****

**EMAIL SPAM CLASSIFIER PROJECT**



Submitted by

**Raganee Verma**

**(**intern fliprobo **)**

**ACKNOWLEDGMENT**

I would like to express my special thanks to our mentor Ms. Khushboo Garg mam as well as Flip Robo Technologies who gave me the opportunity to do this project Email Spam Classification, which also helped me in doing lots of knowledge and research work. wherein I came to know about so many new things, especially the Natural Language Processing and Natural Language Toolkit parts.

I am also using a few external resources that helped me to complete this project and I learn from the samples and modify things according to my project requirement. The external resources, research paper and articles that were used in creating this project are listed belowFinally, I would want to convey my sincere thanks Datatrained Academy

**The website that I referred are**:

https://learning.datatrained.com

https://www.w3schools.com

https://medium.com/coders-camp

https://github.com

https://www.geeksforgeeks.org

https://www.javatpoint.com/nlp

https://www.educative.io/answers/preprocessing-steps-in-natural-

language-processing-nlp

https://www.youtube.com/watch?v=5ctbvkAMQO4

https://www.youtube.com/watch?v=X2vAabgKiuM

**INTRODUCTION**

● **Business Problem Framing**

Spam Filtering

Spam Detector is used to detect unwanted, malicious and virus

infected texts and helps to separate them from the nonspam texts.

It uses a binary type of classification containing the labels such as

‘ham’ (nonspam) and spam. Application of this can be seen in

Google Mail (GMAIL) where it segregates the spam emails in order

to prevent them from getting into the user’s inbox.

The SMS Spam Collection is a set of SMS tagged messages that have

been collected for SMS Spam research. It contains one set of SMS

messages in English of 5,574 messages, tagged according to ham

(legitimate) or spam.

● **Conceptual Background of the Domain Problem**

The main goal of the assignment is to show how you could design a

spam filtering system from scratch.

The files contain one message per line. Each line is composed by

two columns:

- v1 contains the label (ham or spam)

- v2 contains the raw text.

● **Motivation for the Problem Undertaken**

Implementing spam filtering is extremely important for any

organization. Not only does spam filtering help keep garbage out of

email inboxes, it helps with the quality of life of business emails

because they run smoothly and are only used for their desired

purpose. Spam filtering is essentially an anti-malware tool, as many

attacks through email are trying to trick users to click on a malicious

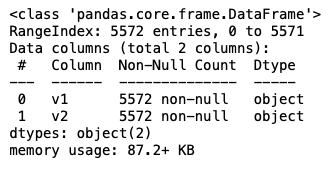
attachment, asking them to supply their credentials, and much

more.

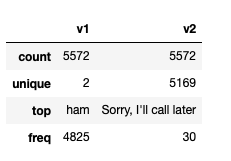
**Analytical Problem Framing**

**● Mathematical/ Analytical Modeling of the Problem**

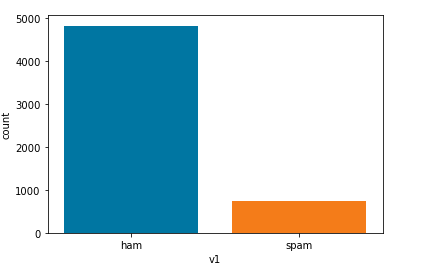
- Information of the dataset:



- Description of the dataset:



-Data visualization



**● Data Sources and their formats**

- A collection of 5573 rows of SMS spam messages was

manually extracted from the Grumbletext Web site. This is a

UK forum in which cell phone users make public claims about

SMS spam messages, most of them without reporting the

very spam message received. The identification of the text of

spam messages in the claims is a very hard and time-

consuming task, and it involves carefully scanning hundreds

of web pages.

- A subset of 3,375 SMS randomly chosen ham messages of the

NUS SMS Corpus (NSC), which is a dataset of about 10,000

legitimate messages collected for research at the Department

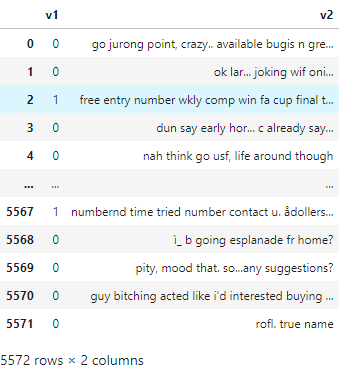
of Computer Science at the National University of Singapore.

The messages largely originate from Singaporeans and mostly

from students attending the University. These messages were

collected from volunteers who were made aware that their

contributions were going to be made publicly available.



● **Data Preprocessing Done**

In data pre-processing, I have done the various steps to clean the

dataset, as the dataset contains the comment that are in object

datatype, which cannot be read by the model, so before giving the

features to the model I had to convert that object datatype to

meaningful data and that can be understand by the model, so for

this I have used the NLP (Natural Processing Language).

“Natural language processing (NLP) refers to the branch of

computer science and more specifically, the branch of artificial

intelligence (AI) concerned with giving computers the ability to

understand text and spoken words in much the same way human

beings can.”

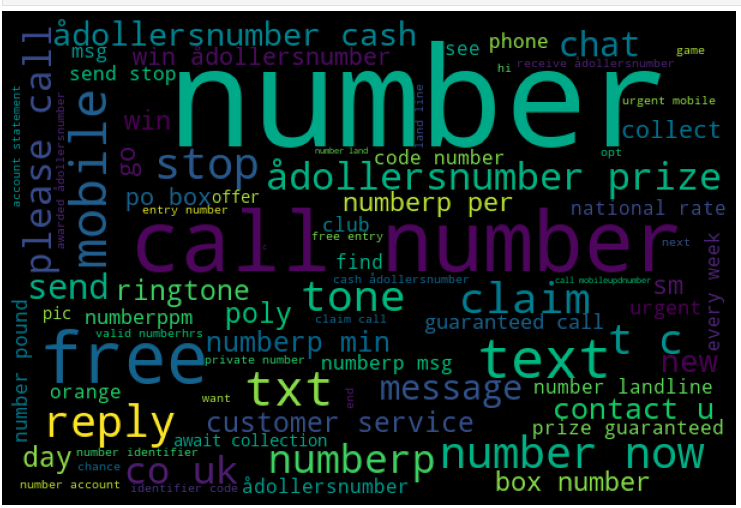






**WordCloud**

 A word cloud (also known as a tag cloud) is **a visual representation of words**. Cloud creators are used to highlight popular words and phrases based on frequency and relevance. They provide you with quick and simple visual insights that can lead to more in-depth analyses.

****

**● Data Inputs- Logic- Output Relationships**

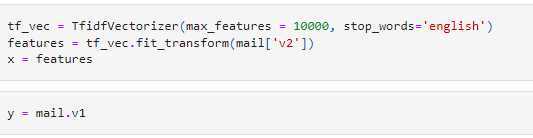
Used TF-IDF Vectorizer to encode the comments section.

“TfidfVectorizer is the base building block of many NLP pipelines. It

is a simple technique to vectorize text documents i.e. transform

sentences into arrays of numbers and use them in subsequent

tasks.”



**● Hardware and Software Requirements and Tools Used**

Anaconda-navigator

jupyter notebook

matplotlib-inline==0.1.6

numpy==1.23.2

packaging==21.3

pickleshare==0.7.5

platformdirs==2.5.2

prompt-toolkit==3.0.30

pyparsing==3.0.9

python-dateutil==2.8.2

scikit-learn==1.1.2

scipy==1.9.0

sklearn==0.05

NLP

Model/s Development and Evaluation

**● Identification of possible problem-solving approaches**

- EDA

- Description

- Visualization

- Data cleaning

- Data Pre-processing (NLP)

- Word Cloud

- Encoding

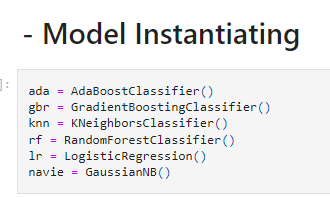
- Model Building

- Select the best model

- Cross-Validation

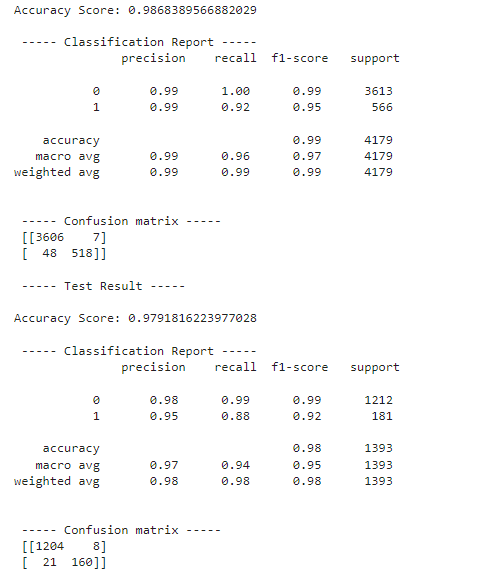
- Hyperparameter tuning

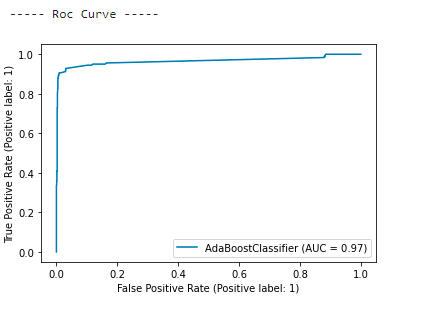




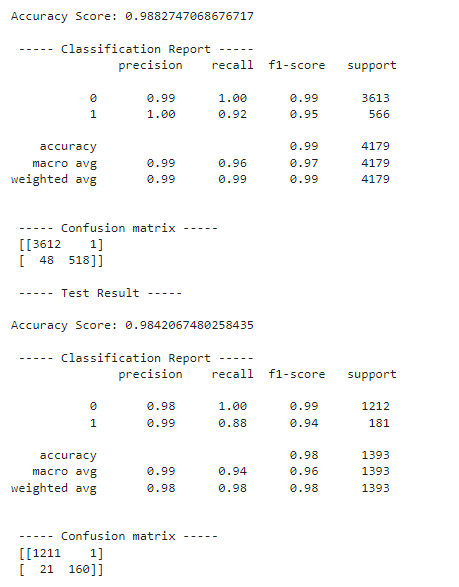
**● Run and Evaluate selected models**

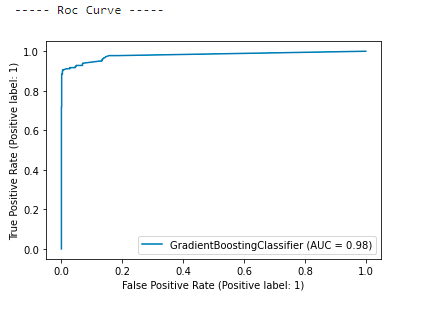
- AdaBoost Classifier



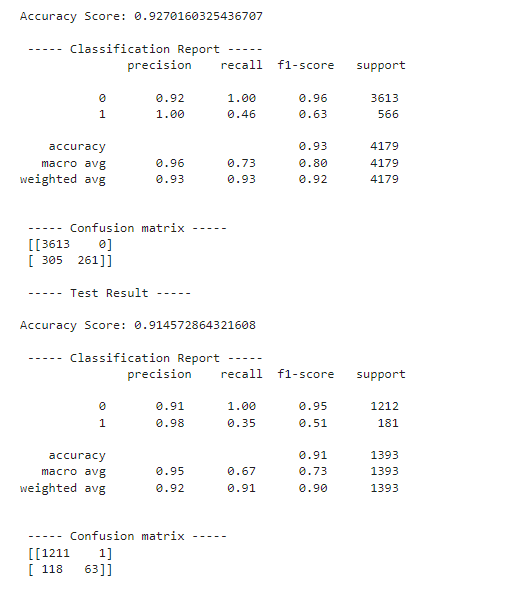


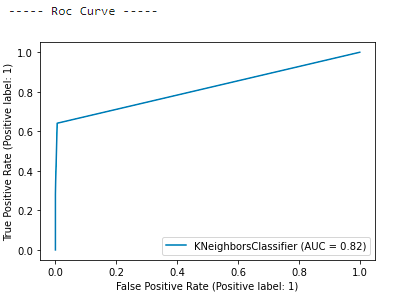
- GradientBoosting Classifier



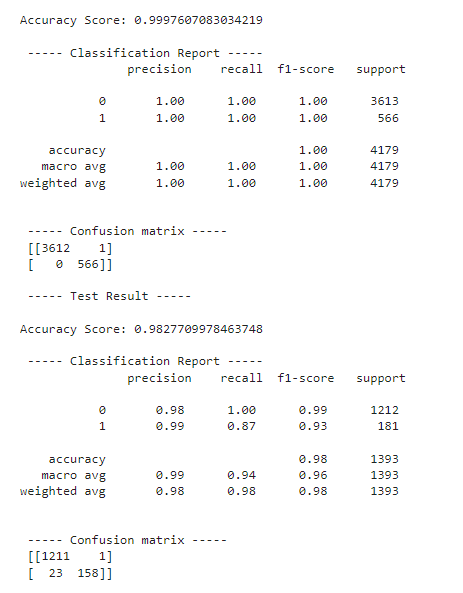


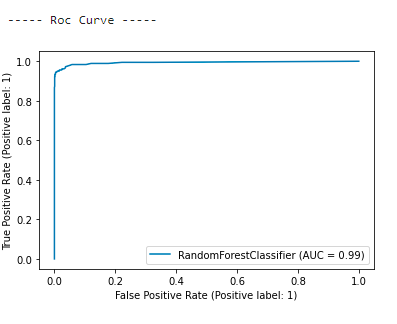
- KNeighbors Classifier



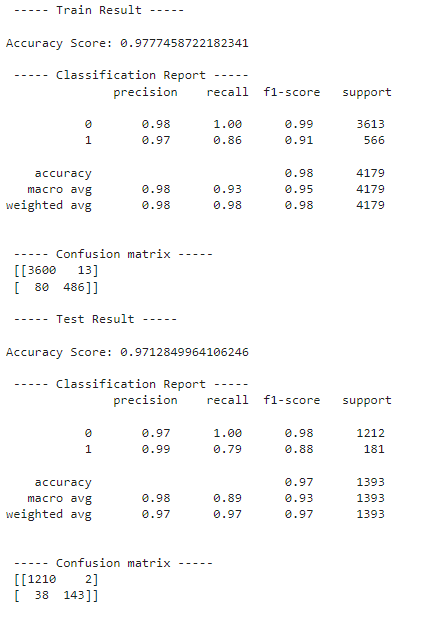


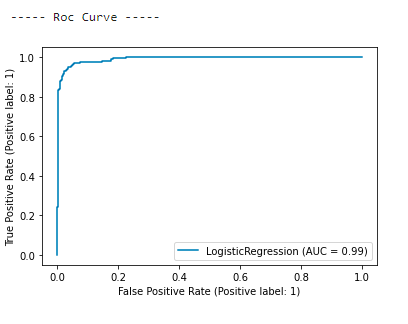
- RandomForest Classifier



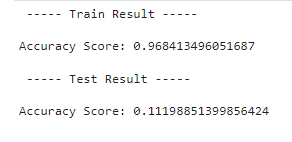


- Logistic Regression





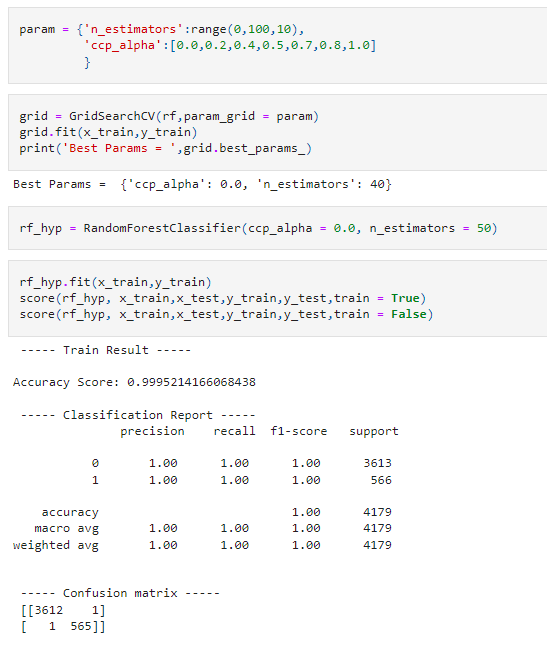
- Naive Bayes (GaussianNB)

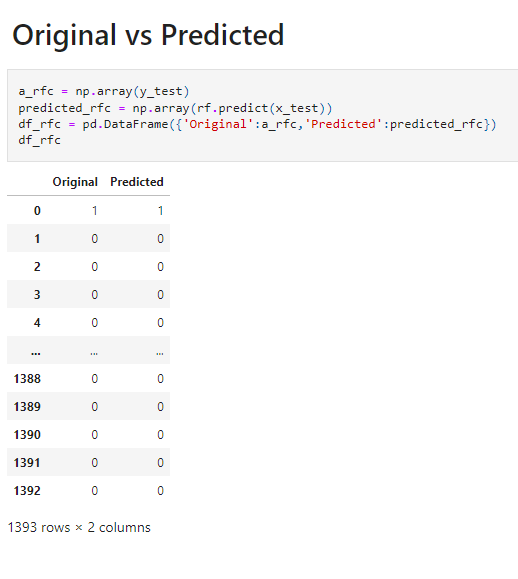


**● Interpretation of the Results**

RandomForest Classifier is giving the best result as compared to

Others. So we can perform hyper parameter tuning for this model.







**CONCLUSION**

● **Learning Outcomes of the Study in respect of DataScience**

Apply computing theory, languages, and algorithms, as well as

mathematical and statistical models, and the principles of

optimization to appropriately formulate and use data analyses.

Formulate and use appropriate models of data analysis to solve

hidden solutions to business-related challenges. Perform well in a

group.