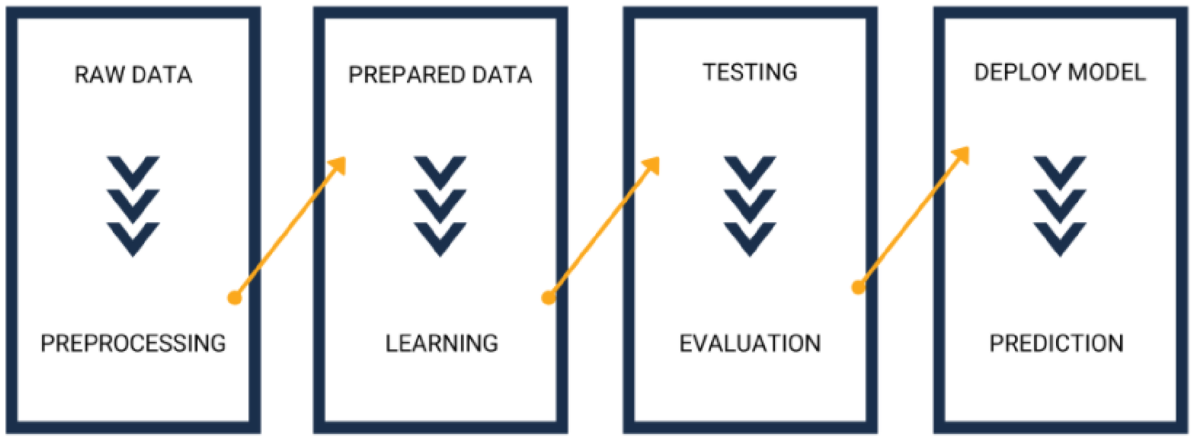
"Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves." (Expert.ai Team, 6 May 2020)

Also we can say that, the real means behind ML is 'Machine Teaching'. We know that what the machine needs to learn, so our job is to create a learning framework and provide properly formatted, relevant, clean data for the machine to learn from. (lexalytics, 2020)

How is it used for cybersecurity?

"Machine learning helps automate the process of finding, contextualizing, and triaging relevant data at any stage in the threat intelligence lifecycle."(Echosec Systems, 26 May 2020)

These are the steps of the data process in the ml.



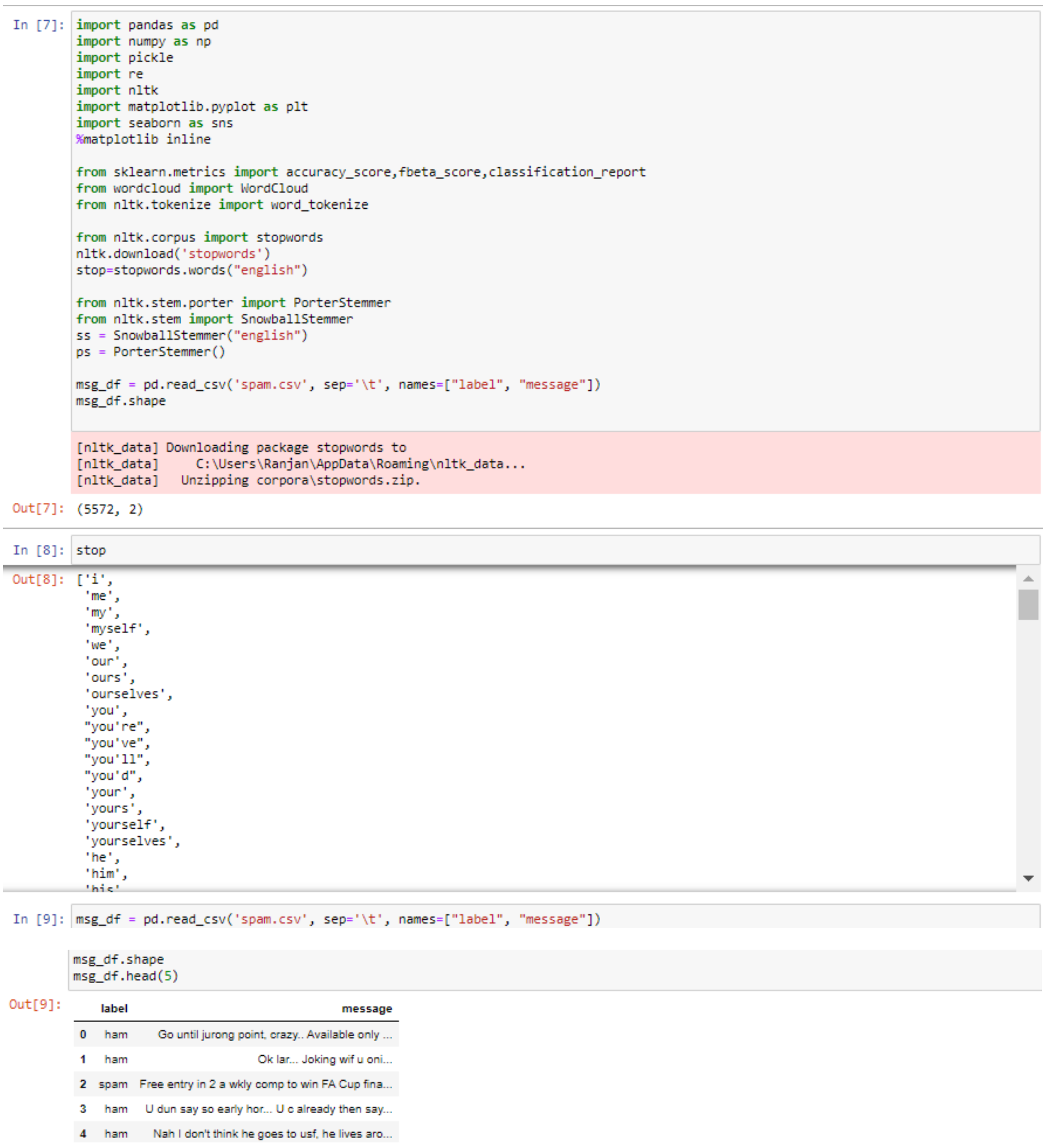
(https://www.echosec.net/blog/how-is-machine-learning-used-in-cybersecurity)

While performing the steps of the code in the open source example, the procedure given above is used.

Complete Deployment of Machine Learning Model Using NLP: Spam Classifier: Modelling Part

“First we create a model, once we have created a model, we will create a pickle file from the model and use that pickle file and we will deploy on the cloud herouku in order to do that we are gonna use flask framework in python But im gonna show u creation of the model part”. What does this app?It checks whether received mail or message spam or not.

In the bellow part, there are codes that are written in the jupyter notebook and then, there is the explanation of the code.



```
In [7]: import pandas as pd
        import numpy as np
        import pickle
        import re
        import nltk
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline

        from sklearn.metrics import accuracy_score,fbeta_score,classification_report
        from wordcloud import WordCloud
        from nltk.tokenize import word_tokenize

        from nltk.corpus import stopwords
        nltk.download('stopwords')
        stop=stopwords.words("english")

        from nltk.stem.porter import PorterStemmer
        from nltk.stem import SnowballStemmer
        ss = SnowballStemmer("english")
        ps = PorterStemmer()

        msg_df = pd.read_csv('spam.csv', sep='\t', names=["label", "message"])
        msg_df.shape

        [nltk_data] Downloading package stopwords to
        [nltk_data]     C:\Users\Ranjan\AppData\Roaming\nltk_data...
        [nltk_data]   Unzipping corpora\stopwords.zip.

Out[7]: (5572, 2)
```

```
In [8]: stop

Out[8]: ['i',
         'me',
         'my',
         'myself',
         'we',
         'our',
         'ours',
         'ourselves',
         'you',
         "you're",
         "you've",
         "you'll",
         "you'd",
         'your',
         'yours',
         'yourself',
         'yourselves',
         'he',
         'him',
         'his'
```

```
In [9]: msg_df = pd.read_csv('spam.csv', sep='\t', names=["label", "message"])

        msg_df.shape
        msg_df.head(5)
```

Out[9]:

	label	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...

In [10]:
```
msg_df.describe()
```
Out[10]:

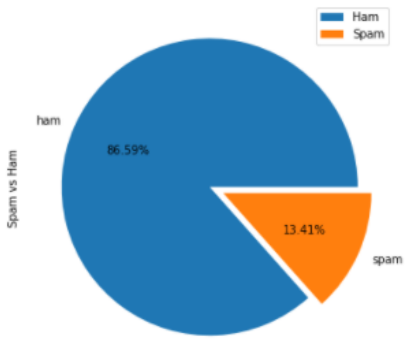
	label	message
count	5572	5572
unique	2	5169
top	ham	Sorry, I'll call later
freq	4825	30

In [11]:
```
msg_df.groupby('label').describe().T
```
Out[11]:

	label	ham	spam
message	count	4825	747
	unique	4516	653
	top	Sorry, I'll call later	Please call our customer service representativ...
	freq	30	4

In [12]:
```
msg_df["label"].value_counts()
```
Out[12]:
```
ham     4825
spam     747
Name: label, dtype: int64
```

In [13]:
```
msg_df["label"].value_counts().plot(kind = 'pie', explode = [0, 0.1], figsize = (6, 6), autopct = '%1.2f%%')
plt.ylabel("Spam vs Ham")
plt.legend(["Ham", "Spam"])
plt.show()
```



In [14]:
```
msg_df.groupby("message")["label"].agg([len, np.max]).sort_values(by = "len", ascending = False).head(n = 10)
```
Out[14]:

message	len	amax
Sorry, I'll call later	30	ham
I cant pick the phone right now. Pls send a message	12	ham
Ok...	10	ham
Ok	4	ham
Okie	4	ham
7 wonders in My WORLD 7th You 6th Ur style 5th Ur smile 4th Ur Personality 3rd Ur Nature 2nd Ur SMS and 1st "Ur Lovely Friendship"... good morning dear	4	ham
Wen ur lovable bcums angry wid u, dnt take it seriously.. Coz being angry is d most childish n true way of showing deep affection, care n luv!.. kettoda manda... Have nice day da.	4	ham
Your opinion about me? 1. Over 2. Jada 3. Kusruthi 4. Lovable 5. Silent 6. Spl character 7. Not matured 8. Stylish 9. Simple Pls reply..	4	ham
Please call our customer service representative on FREEPHONE 0808 145 4742 between 9am-11pm as you have WON a guaranteed £1000 cash or £5000 prize!	4	spam
Ok.	4	ham

In [15]:
```
msg_df['length']=msg_df['message'].apply(len)
msg_df.head()
```

Out[15]:

	label	message	length
0	ham	Go until jurong point, crazy.. Available only ...	111
1	ham	Ok lar... Joking wif u oni...	29
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	ham	U dun say so early hor... U c already then say...	49
4	ham	Nah I don't think he goes to usf, he lives aro...	61

In [16]:
```
msg_df.length.describe()
```

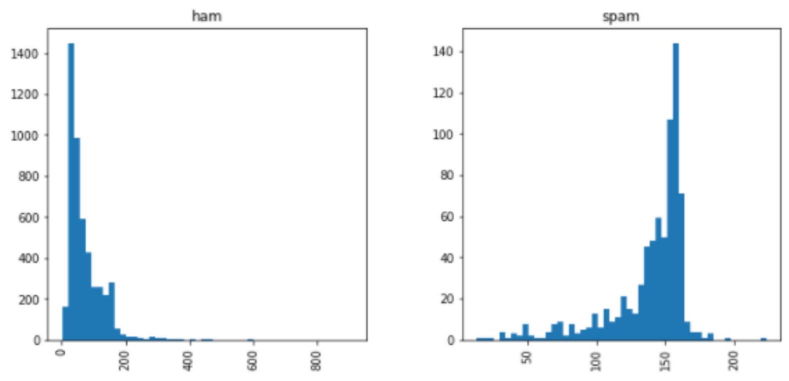
Out[16]:
```
count    5572.000000
mean       80.489950
std        59.942907
min         2.000000
25%        36.000000
50%        62.000000
75%       122.000000
max       910.000000
Name: length, dtype: float64
```

In [17]:
```
msg_df[msg_df['length']==910]['message'].iloc[0]
```

Out[17]:
```
"For me the love should start with attraction.i should feel that I need her every time around me.she should be the first thing
which comes in my thoughts.I would start the day and end it with her.she should be there every time I dream.love will be then w
hen my every breath has her name.my life should happen around her.my life will be named to her.I would cry for her.will give al
l my happiness and take all her sorrows.I will be ready to fight with anyone for her.I will be in love when I will be doing the
craziest things for her.love will be when I don't have to proove anyone that my girl is the most beautiful lady on the whole pl
anet.I will always be singing praises for her.love will be when I start up making chicken curry and end up makiing sambar.life
will be the most beautiful then.will get every morning and thank god for the day because she is with me.I would like to say a l
ot..will tell later.."
```

In [18]:
```
msg_df.hist(column='length', by='label', bins=50,figsize=(11,5))
```

Out[18]:
```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x000001DE3A83AE80>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000001DE3A8AD0F0>],
      dtype=object)
```



Looks like the lengthy is the message, more likely it is a spam. Let's not forget this

Text Transformation

Data Cleaning (Removing unimportant data/ Stopwords/ Stemming)

In [19]:
```
msg_df.head(4)
```

Out[19]:

	label	message	length
0	ham	Go until jurong point, crazy.. Available only ...	111
1	ham	Ok lar... Joking wif u oni...	29
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	ham	U dun say so early hor... U c already then say...	49

In [20]:
```
import string
def cleanText(message):
    #message = message.translate(str.maketrans('ranjan', 'ranjan', string.punctuation))
    message = re.sub('[^a-zA-Z]', ' ', message)
    message = message.lower()
    message = message.split()
    words = [ss.stem(word) for word in message if word not in stop]
    return " ".join(words)

msg_df["message"] = msg_df["message"].apply(cleanText)
msg_df.head(n = 10)
```

Out[20]:

	label	message	length
0	ham	go jurong point crazi avail bugi n great world...	111
1	ham	ok lar joke wif u oni	29
2	spam	free entri wkli comp win fa cup final tkts st ...	155
3	ham	u dun say earli hor u c alreadi say	49
4	ham	nah think goe usf live around though	61
5	spam	freemsg hey darl week word back like fun still...	147
6	ham	even brother like speak treat like aid patent	77
7	ham	per request mell mell oru minnaminungint nurun...	160
8	spam	winner valu network custom select receivea pri...	157
9	spam	mobil month u r entitl updat latest colour mob...	154

In [21]:
```
spam_messages = msg_df[msg_df["label"] == "spam"]["message"]
ham_messages = msg_df[msg_df["label"] == "ham"]["message"]
```

In [22]:
```
spam_messages
```

Out[22]:
```
2       free entri wkli comp win fa cup final tkts st ...
5       freemsg hey darl week word back like fun still...
8       winner valu network custom select receivea pri...
9       mobil month u r entitl updat latest colour mob...
11      six chanc win cash pound txt csh send cost p d...
                              ...
5537    want explicit sex sec ring cost p min gsex pob...
5540    ask mobil chatlin inclu free min india cust se...
5547    contract mobil mnths latest motorola nokia etc...
5566    remind get pound free call credit detail great...
5567    nd time tri contact u u pound prize claim easi...
Name: message, Length: 747, dtype: object
```

In [26]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Ranjan\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

Out[26]:
```
True
```

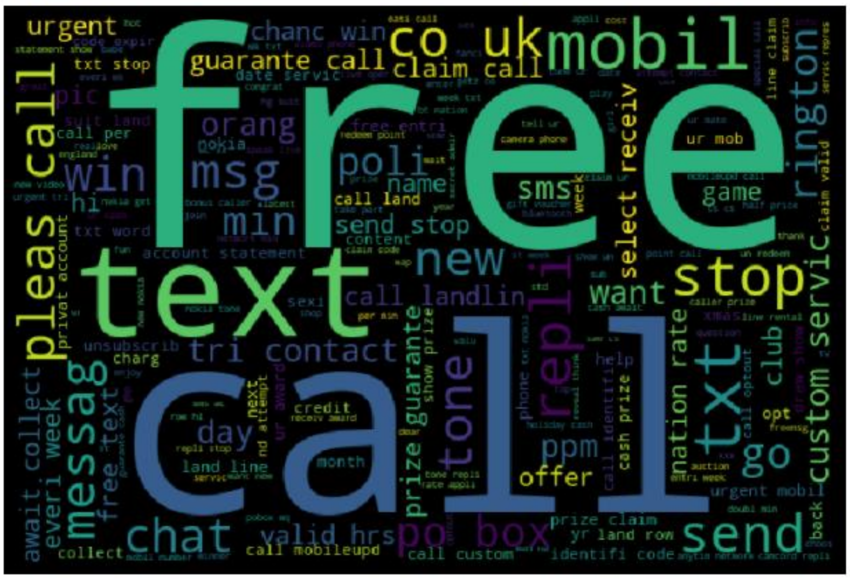
In [27]:
```
spam_words = []
ham_words = []

def extractSpamWords(spamMessages):
    global spam_words
    words = [word for word in word_tokenize(spamMessages)]
    spam_words = spam_words + words

def extractHamWords(hamMessages):
    global ham_words
    words = [word for word in word_tokenize(hamMessages) ]
    ham_words = ham_words + words

spam_messages.apply(extractSpamWords)
ham_messages.apply(extractHamWords)
```

Out[27]:
```
0       None
1       None
3       None
4       None
6       None
        ...
5565    None
5568    None
5569    None
5570    None
5571    None
Name: message, Length: 4825, dtype: object
```

In [28]:
```
ham_words
```

Out[28]:
```
['go',
 'jurong',
 'point',
 'crazi',
 'avail',
 'bugi',
 'n',
 'great',
 'world',
```

In [29]:
```python
spam_wordcloud = WordCloud(width=600, height=400).generate(" ".join(spam_words))
plt.figure( figsize=(10,8), facecolor='k')
plt.imshow(spam_wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)
plt.show()
```



In [30]:
```python
ham_wordcloud = WordCloud(width=600, height=400).generate(" ".join(ham_words))
plt.figure( figsize=(10,8), facecolor='k')
plt.imshow(ham_wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)
plt.show()
```

In [31]:
```python
msg_df
```

Out[31]:

	label	message	length
0	ham	go jurong point crazi avail bugi n great world...	111
1	ham	ok lar joke wif u oni	29
2	spam	free entri wkli comp win fa cup final tkts st ...	155
3	ham	u dun say earli hor u c alreadi say	49
4	ham	nah think goe usf live around though	61
...	...	...	...
5567	spam	nd time tri contact u u pound prize claim easi...	160
5568	ham	b go esplanad fr home	36
5569	ham	piti mood suggest	57
5570	ham	guy bitch act like interest buy someth els nex...	125
5571	ham	rofl true name	26

5572 rows × 3 columns

In [32]:
```python
def encodeCategory(cat):
    if cat == "spam":
        return 1
    else:
        return 0

msg_df["label"] = msg_df["label"].apply(encodeCategory)
```

In [33]:
```python
msg_df
```

Out[33]:

	label	message	length
0	0	go jurong point crazi avail bugi n great world...	111
1	0	ok lar joke wif u oni	29
2	1	free entri wkli comp win fa cup final tkts st	155
5570	0	guy bitch act like interest buy someth els nex...	125
5571	0	rofl true name	26

5572 rows × 3 columns

In [34]:
```python
from sklearn.feature_extraction.text import TfidfVectorizer
vec = TfidfVectorizer(encoding = "latin-1", strip_accents = "unicode")
features = vec.fit_transform(msg_df["message"])
print(features.shape)
```

```
(5572, 6292)
```

In [35]:
```python
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
X=cv.fit_transform(msg_df["message"])
print (X.shape)
```

```
(5572, 6292)
```

In [36]:
```python
cv = CountVectorizer()

X=cv.fit(msg_df["message"])
X.vocabulary_
X.get_feature_names()
```

Out[36]:
```
['aa',
 'aah',
 'aaniy',
 'aaooooright',
 'aathi',
```

In [37]:
```python
X = cv.fit_transform(msg_df["message"]).toarray()
X
```

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

In [38]:
```python
df = pd.DataFrame(X,columns=cv.get_feature_names())
df
df['len']=msg_df['length']
df
```

Out[38]:

	aa	aah	aaniy	aaooooright	aathi	ab	abbey	abdomen	abeg	abel	...	zf	zhong	zindgi	zoe	zogtorius	zoom	zouk	zs	zyada	len
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	111
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	29
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	155
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	49
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	61
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5567	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	160
5568	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	36
5569	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	57
5570	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	125
5571	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	26

5572 rows × 6293 columns

In [39]:
```python
df
```

Out[39]:

	aa	aah	aaniy	aaooooright	aathi	ab	abbey	abdomen	abeg	abel	...	zf	zhong	zindgi	zoe	zogtorius	zoom	zouk	zs	zyada	len
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	111
1	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	29
5571	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	26

5572 rows × 6293 columns

In [40]:
```python
print(X)
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

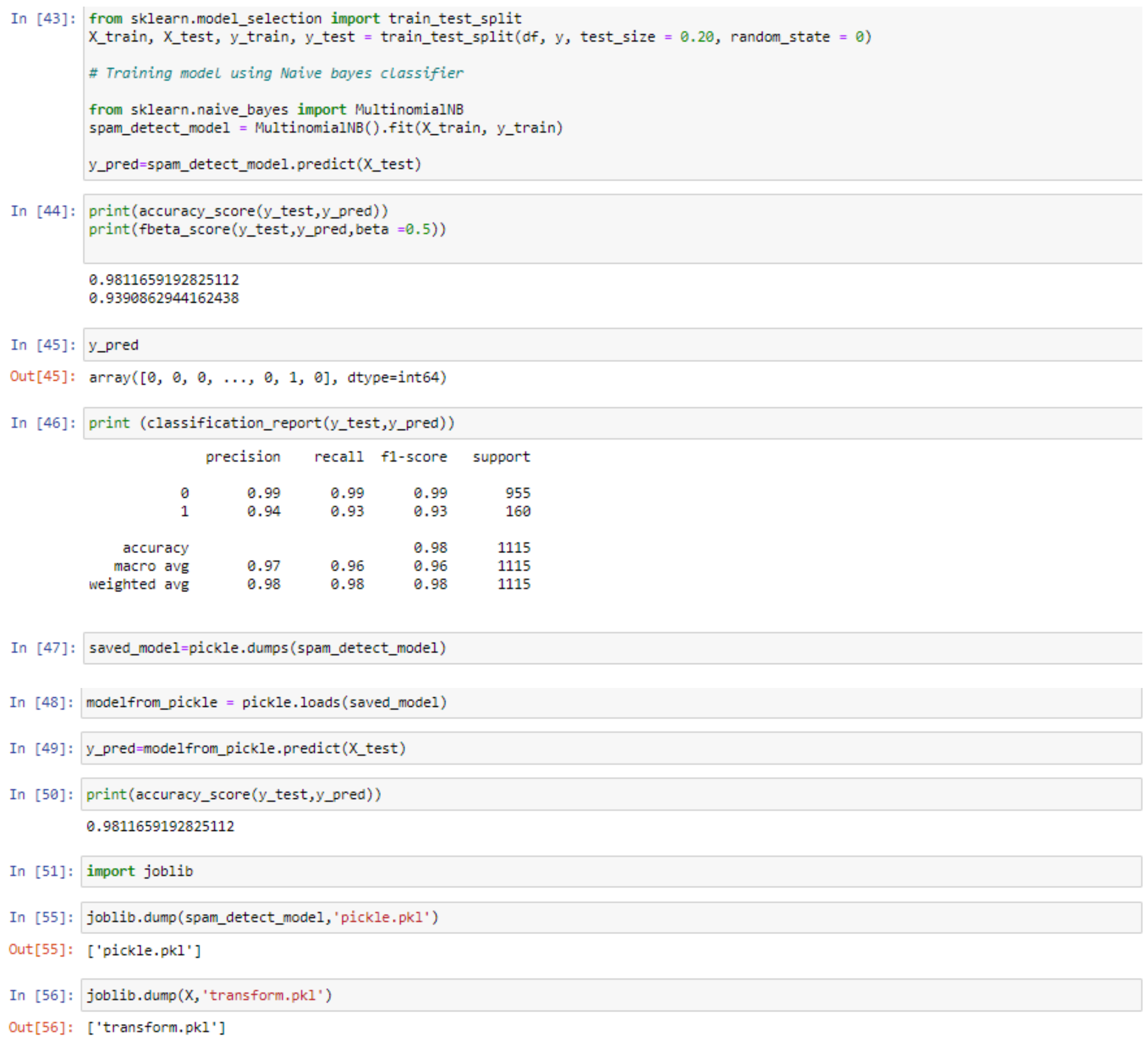
In [41]:
```python
df.head()
```

Out[41]:

	aa	aah	aaniy	aaooooright	aathi	ab	abbey	abdomen	abeg	abel	...	zf	zhong	zindgi	zoe	zogtorius	zoom	zouk	zs	zyada	len
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	111
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	29
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	155
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	49
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	61

5 rows × 6293 columns

In [42]:
```python
y=msg_df['label']
```



```
In [43]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(df, y, test_size = 0.20, random_state = 0)

         # Training model using Naive bayes classifier

         from sklearn.naive_bayes import MultinomialNB
         spam_detect_model = MultinomialNB().fit(X_train, y_train)

         y_pred=spam_detect_model.predict(X_test)

In [44]: print(accuracy_score(y_test,y_pred))
         print(fbeta_score(y_test,y_pred,beta =0.5))

         0.9811659192825112
         0.9390862944162438

In [45]: y_pred
Out[45]: array([0, 0, 0, ..., 0, 1, 0], dtype=int64)

In [46]: print (classification_report(y_test,y_pred))

                       precision    recall  f1-score   support

                    0       0.99      0.99      0.99       955
                    1       0.94      0.93      0.93       160

             accuracy                           0.98      1115
            macro avg       0.97      0.96      0.96      1115
         weighted avg       0.98      0.98      0.98      1115

In [47]: saved_model=pickle.dumps(spam_detect_model)

In [48]: modelfrom_pickle = pickle.loads(saved_model)

In [49]: y_pred=modelfrom_pickle.predict(X_test)

In [50]: print(accuracy_score(y_test,y_pred))

         0.9811659192825112

In [51]: import joblib

In [55]: joblib.dump(spam_detect_model,'pickle.pkl')
Out[55]: ['pickle.pkl']

In [56]: joblib.dump(X,'transform.pkl')
Out[56]: ['transform.pkl']
```

Firstly in the code, necessary libraries in python have imported such as pandas, numpy, pickle(which he will create a pickle file), re (regular expressions), nltk (natural language toolkit), matplotlib.pylot (is a collection of fuctions), seaborn (data visualization library based on matplotlib.)

Then,the Accuracy score, fbeta score and classification score, they are all form the classification metrics from the sklearn.metrics.

Wordcloud imported from Word cloud which is gonna be created in the folloving sections.

word tokenizer imported from NLTK and its used for to converting dataframe for better text understanding in machine learning applications.

Stopwords basically; words that can be safely ignored withouth sacrifacing the meaning of the sentence for eg. I, We, You, They etc. and imported from nltk.corpus. If it is the first time of using this library it needs to be downloaded and then stop variable have been created and its gonna be used for storing all the stop words in english.

After that, PorterStemmer & SnowballStemmer are imported and they are basicaly extracting the words into the base/stem form(for eg.converting 3 similar words into one eg. hard, hardest and harder to hard).

new dataframe has created with the name of the msg_df and spam.csv File has been imported(and this file consist of spam and ham messages) and then the columns have named as label (for spam/ham) and message.

The file size is five thousan five hundred seventy two with 2 columns.

These are the stop words.

Then the first five files are shown in the graph format. After that under the label and message columns, count, unique, top and freq have described.

Caunt is the total, unique has two labels which are spam & ham and unique message number is 5169, The majority consists of the ham category and the message is “Sorry. I’ll call you later.”. Since the gap between ham & sapm so big such as the number of ham is 4825 and spam is 747 then the pie chart with pyplot has been made in order to see in visualized form.

Since here it has 2 labels and now, the third label which is length has been added. The length label basicly counts the words in the message. After that, the first five messages have been shown with length. Then, length instances have been shown, such as total message, mean, standard deviation, min, max, etc and we can clearly see that, the max message length is 910. Since the message length is so big, that message has been checked whether spam or not.

After that, the relationship has been checked whether is there any relationship between label and length. According to the chart the high message length indicates that it is tend to be spam.

After this part, Text Transformation part comes, it is basically data cleaning/removing of the unimportant data.

In the cleantext function part, if the message has other than the alphabet, ıt is replaced with space and then all the characters turn the lowercase form then they split into words. After that for the list comprehension snowballstemmer is used and then it is returning splitted words together again and the cleantext function is being applied for all the messages. After that messages are compared with before and after.

For the creating Wordcloud, firstly spam and ham messages are seperated with addign filter. After that 2 variables have been created for which words are gonna come in spam messages and ham messages.

Then words blank lists are gonna be created what are the words that are the coming in the spam and ham messages.

ExtractSpamWords & extractHamWords are the two fuctions that extracts the words and then spam&ham words have been referenced in order to use in this function and then list comprehension created with tokenize technique, its basicly converts paragraphs into sentences and sentences to the words then they are conctenated. these functions applied all the spam messages and ham messages seperately.

Wordcloud is created with giving a name as spam. wordcloud for spam messages, and it is also done for ham messages too with giving a name as ham. Then to be able to show which words are mostly used in a sentence, pyplot is used for showing the image.

For the encoding category if else statement is used. If it is spam it returns 1 or returns 0.

Machines only understand from the numbers so the texts should be converted into numerical forms so in this part the text vectorization nlp comes. With the counter vectorizer , it counts the number of messages and the number of unique words. In the next step all the unique words in the sentences are shown with index numbers in the array structure . To be able to use array it must be converted into dataframe structure in order to do that pandas used. Then the new variable 'len' which is refers to length is created.

For the output, y variable is created with using label.Firstly Test and Train data are splitted and then 80% for the train data and 20% for the test data used in this model applying part.

Then The Naive Bayes classifier's MultinominalNB is used for to train the model. Then this model took the name of spam_detect_model and once it has trained the predict function can be used.

After that the y_pred is the predicted value, its found by spam_detect_model with predict(x_test) data. It has predicted so the accuracy can be found from this point.

The dataset which is consists of spam and ham messages are unbalanced, so fbeta_score is used with the rate of 0.5.

The classification report is also used in here due to unbalanced dataset. F1-score is the harmonic avg of the precision and recall ,and the support is the number of instances of the actual answer found in that class.

From this step model have to be deployed ,and pickle file have to be created and the knowledge of the model has to be created in order to use whole dataset in the web server and the knowledge have to be transformed into pickle file.

Saved_model is the name of the defined saved model and with the dumps function, objects are stored in a file. Then, the first thing is to check whether saved model is true or not so new model which is modelfrom_pickle is created and for the knowledge, saved_model has been used. Then for the modelfrom_pickle's prediction, x_test model is applied.

Finally for the saving pickle file, joblib has imported then again dumps function is used for storing in a file which is pickle, and the used model is that spam_detect_model.

The model (learning framework) has been created, for the heroku cloud part, pickle file and app.py which is an instance of the Flask object has to be deployed. App.py will act as the central configuration object for the entire application.

and for the surface of the app page, HTML & CSS gonna be used.

The final form of the app link: https://spam-detectionn.herokuapp.com/

REFERENCES

Expert.ai Team, (May, 2020). What is Machine Learning? A Definition. Retrieved May 15, 2020 from https://www.expert.ai/blog/machine-learning-definition/

Lexalytics, (September, 2020). Machine Learning (ML) for Natural Language Processing (NLP). Retrieved May 15, 2020 from https://www.lexalytics.com/lexablog/machine-learning-natural-language-processing

Sharma, R. (July, 2020). Spam classification machine learning | End to End Deployment Part – 1&2 | Spam Classifier using Python. Retrieved May 15, 2020 from
https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbWZxVXZxLS04T1kwN2ctQ2pYOFAzaWlCcFR2UXxBQ3Jtc0trbDc3RDVCYUNQWlowNXk0dllOdUdzSlBQemNGMVFWQjVqWFhnUTJaSHBDR0pMdWNQbXp0ellweEZReXM5bGMtZzdDMWhBNmw5M25xaTQyQkNhRlRjZ0FyX0tqQy0xNEFhZmlqQUROMlpXRFl0OHB5TQ&q=https%3A%2F%2Fdrive.google.com%2Fdrive%2Ffolders%2F1Cz7kAIyx-1JSIdM5fy_hiE75LPR66SjF