



Email-Spam Detection

Submitted by:

Vandana Jain

Internship 10

ACKNOWLEDGMENT

The internship opportunity I had with FlipRobo was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me through this project period.

I would like to thank our SME for suggesting this project and for his whole hearted cooperation and constant encouragement throughout the project.

INTRODUCTION

- **Business Problem Framing**

In recent times, unwanted commercial bulk emails called spam has become a huge problem on the internet. The person sending the spam messages is referred to as the spammer. Such a person gathers email addresses from different websites, chatrooms, and viruses . Spam prevents the user from making full and good use of time, storage capacity and network bandwidth. The huge volume of spam mails flowing through the computer networks have destructive effects on the memory space of email servers, communication bandwidth, CPU power and user time . The menace of spam email is on the increase on yearly basis and is responsible for over 77% of the whole global email traffic [3]. Users who receive spam emails that they did not request find it very irritating. It is also resulted to untold financial loss to many users who have fallen victim of internet scams and other fraudulent practices of spammers who send emails pretending to be from reputable companies with the intention to persuade individuals to disclose sensitive personal information like passwords, Bank Verification Number (BVN) and credit card numbers.

Spam e-mails can be not only annoying but also dangerous to consumers. Spam e-mails can be defined as :

1. Anonymity
2. Mass Mailings
3. Unsolicited:

Spam e-mail are message randomly sent to multiple addressees by all sorts of groups, but mostly lazy advertisers and criminals who wish to lead you to phishing sites.

- **Motivation for the Problem Undertaken**

The upsurge in the volume of unwanted emails called spam has created an intense need for the development of more dependable and robust antispam filters. Machine learning methods of recent are being used to successfully detect and filter spam emails. Build a model which can be used to predict in terms of a probability for mail to be spam. In this case, Label '1' indicates that the email is spam , while, Label '0' indicates that the email is not spam.

- **PROBLEM STATEMENT**

- Unwanted e-mails irritating internet connection
- Critical e-mail message are missed and / or delayed.
- Millions of compromised computers
- Billions of dollars lost worldwide
- Identity theft
- Spam can crash mail servers and fill up hard drives

- **OBJECTIVE**

The objective of identification of Spam e-mails are :

- To give knowledge to the user about the fake e-mails and relevant e-mails
- To classify that mail spam or not.

- **SCOPE OF THE PROJECT:**

- It provides sensitivity to the client and adapts well to the future spam techniques.
- It considers a complete message instead of single words with respect to its organization.
- It increases Security and Control.
- It reduces IT Administration Costs.
- It also reduce Network Resource Costs.

Analytical Problem Framing

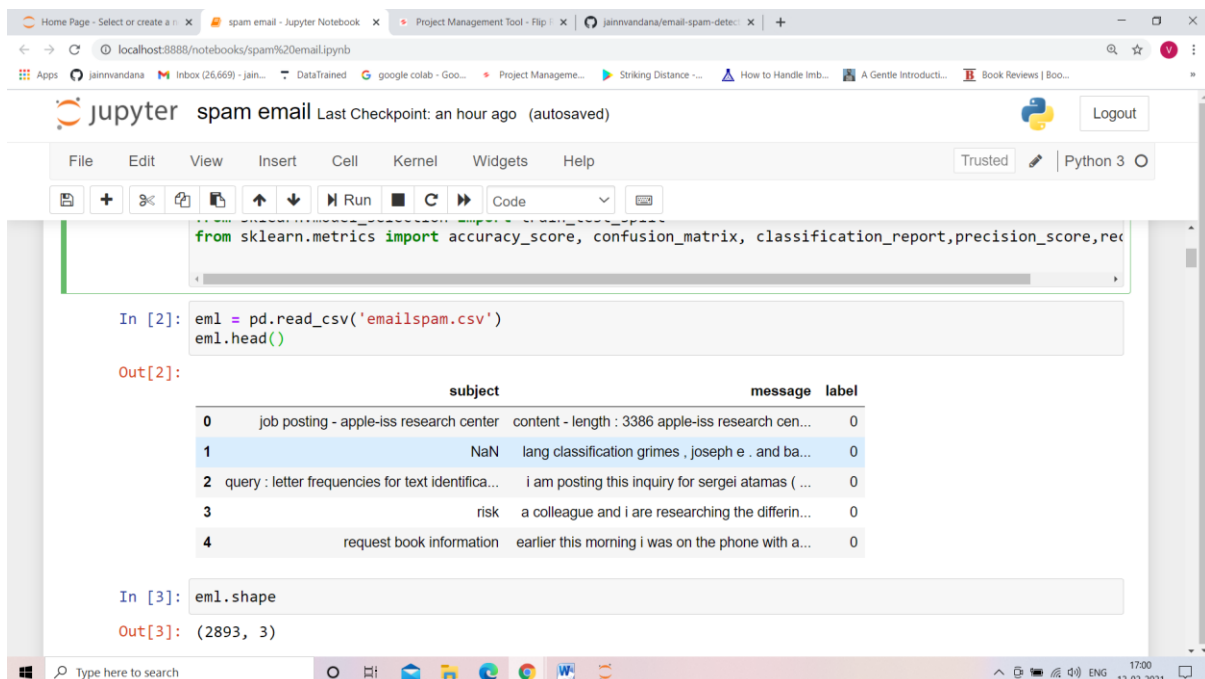
- Mathematical/ Analytical Modeling of the Problem

Machine Learning is defined by Tom Mitchell in his book as “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E ”. Supervised learning is when the output is known for the corresponding inputs, and is also provided for the machine to learn.

- EDA (Exploratory data analysis)
- Data Preprocessing
- Feature Extraction
- Scoring & Metrics

- Data Sources and their formats

The data is provided to us from our client database. It is hereby given to us for the exercise to improve the selection of mails for spam or not spam. It is given in the csv file format.



```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, precision_score, recall_score

In [2]: eml = pd.read_csv('emailspam.csv')
eml.head()

Out[2]:
```

	subject	message	label
0	job posting - apple-iss research center	content - length : 3386 apple-iss research cen...	0
1	NaN	lang classification grimes , joseph e . and ba...	0
2	query : letter frequencies for text identifica...	i am posting this inquiry for sergei atamas (...	0
3	risk	a colleague and i are researching the differin...	0
4	request book information	earlier this morning i was on the phone with a...	0

```
In [3]: eml.shape

Out[3]: (2893, 3)
```

- Data Preprocessing Done

The dataset that will be used to train the model has some challenges. Text Cleaning is a very important step in machine learning because your data may contains a lot of noise and unwanted character such as punctuation, white space, numbers, hyperlink and etc.

Some standard procedures are:

- convert all letters to lower/upper case
- removing numbers
- removing punctuation
- removing white spaces
- removing hyperlink
- removing stop words such as *a, about, above, down, doing* and the list goes on... Sometimes, the extremely common word which would appear to be of very little value in helping select documents matching user need are excluded from the vocabulary entirely.
- Word Stemming: Stemming algorithms work by removing the end or the beginning of the words, using a list of common prefixes and suffixes that can be found in that language.
- Word lemmatization : Lemmatization is utilizing the dictionary of a particular language and tried to convert the words back to its base form. It will try to take into account of the meaning of the verbs and convert it back to the most suitable base form.

Home Page - Select or create a notebook | spam email - Jupyter Notebook | Project Management Tool - Flip | jainnvandana/email-spam-detector | +

localhost:8888/notebooks/spam%20email.ipynb

jupyter spam email Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

2893 rows x 4 columns

```
In [8]: eml['message'] = eml['message'].str.lower()

In [9]: # Replace email addresses with 'email'
eml['message'] = eml['message'].str.replace(r'^(.+@[\.\.]*[a-z]{2,})$', 'emailaddress')

# Replace URLs with 'webaddress'
eml['message'] = eml['message'].str.replace(r'^http://[a-zA-Z0-9\-\.\.]+\.[a-zA-Z]{2,3}/(\S*)?$', 'webaddress')

# Replace money symbols with 'moneysymb' (£ can be typed with ALT key + 156)
eml['message'] = eml['message'].str.replace(r'£|$', 'dollars')

# Replace 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumbr'
eml['message'] = eml['message'].str.replace(r'^(?[\d]{3})?[\s-]?[\d]{3}[\s-]?[\d]{4}$', 'phonenumbr')

# Replace numbers with 'numbr'
eml['message'] = eml['message'].str.replace(r'\d+(\.\d+)?', 'numbr')
```

Home Page - Select or create a notebook | spam email - Jupyter Notebook | Project Management Tool - Flip | jainnvandana/email-spam-detector | +

localhost:8888/notebooks/spam%20email.ipynb

jupyter spam email Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
eml['message'] = eml['message'].str.replace(r'\d+(\.\d+)?', 'numbr')

In [10]: # Remove punctuation
eml['message'] = eml['message'].str.replace(r'^[^\w\d\s]', ' ')

# Replace whitespace between teemlrms with a single space
eml['message'] = eml['message'].str.replace(r'^\s+', ' ')

# Remove Leading and trailing whitespace
eml['message'] = eml['message'].str.replace(r'^\s+|\s+?$', ' ')

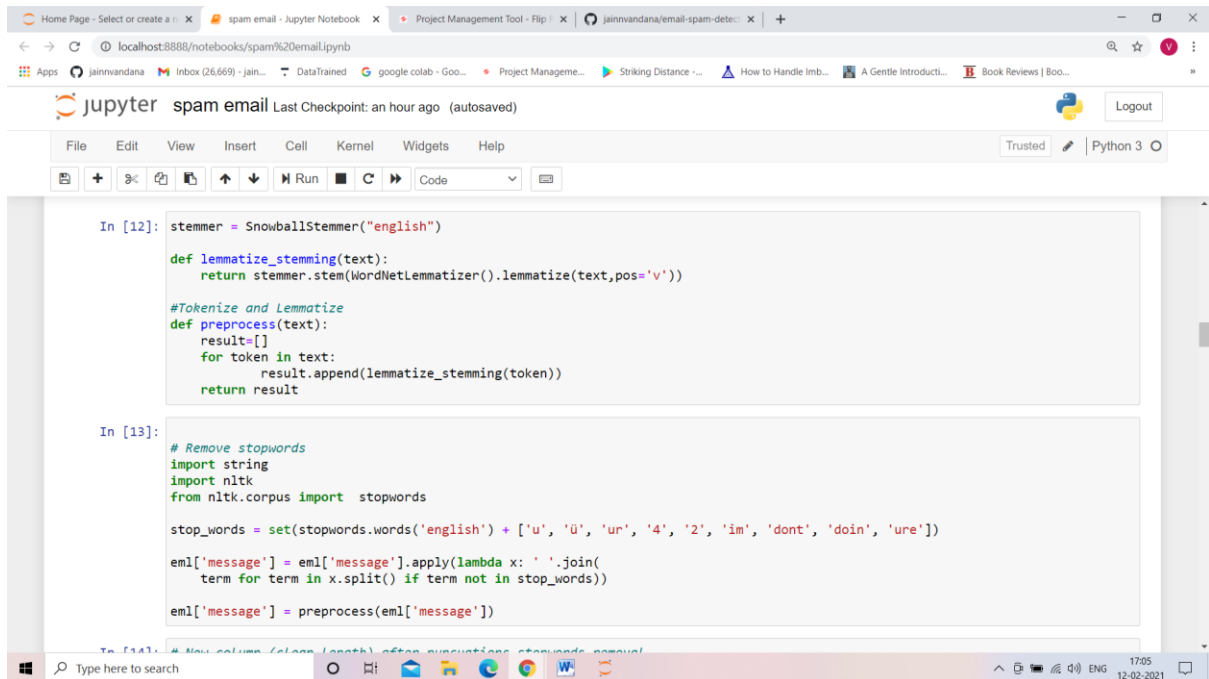
In [11]: eml.head()

Out[11]:
```

	subject	message	label	length
0	job posting - apple-iss research center	content length numbr apple iss research center...	0	2856
1	NaN	lang classification grimes joseph e and barbar...	0	1800
2	query : letter frequencies for text identifica...	i am posting this inquiry for sergei atamas sa...	0	1435
3	risk	a colleague and i are researching the differin...	0	324
4	request book information	earlier this morning i was on the phone with a...	0	1046

```
In [12]: stemmer = SnowballStemmer("english")

def lammatize_stemming(text):
```



```
In [12]: stemmer = SnowballStemmer("english")

def lemmatize_stemming(text):
    return stemmer.stem(WordNetLemmatizer().lemmatize(text,pos='v'))

#Tokenize and Lemmatize
def preprocess(text):
    result=[]
    for token in text:
        result.append(lemmatize_stemming(token))
    return result

In [13]: # Remove stopwords
import string
import nltk
from nltk.corpus import stopwords

stop_words = set(stopwords.words('english') + ['u', 'ü', 'un', '4', '2', 'im', 'dont', 'doin', 'ure'])

eml['message'] = eml['message'].apply(lambda x: ' '.join(
    term for term in x.split() if term not in stop_words))

eml['message'] = preprocess(eml['message'])

In [14]: # New column (len) after removing stopwords removal
```

- **Hardware and Software Requirements and Tools Used**

Hardware : Since the computational aspect of the project is of importance to PANDA, it is important to know the hardware that was used in the evaluation process. The training and evaluation of the neural network model has been done on a Windows 10 computer using a quad-core CPU at i3.

Software : anaconda 3 , windows 10 ,Microsoft office.

Tools used : python , machine learning libraries, Nltk, Nlp libraries.

The screenshot shows a Jupyter Notebook titled "spam email" running on a local host. The notebook has two input cells. The first cell, labeled "In [1]:", contains a block of Python code for importing various libraries: pandas, numpy, matplotlib.pyplot, seaborn, warnings, string, nltk, and sklearn. It also includes imports for stopwords, WordNetLemmatizer, SnowballStemmer, TfidfVectorizer, MultinomialNB, train_test_split, and accuracy_score, confusion_matrix, classification_report, precision_score, and recall_score. The second cell, labeled "In [2]:", contains code to read a CSV file named "emailspam.csv" and display the first five rows using the head() method. Below the second cell, the output is shown as a table with three columns: "subject", "message", and "label". The Windows taskbar is visible at the bottom of the screen.

```
In [1]: #Import Libs
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

import string
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer, SnowballStemmer

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, precision_score, recall_score

In [2]: eml = pd.read_csv('emailspam.csv')
eml.head()
```

subject	message	label
---------	---------	-------

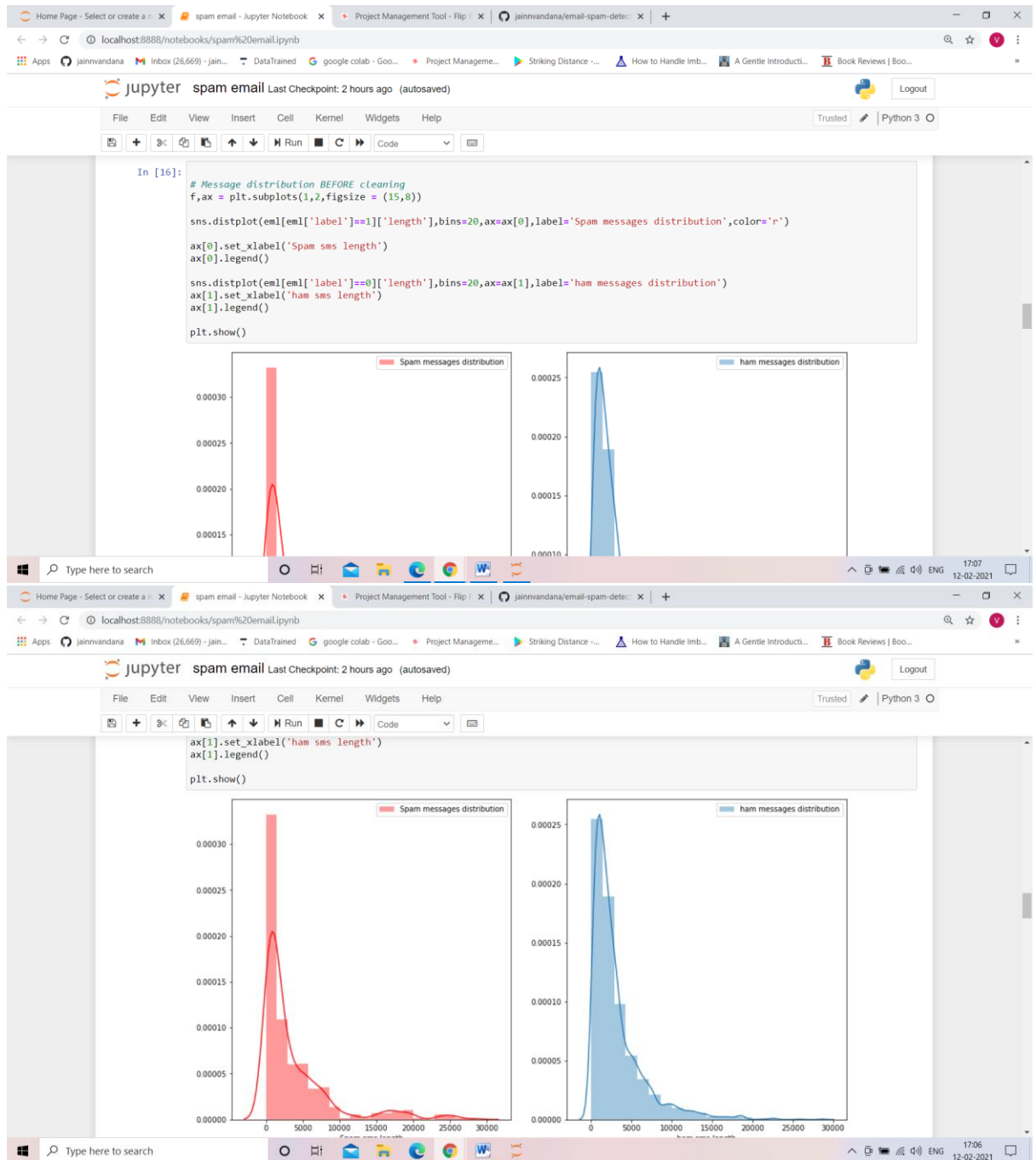
Model/s Development and Evaluation

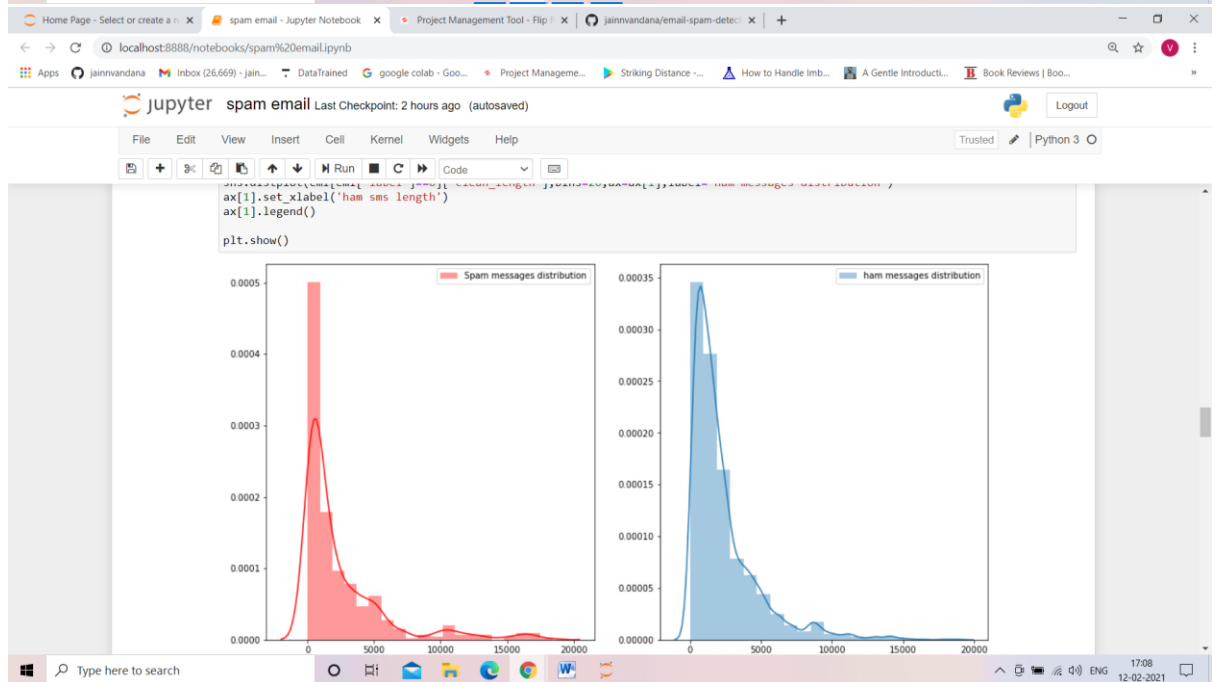
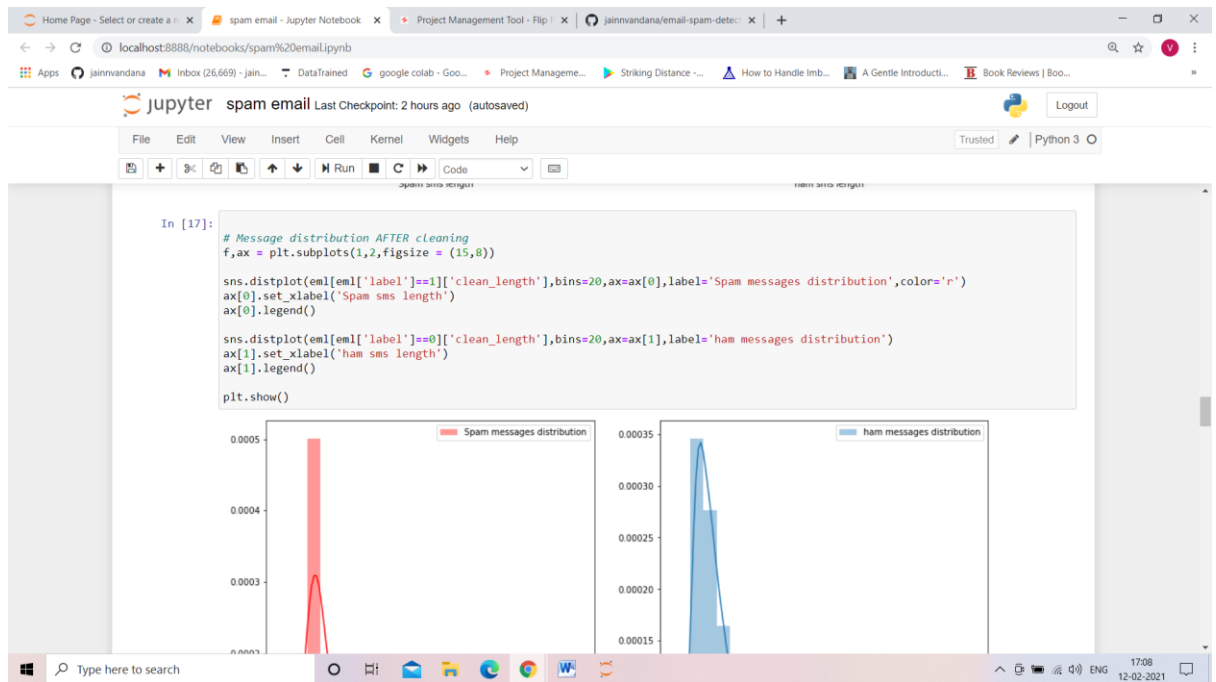
- Identification of possible problem-solving approaches (methods)

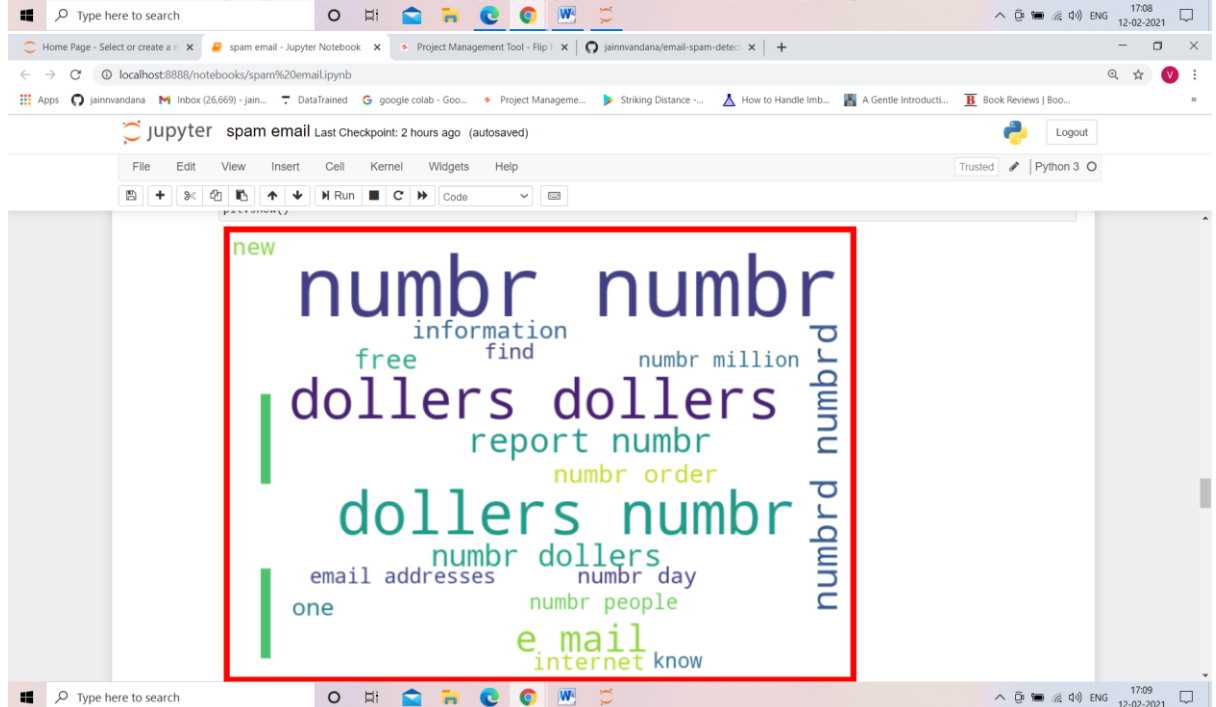
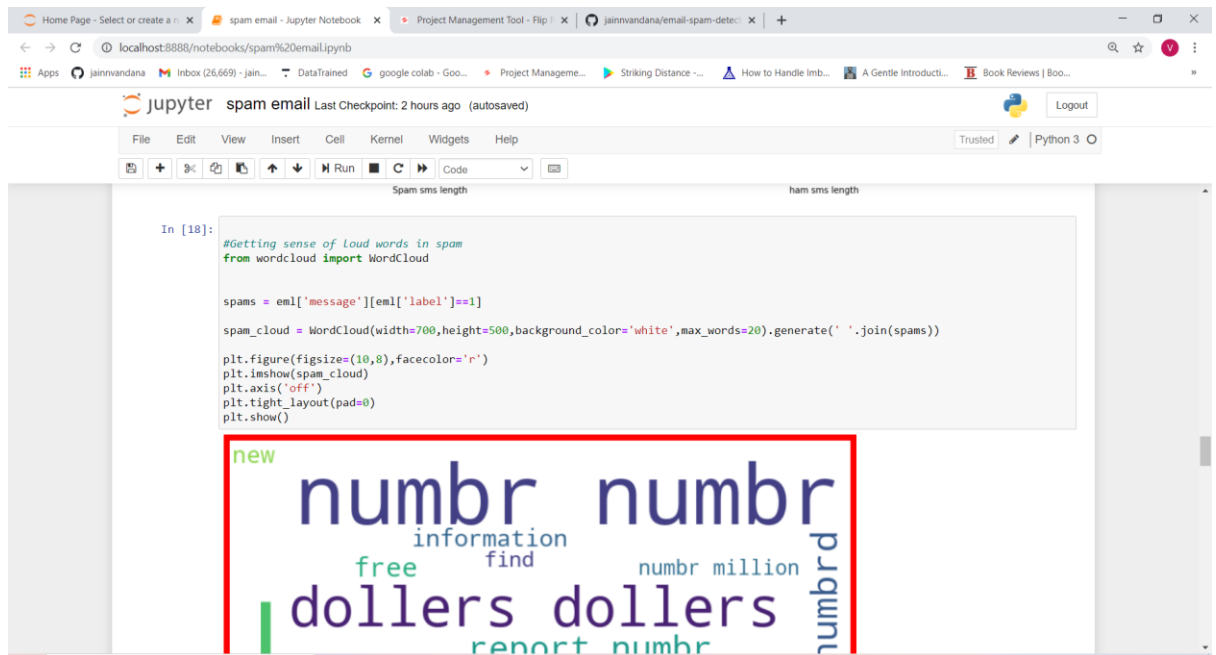
NAÏVE BAYS CLASSIFIER

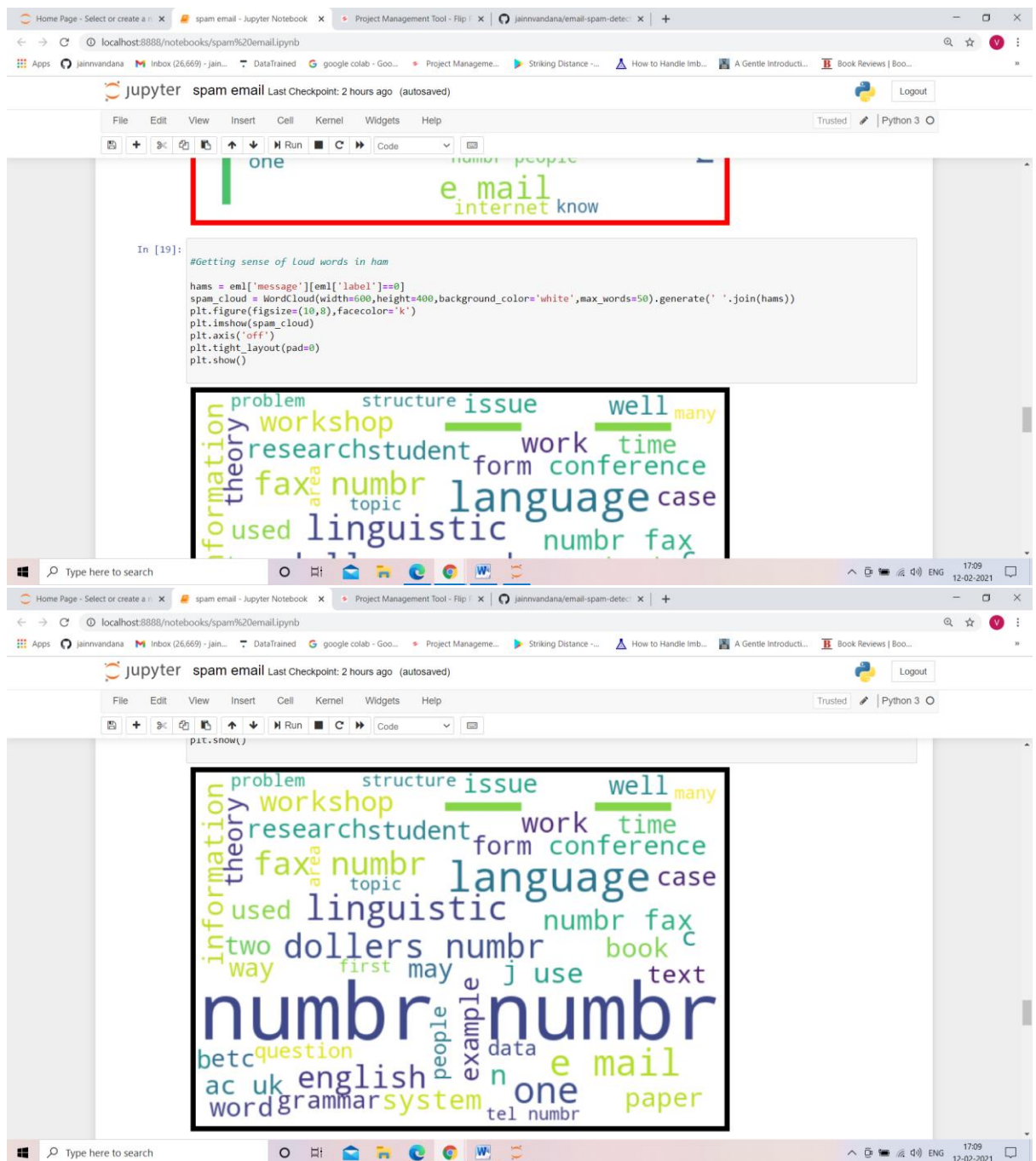
Simple probabilistic classifier that calculates a set of probabilities by counting the frequency and combination of values in a given dataset. Represent as a vector of feature values. It is very useful to classify the e-mails properly. The precision and recall of this method is known to be very effective.

- Visualizations









- Key Metrics for success in solving problem under consideration

When it comes to evaluation of a data science model's performance, sometimes accuracy may not be the best indicator. Some problems that we are solving in real life might have a very imbalanced class and using accuracy might not give us enough confidence to understand the algorithm's performance.

In the email spamming problem that we are trying to solve, the spam data is approximately 17% of our data. If our algorithm predicts all the email as non-spam, it will achieve an accuracy of 83%. And for some problem that has only 1% of positive data, predicting all the sample as negative will give them an accuracy of 99% but we all know this kind of model is useless in a real life scenario.

Precision & Recall is the common evaluation metrics that people use when they are evaluating class-imbalanced classification model. Let's try to understand what questions Precision & Recall is trying to answer,

- Precision: What proportion of positive identifications was actually correct ?
- Recall: What actual proportion of actual positives was identified correctly?

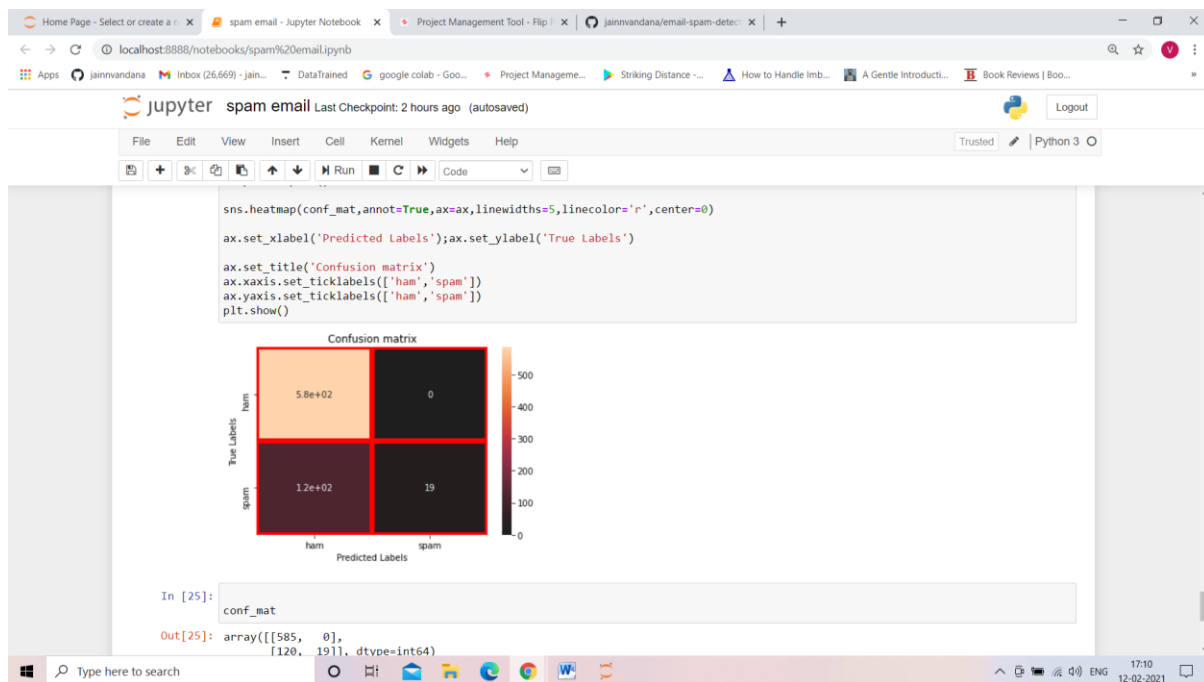
So, precision is evaluating, when a model predict something as positive, how accurate the model is. On the other hand, recall is evaluating how well a model in finding all the positive samples.

Confusion Matrix

Confusion Matrix is a very good way to understand results like true positive, false positive, true negative and so on.

Sklearn documentation has provided a sample code of how to plot nice looking confusion matrix to visualize your result..

Confusion Matrix of the result



CONCLUSION

Now that we have implemented the algorithm. Label '1' has approximately 13% records, while, label '0' has approximately 83% records. Let's have the results. We have implemented the dataset using Naive Bayes. We have the accuracy score of prediction of spam mails is 83%.