FLIP ROBO

Spam Classification Model Project

Submitted by:

- Prathamesh Vilas Mistry

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to “FlipRobo Technologies” that made this project possible. I would like thank my SME Mr. Harsh Ayush for his guidance in building this project. I would also like to thank Data Trained Institution to making me capable of making proper decisions in the field of damascene and Machine learning. Lastly, I would like to thank my parents to make all of this happen.

References:

- Data Preparation for Machine Learning by Jason Brownlee
- Natural Language Processing with Python
- Natural Language Processing with Python by Steven Bird, Ewan Klein, Edward Loper
- Medium.com
- AnalyticsVidhya.com

Research Papers:

- Using Naïve Bayes Model and Natural Language Processing for Classifying Messages on Online Forum

INTRODUCTION

- Business Problem Framing

You were recently hired in Start-up Company and you were asked to build a system to identify spam emails. We have to build a machine learning model that will predict if the email is 'HAM' or 'SPAM'.

- Conceptual Background of the Domain Problem

Natural Language processing or NLP is a subset of Artificial Intelligence (AI), where it is basically responsible for the understanding of human language by a machine or a robot.

One of the important subtopics in NLP is Natural Language Understanding (NLU) and the reason is that it is used to understand the structure and meaning of human language, and then with the help of computer science transform this linguistic knowledge into algorithms of Rules-based machine learning that can solve specific problems and perform desired tasks.

- Review of Literature

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. Electronical Communication is the need of the day. For shopping for bread to hollering the Emergency Services there’s no second to communication.

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.

- Motivation for the Problem Undertaken

This is a Natural Language Processing Problem and this will lead us to create highly communicative machines using human-native language.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

The Problem is of Classification.

The Data consists of 2 features in the dataset.

There are more than 2.7K samples in the dataset.

All the Data is of string datatype.

Use appropriate algorithms to build up the model.

The data is in English language which also consists of numbers and special characters.

The dirty data should be cleaned in order to retrieve meaning from the data

There are 62 records with missing values in the dataset.

Also create word dictionary and word cloud for further and future analytics.

The target classes has 20:80 ratio for spam: ham

The target has 2 classes only, it is a binary classification problem.

Using appropriate metrics for scoring and evaluations .

- Data Sources and their formats

The data was provided by the client to “FlipRobo Technologies”. The data is in the form of a comma separated file (CSV). The data i.e. the features and the target are in the single file.

- ## Data Pre-processing Done

The Data pre-processing done is as follows:

1. Removing Stop words from the data.
2. Removing punctuations and other special characters from the records
3. Some more granular cleaning for treating hyphen and underscore joined words.
4. Removing the words which are less than 3 letters in length
5. Perform Stemming using PorterStemmer class from sklearn library
6. Perform Lemmatizing using WordNet class from sklearn library
7. Further, we remove all the words which do not convey any meaning in the context of the English Language
8. Vectorize the data using tf-idf Vectoriser

- ## Data Inputs- Logic- Output Relationships

Describe the relationship behind the data input, its format, the logic in between and the output. Describe how the input affects the output.

Data is fed in the form of a Pandas data frame to the model. The data is the vectorised meaningful words of the records. For the output we get the predicted label value of the record, that is whether the document is likely to be a same email of not. The output results in a binary value either 1 or 0 respectively.

- State the set of assumptions (if any) related to the problem under consideration

There are no such formal assumptions as we are using the Naïve Bayes' Multinomial Naïve Bayes algorithm. The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts.

- Hardware and Software Requirements and Tools Used

Hardware Required:

- A computer with a processor i3 or above.
- More than 4 GiB of Ram.
- GPU preferred.
- Around 100 Mib of Storage Space.

Software Required:

- Python 3.6 or above
- Jupyter Notebook.
- Google Collab.
- Excel

Tools/Libraries Used:

1. Computing Tools:
   - Numpy
   - Pandas
   - Scipy
   - Sk-learn
   - NLTK
2. Visualizing Tools:
   - Matplotlib
   - Seaborn
3. Saving Tools:
   - Joblib

Model/s Development and Evaluation

- ## Identification of possible problem-solving approaches (methods)

  Describe the approaches you followed, both statistical and analytical, for solving of this problem.

- ## Testing of Identified Approaches (Algorithms)

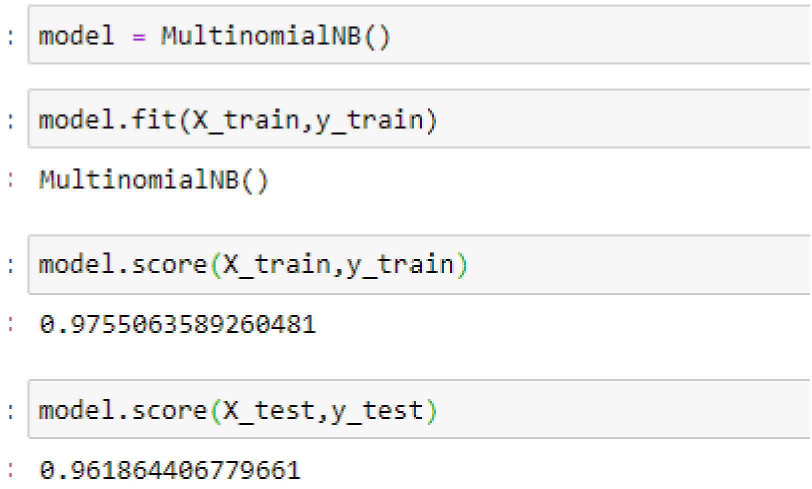
  Listing down all the algorithms used for the training and testing.

  The Algorithms used for testing, training and Validating the models are as follows:

  - Logistic Regression
  - SVC (with an rbf kernel)
  - Decision Tree
  - K Nearest Neighbour
  - Naïve Bayes
  - Random Forest
  - Gradient Boosting

- Run and Evaluate selected models

  Algorithms used and their Evolutions of the Selected Models:



```
: model = MultinomialNB()
```

```
: model.fit(X_train,y_train)
```

```
: MultinomialNB()
```

```
: model.score(X_train,y_train)
```

```
: 0.9755063589260481
```

```
: model.score(X_test,y_test)
```

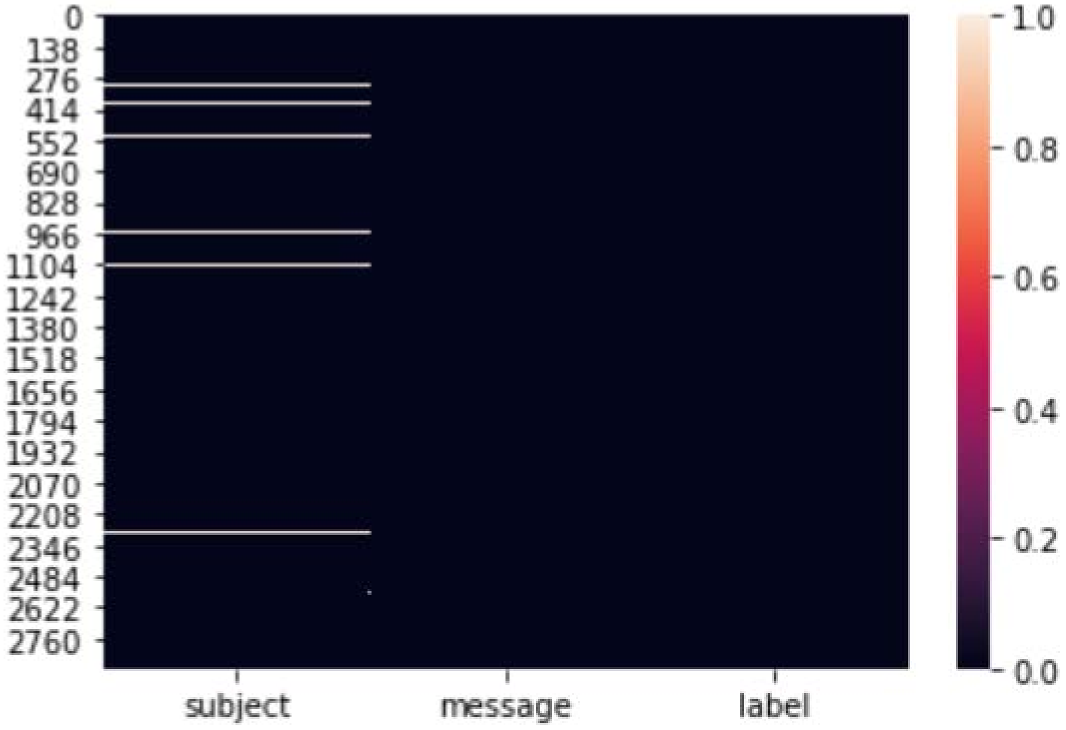
```
: 0.961864406779661
```

- Key Metrics for success in solving problem under consideration

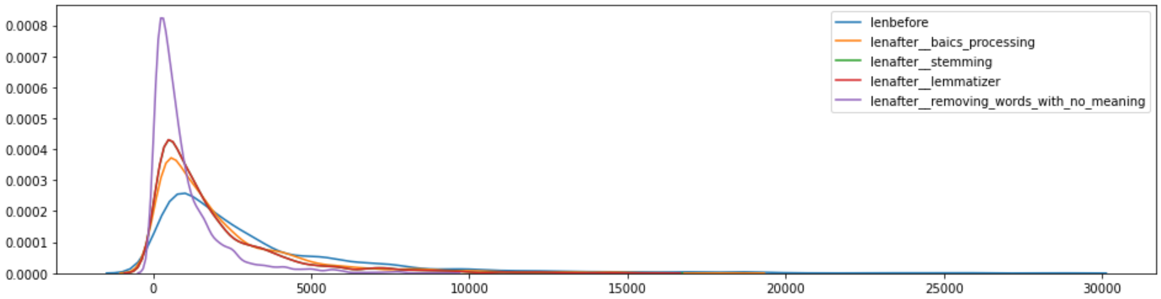
  The key-metric under considerations is Recall and F1 score although the model was finalized on basis of other metrics as Matthew's Correlation Coefficient (MCC) as well as F1-score.

- Visualizations

  ➢ Very few missing data

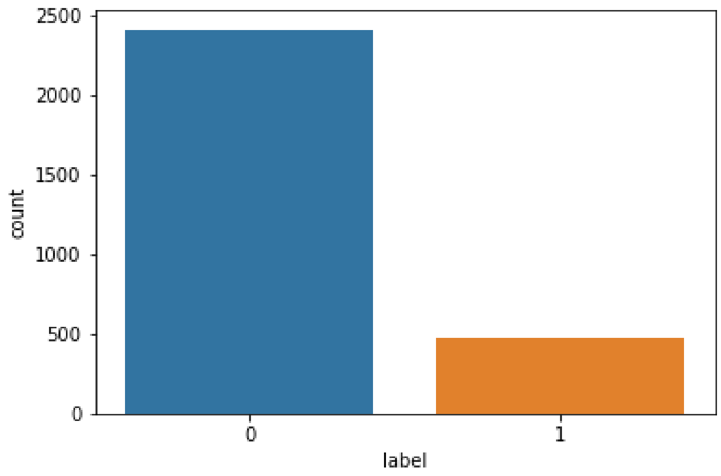


  ➢ KDE plot of length of records after various processing step wise (blue: defaulters)

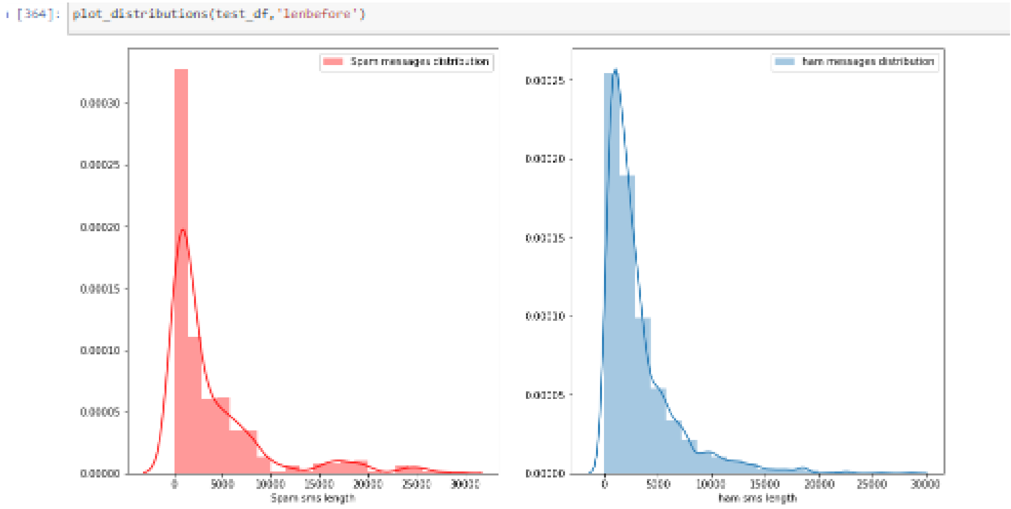


As the data is processed the spread of the data is constrained .

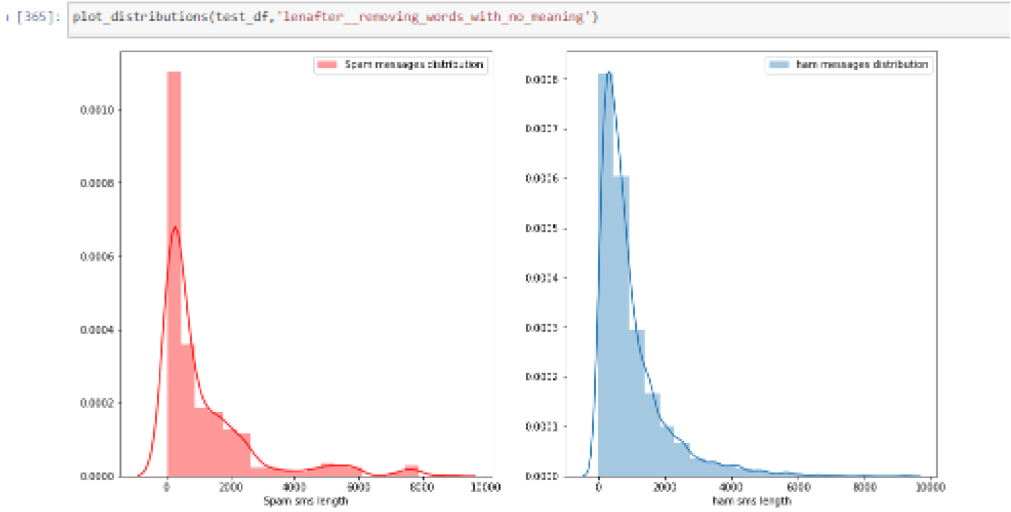
- Imbalance of the target classes



- Distributions of Ham and Spam emails before(upper row) and after (bottom row) processing



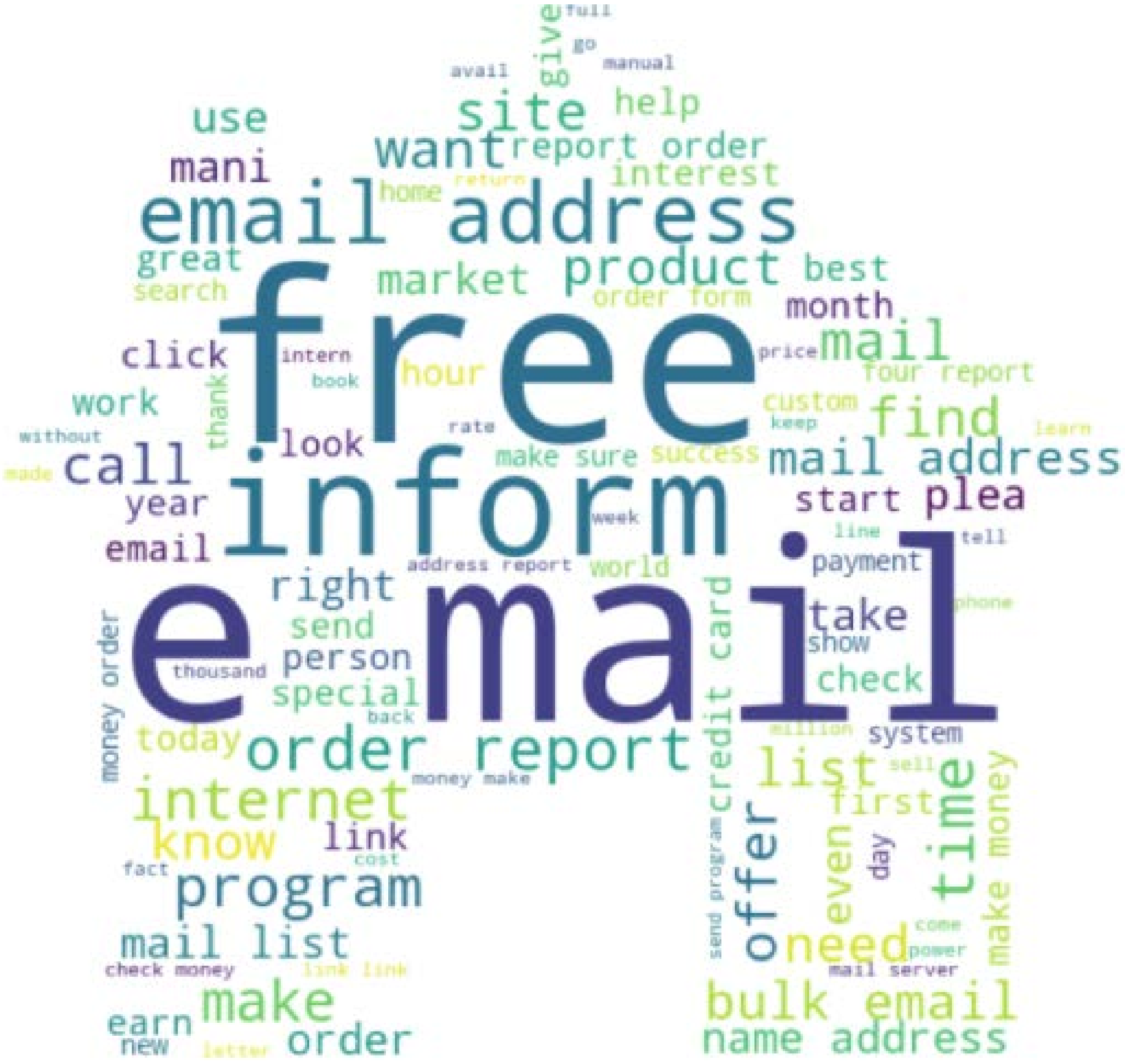
```
[364]: plot_distributions(test_df,'lenbefore')
```



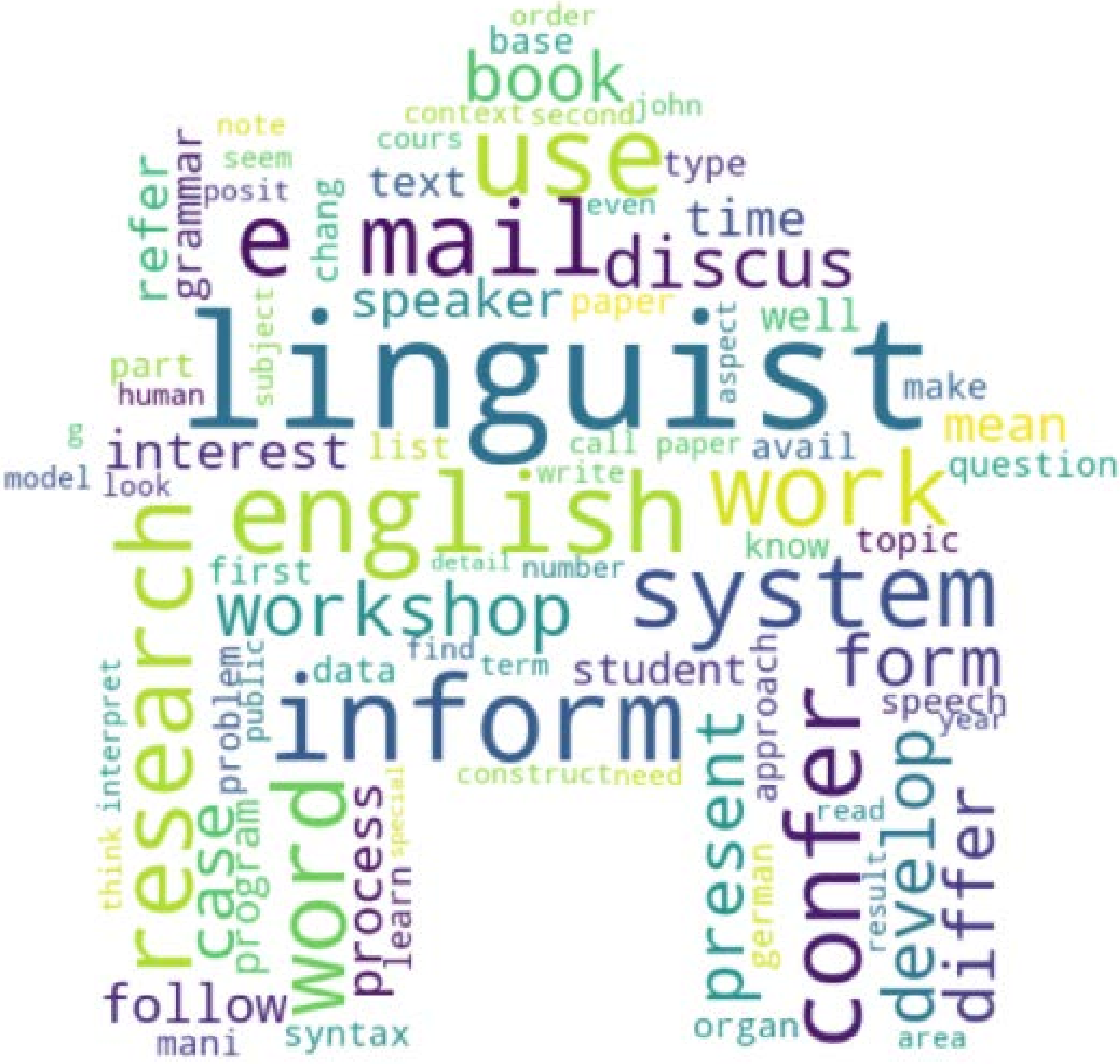
```
[365]: plot_distributions(test_df,'lenafter__removing_words_with_no_meaning')
```

- WordClouds

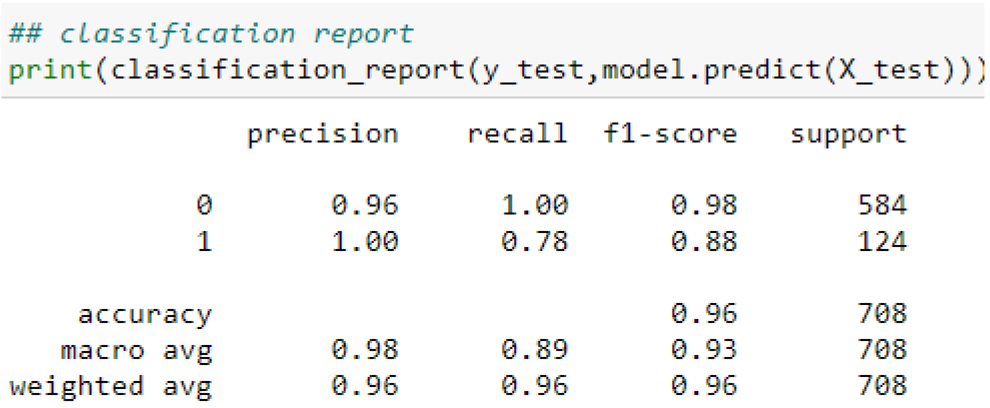
This is the word Cloud for the **Spam** messages



This is the word Cloud for the **<u>Ham</u>** messages



- Interpretation of the Results



```
## classification report
print(classification_report(y_test,model.predict(X_test)))
```

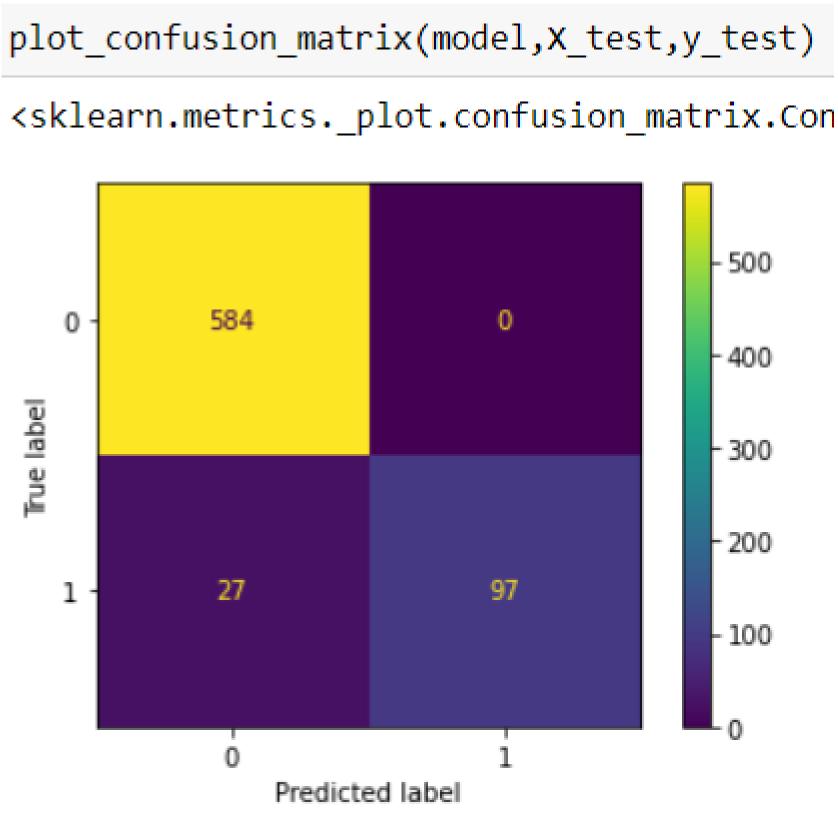
```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98       584
           1       1.00      0.78      0.88       124

    accuracy                           0.96       708
   macro avg       0.98      0.89      0.93       708
weighted avg       0.96      0.96      0.96       708
```

This is the classifications report on the test set. Since we have high imbalance in our target classes we used AUC_ROC to evaluate the model.

➢ Confusion Matrix on the test data



```
plot_confusion_matrix(model,X_test,y_test)
```

```
<sklearn.metrics._plot.confusion_matrix.Con
```

CONCLUSION

- ## Key Findings and Conclusions of the Study

NLP gets hard are humans are not used to typing as proper grammar these years.

Sweet spot should be found between whether to pick stemming or lemmatization or both.

Naïve Bayes algorithms are quicker than rest of the algorithms.

- ## Learning Outcomes of the Study in respect of Data Science

List down your learnings obtained about the power of visualization, data cleaning and various algorithms used. You can describe which algorithm works best in which situation and what challenges you faced while working on this project and how did you overcome that.

Outcomes of the Study:

- Almost 90 percent of the time is spent of data cleaning and data modelling.
- You do not get a Gaussian distribution in real-word problem.
- NLP becomes difficult due to sloppy use of language by humans
- This also created issue while teaching to machines
- Algorithms like Support Vector Machines and K nearest neighbours may take a long time to converge on a Hugh dataset like this.
- Naïve Bays is very quick as of converging rate.

- Limitations of this work and Scope for Future Work

  More data is always appreciated

  The model could be integrated with the any email app used by the Data Analysts and Developers for easy spam filtering decision.

  The model could be placed into a Continuous Integration and Continuous Deployment for online training environment.