"EMAIL SPAM CLASSIFICATION"

Submitted by:

Sarika Thorat

ACKNOWLEDGMENT

I would like to express my sincere thanks of gratitude to my SME as well as “Flip Robo Technologies” team for letting me work on “Email Spam Classifier” project also huge thanks to my academic team “Data Trained”. Their suggestions and directions have helped me in the completion of this project successfully. This project also helped me in doing lots of research wherein I came to know about so many new things.

Finally, I would like to thank my family and friends who have helped me with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

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 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.

Review of Literature

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:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   v1      5572 non-null   object
 1   v2      5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

- Description of the dataset:

	v1	v2
count	5572	5572
unique	2	5169
top	ham	Sorry, I'll call later
freq	4825	30

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.”

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

Hardware required: -

1. Processor — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

Software/s required: -

1.Anaconda

2.Jupyter Notebook

Libraries required:

To run the program and to build the model we need some basic libraries as follows

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import string
from wordcloud import WordCloud
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.ensemble import AdaBoostClassifier,GradientBoostingClassifier,RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import *
from sklearn.model_selection import KFold,cross_val_score

import pickle
import warnings
warnings.filterwarnings('ignore')
```

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

- EDA
- Description
- Visualization
- Data cleaning
- Data Pre-processing (NLP)
- Word Cloud
- Encoding
- Model Building
- Select the best model
- Cross-Validation
- Hyperparameter tuning

Testing of Identified Approaches (Algorithms)

Algorithms used for the training and testing:

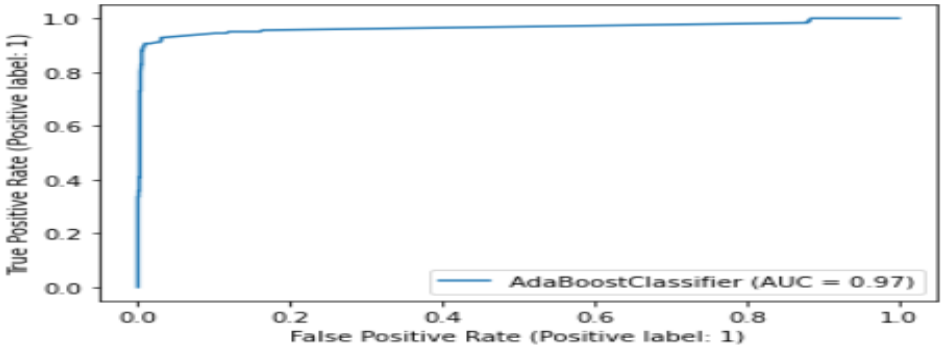
- AdaBoost Classifier
- GradientBoosting Classifier
- KNeighbors Classifier
- RandomForest Classifier
- Logistic Regression
- Naive Bayes (GaussianNB)

Run and Evaluate selected models

- AdaBoost Classifier

```
----- Train Result -----

Accuracy Score: 0.9868389566882029

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      3613
           1       0.99      0.92      0.95       566

    accuracy                           0.99      4179
   macro avg       0.99      0.96      0.97      4179
weighted avg       0.99      0.99      0.99      4179


----- Confusion matrix -----
[[3606    7]
 [  48  518]]

----- Test Result -----

Accuracy Score: 0.9791816223977028

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      1212
           1       0.95      0.88      0.92       181

    accuracy                           0.98      1393
   macro avg       0.97      0.94      0.95      1393
weighted avg       0.98      0.98      0.98      1393


----- Confusion matrix -----
[[1204    8]
 [  21  160]]
```

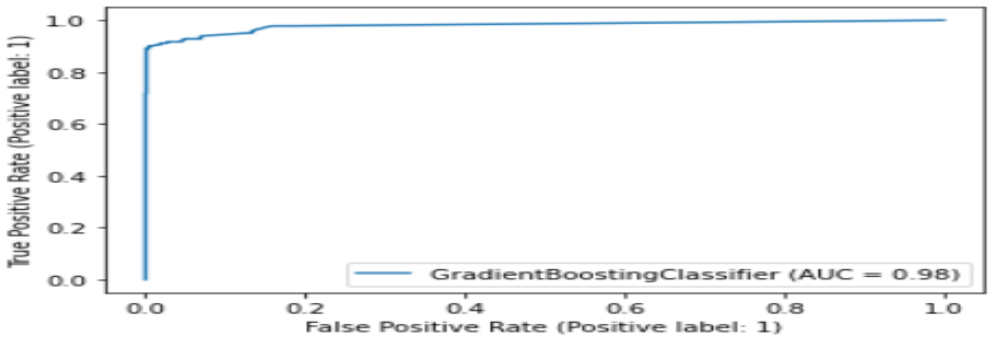
----- Roc Curve -----



- GradientBoosting Classifier

```
----- Train Result -----

Accuracy Score: 0.9882747068676717

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      3613
           1       1.00      0.92      0.95       566

    accuracy                           0.99      4179
   macro avg       0.99      0.96      0.97      4179
weighted avg       0.99      0.99      0.99      4179


----- Confusion matrix -----
[[3612    1]
 [  48  518]]

----- Test Result -----

Accuracy Score: 0.9849246231155779

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1212
           1       0.99      0.89      0.94       181

    accuracy                           0.98      1393
   macro avg       0.99      0.94      0.97      1393
weighted avg       0.99      0.98      0.98      1393


----- Confusion matrix -----
[[1211    1]
 [  20  161]]
```

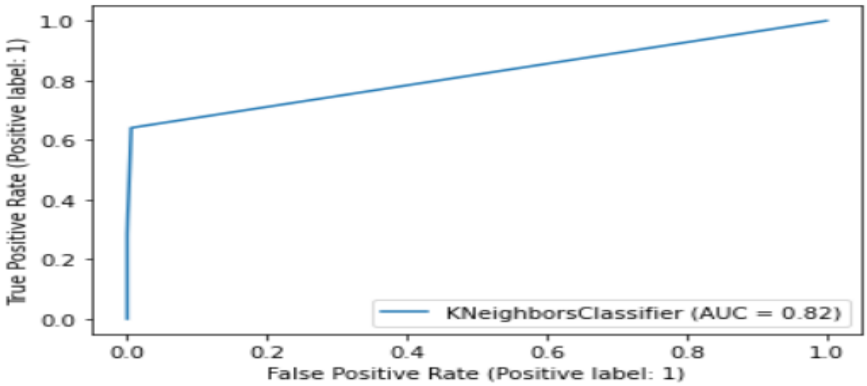
----- Roc Curve -----



- KNeighbors Classifier

```
----- Train Result -----

Accuracy Score: 0.9270160325436707

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.92      1.00      0.96      3613
           1       1.00      0.46      0.63       566

    accuracy                           0.93      4179
   macro avg       0.96      0.73      0.80      4179
weighted avg       0.93      0.93      0.92      4179


----- Confusion matrix -----
[[3613    0]
 [ 305  261]]

----- Test Result -----

Accuracy Score: 0.9145728643216080

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.91      1.00      0.95      1212
           1       0.98      0.35      0.51       181

    accuracy                           0.91      1393
   macro avg       0.95      0.67      0.73      1393
weighted avg       0.92      0.91      0.90      1393


----- Confusion matrix -----
[[1211    1]
 [ 118   63]]
```

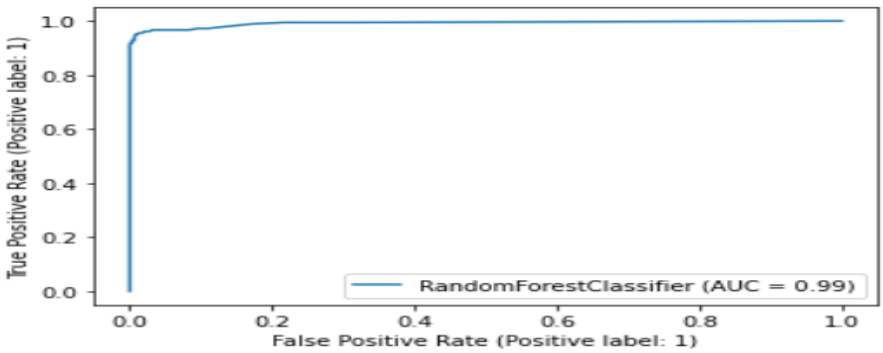
----- Roc Curve -----



- RandomForest Classifier

```
----- Train Result -----

Accuracy Score: 0.9997607083034219

----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3613
           1       1.00      1.00      1.00       566

    accuracy                           1.00      4179
   macro avg       1.00      1.00      1.00      4179
weighted avg       1.00      1.00      1.00      4179


----- Confusion matrix -----
[[3612    1]
 [   0  566]]

----- Test Result -----

Accuracy Score: 0.9842067480258435

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1212
           1       1.00      0.88      0.94       181

    accuracy                           0.98      1393
   macro avg       0.99      0.94      0.96      1393
weighted avg       0.98      0.98      0.98      1393


----- Confusion matrix -----
[[1212    0]
 [  22  159]]
```

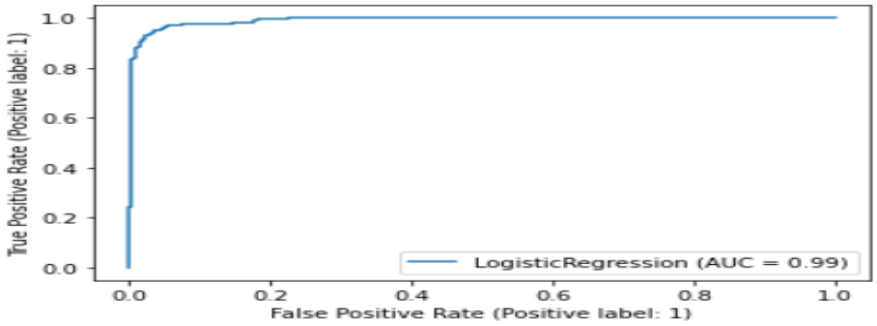
----- Roc Curve -----



- Logistic Regression

```
----- Train Result -----

Accuracy Score: 0.9777458722182341

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      3613
           1       0.97      0.86      0.91       566

    accuracy                           0.98      4179
   macro avg       0.98      0.93      0.95      4179
weighted avg       0.98      0.98      0.98      4179


----- Confusion matrix -----
[[3600   13]
 [  80  486]]

----- Test Result -----

Accuracy Score: 0.9712849964106246

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.97      1.00      0.98      1212
           1       0.99      0.79      0.88       181

    accuracy                           0.97      1393
   macro avg       0.98      0.89      0.93      1393
weighted avg       0.97      0.97      0.97      1393


----- Confusion matrix -----
[[1210    2]
 [  38  143]]
```

----- Roc Curve -----



- Naive Bayes (GaussianNB)

----- Train Result -----

Accuracy Score: 0.968413496051687

----- Test Result -----

Accuracy Score: 0.11198851399856424

Interpretation of the Results

RandomForest Classifier is giving the best result as compared to others.

CONCLUSION

Key Findings and Conclusions of the Study

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.