Minor Project Final Evaluation
(SMS Categorizer)

Division of Work

- Ayush Pathak: Deployment of Servers in Desktop Application , Secondary android developer,
- Yash Varshney: Primary android Developer
- Shrey Batra and Yash Singhal: Wrangling Text Messages and Producing the Model

S.No	Name	Enrollment No	Contribution
1	Yash Varshney	15103314	
2	Yash Singhal	15103315	
3	Ayush Pathak	15103293	
4	Shrey Batra	15103308	

Problem Statement

The Project Submission Consist of Applying Natural Language Processing and Passing it through a Machine Learning Algorithm to Classify the incoming message on an Android Phone automatically depending upon its Content.

Project Submission Checklist

- **A Jupyter Notebook File Detailing the Work Done on Natural Language Processing and Machine Learning for this particular Project**
- **A Jupyter Notebook File Detailing the Work Done as Part of Data Preprocessing and Collection and Compiling of the Messages from the User Phone**
- **A server side script that will allow us to deploy the Model on to the Server and Listen for the incoming messages and classifying as accordingly**

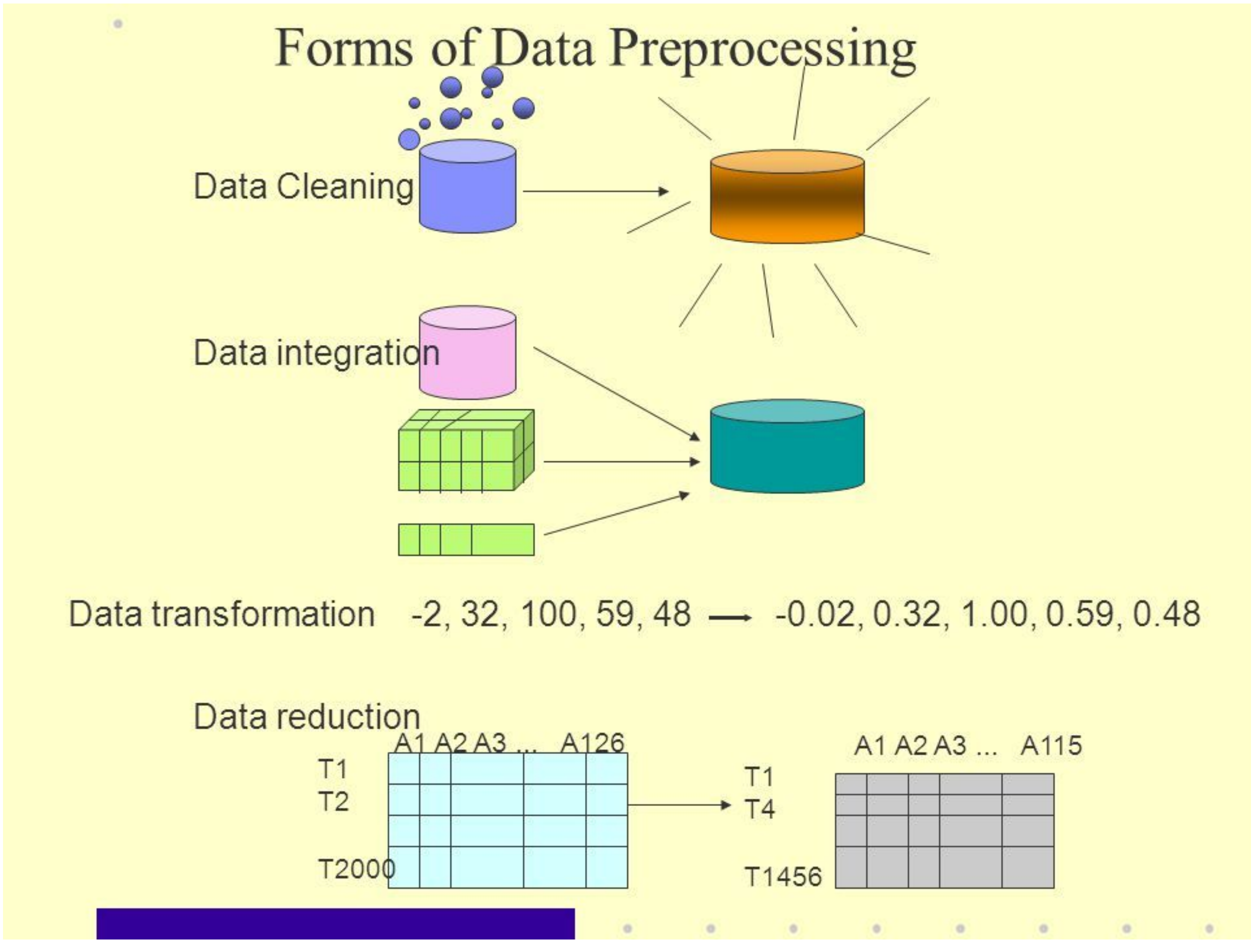
- **A readSms App for Android which was developed by us as an assist to help the development of the project which allowed us to extract out the messages from the User Android Phone**

- **A Sms Categorizer App which being the Final Tool where all things came together . The Categorizer App recieves the Response from the Server and classify the message Accordingly**

Phases of Machine Learning

1. Preprocessing of the Data and Text Mining

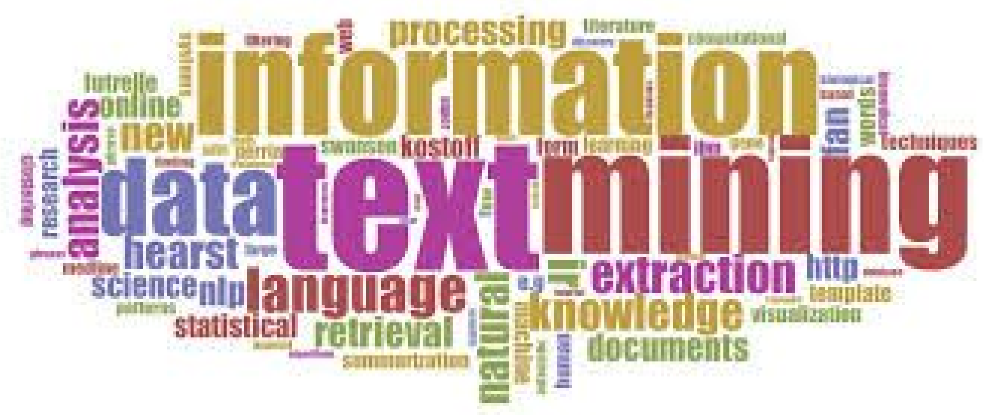
Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing


Forms of Data Preprocessing
Data Cleaning
Data integration
Data transformation -2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48
Data reduction
A1 A2 A3 ... A126
T1
T2
T2000
A1 A2 A3 ... A115
T1
T4
T1456

Data goes through a series of steps during preprocessing:

- **Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.**

- **Data Integration: Data with different representations are put together and conflicts within the data are resolved.**
- **Data Transformation: Data is normalized, aggregated and generalized.**
- **Data Reduction: This step aims to present a reduced representation of the data in a data warehouse**



The purpose of Text Mining is to process unstructured (textual) information, extract meaningful numeric indices from the text, and, thus, make the information contained in the text accessible to the various data mining (statistical and machine learning) algorithms. The Natural Language Processing Usually Consist of Making the Bag of Words as a way to produce Meagninful inputs to the Machine Learning Algorithm

Data Processed after NLP can be feed into an Unsupervised or Supervised Machine Learning Algorithm to Find Patterns or to extract out specific Relevant Information

2. Processing of Data

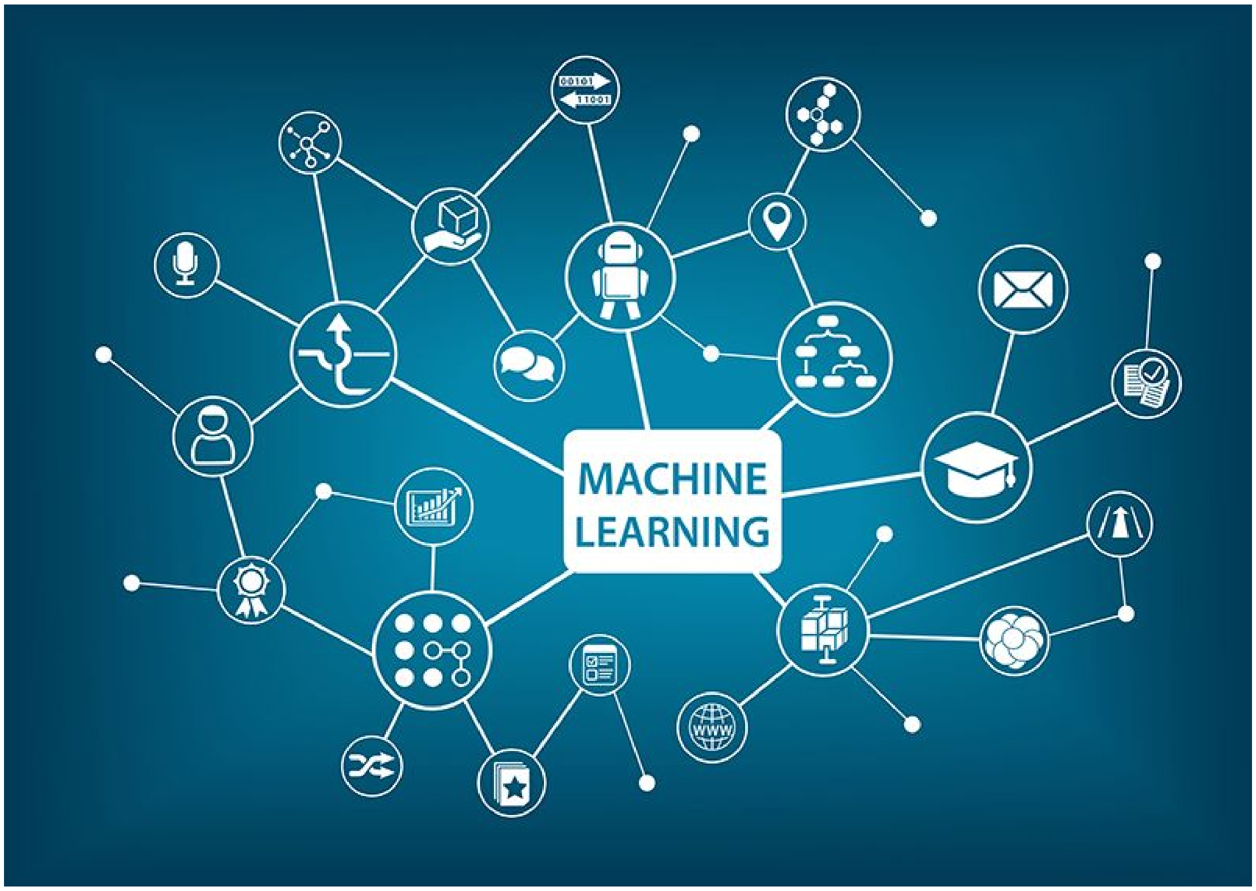
Operations performed on a given set of data to extract the required information in an appropriate form such as diagrams, reports, or tables. This phase is also called the Learning Phase and this Really forms what is the Crux of the Project.

Machine Learning

Machine learning is a type of artificial intelligence (AI) that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output value within an acceptable range.

Machine learning algorithms are often categorized as being supervised or unsupervised. Supervised algorithms require humans to provide both input and desired output, in addition

to furnishing feedback about the accuracy of predictions during training. Once training is complete, the algorithm will apply what was learned to new data.



The processes involved in machine learning are similar to that of data mining and predictive modeling. Both require searching through data to look for patterns and adjusting program actions accordingly.

<u>For this Project we will be using Naive Bayes as the Learning Algorithm and Count Vectorizer and TFIDF Transformer as the Data Preprocessing Step</u>

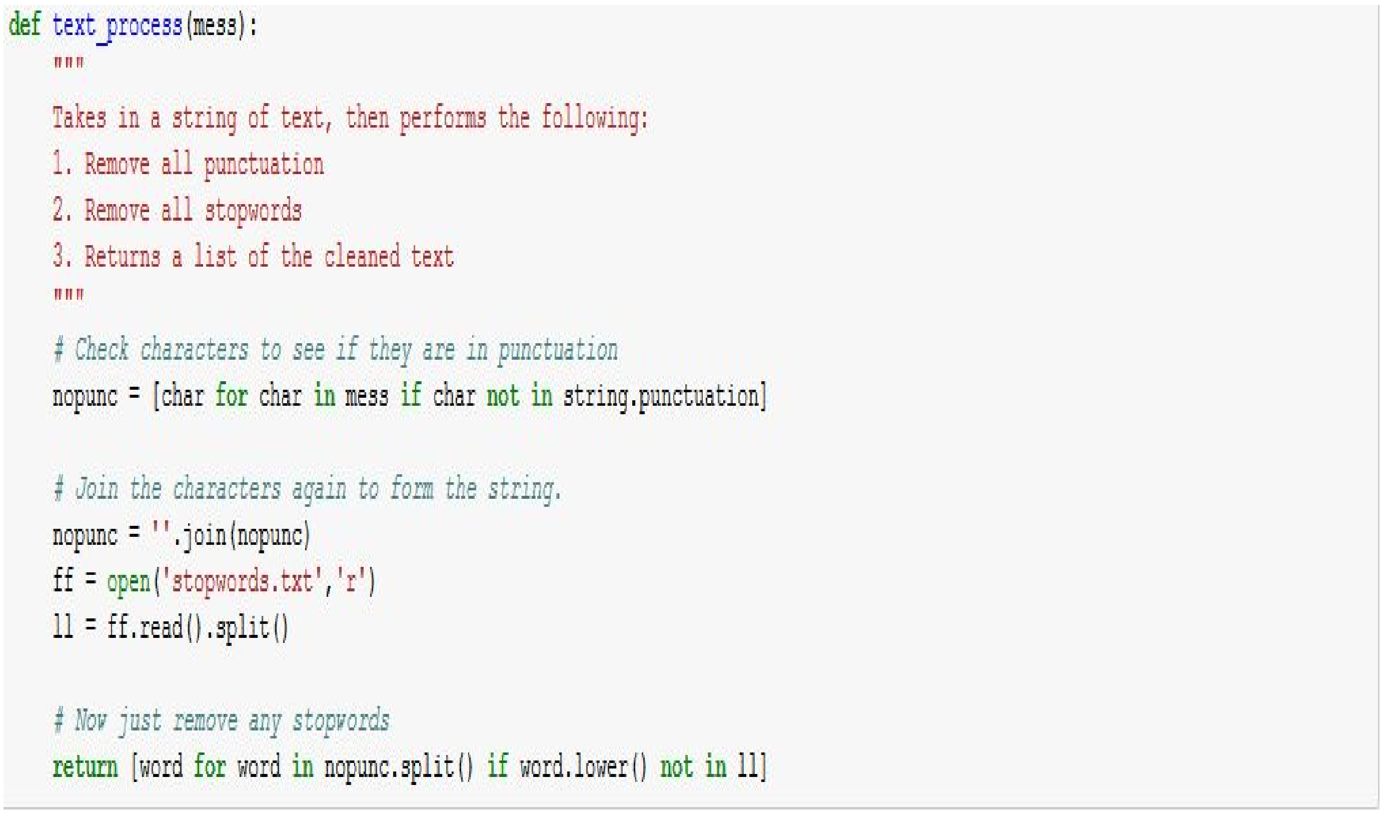
Workflow and Procedure

- **The Read Sms is Used to Extract out the Messages from the Android Phone . The Read Sms allows the Message extraction in the form of csv File along with the Metadata related to the Message**

- **The SMS Messages were then Individually Labeled and Classified into Following Categories**

    <u>Fashion</u>
    <u>food</u>
    <u>finance</u>
    <u>health</u>
    <u>education</u>
    <u>Telecom</u>
    <u>Other</u>

- **The Count Vectorizer and Tfidf Transformer is then instantiated on to the Dataset along with passing of the Above Function. The Function remove the Punctuation along with Removal of StopWords .**



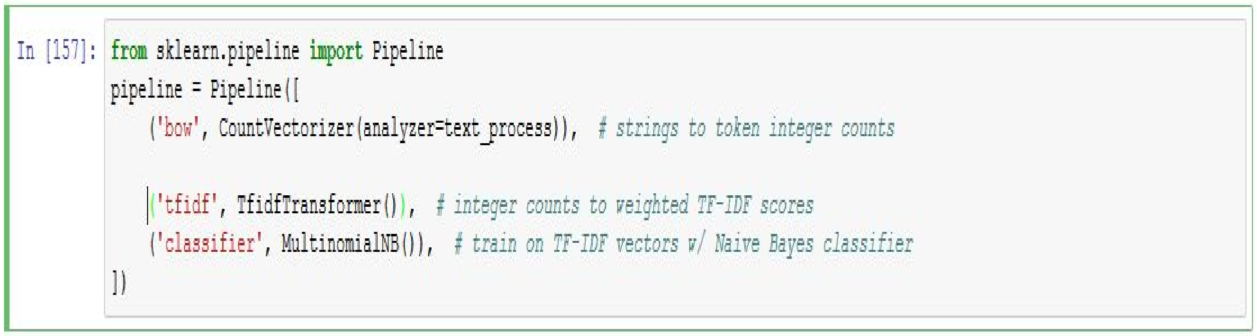
```python
def text_process(mess):
    """
    Takes in a string of text, then performs the following:
    1. Remove all punctuation
    2. Remove all stopwords
    3. Returns a list of the cleaned text
    """
    # Check characters to see if they are in punctuation
    nopunc = [char for char in mess if char not in string.punctuation]

    # Join the characters again to form the string.
    nopunc = ''.join(nopunc)
    ff = open('stopwords.txt','r')
    ll = ff.read().split()

    # Now just remove any stopwords
    return [word for word in nopunc.split() if word.lower() not in ll]
```

(The Function removes The Punctuation and stop words to facilitate the working of vectorizer)

- **The Transformed Data points are then Split using Train Test Split with test size of 20 percent**

- **The Trained Data tupple is then Fed into the Naive Bayes Classifier with alpha Value of 0.5 (The Learning Parameter)**

- **The Model is then Predicted with The Test Data Point to Compute Accuracy with the Labels of the Test Data Point**



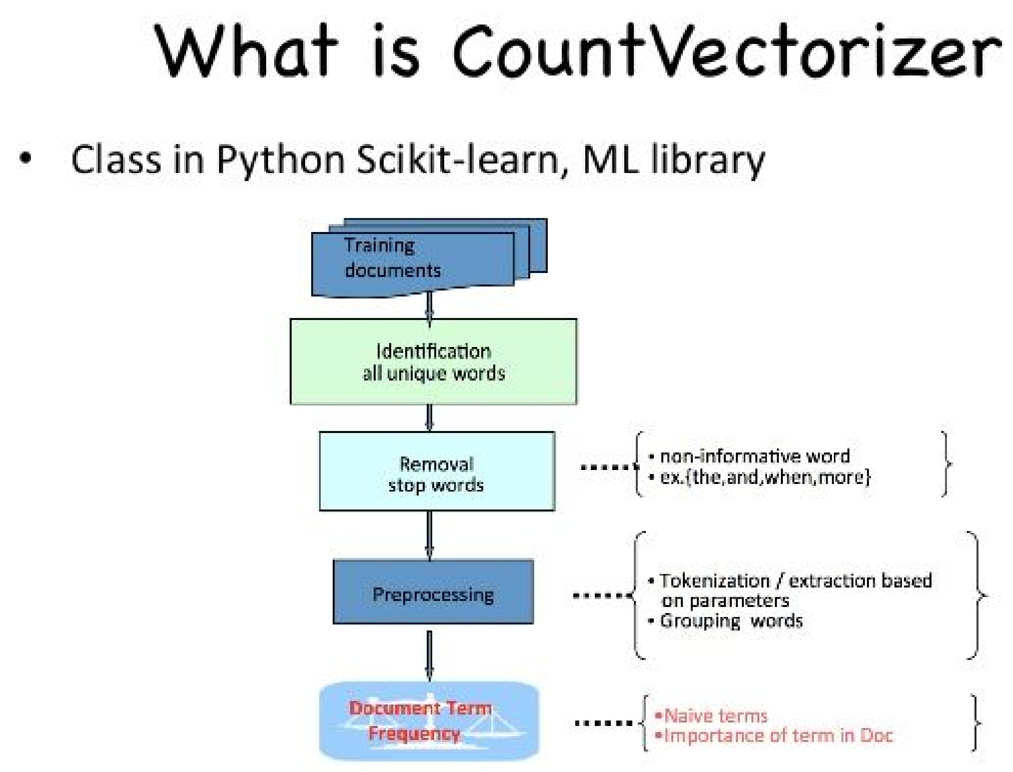
```
In [157]: from sklearn.pipeline import Pipeline
          pipeline = Pipeline([
              ('bow', CountVectorizer(analyzer=text_process)),  # strings to token integer counts

              ('tfidf', TfidfTransformer()),  # integer counts to weighted TF-IDF scores
              ('classifier', MultinomialNB()),  # train on TF-IDF vectors w/ Naive Bayes classifier
          ])
```

- **All the Above Process is Faciliated with Pipeline which Allows the Datapoint to fed into the next Process Directly after getting Output from Previous Step**

- **Deployment of the Model onto The Server Using Flask Framework**

- **Android App (SMS Categorizer) that Categorizer the message according to the Response recieved From the Server that Hosts The Model**

Algorithm and Tools specific to NLP



What is CountVectorizer

- Class in Python Scikit-learn, ML library

- **CountVectorizer**

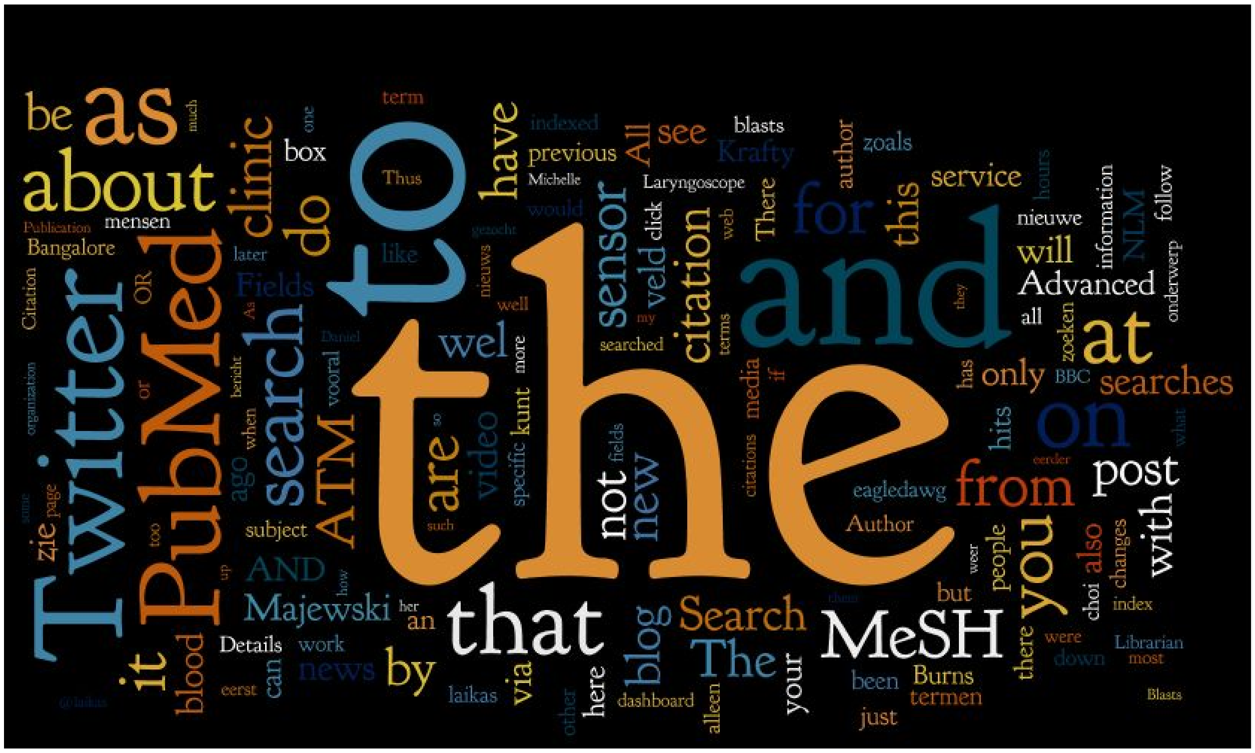
  Convert a collection of text documents to a matrix of token counts This implementation produces a sparse representation

of the counts. The Count Vectorizer is Customizable , we can provide Count Vectorizer the Function That Automatically Calls it for removal of Words Before Tokenizing the words

- **TfIDFTransformer**

Transform a count matrix to a normalized tf or tf-idf representation Tf means term-frequency while tf-idf means term-frequency times inverse document-frequency. This is a common term weighting scheme in information retrieval, that has also found good use in document classification.The goal of using tf-idf instead of the raw frequencies of occurrence of a token in a given document is to scale down the impact of tokens that occur very frequently in a given corpus and that are hence empirically less informative than features that occur in a small fraction of the training corpus.

- **Stop Words**



Stop Words are words which do not contain important significance to be used in Search Queries. Usually these words are filtered out from search queries because they return vast amount of unnecessary information. The Stop words are usually the words corresponding to Common Connectivities

between the Sentences And common Pronouns of the specific Language. The Stop words when Passed through a TFIDFTransformer returns a Very Low Weightage as expected

Supervised Classification (Naive Bayes)

$$\underset{\text{Posterior Probability}}{P(c \mid x)} = \frac{\overset{\text{Likelihood}}{P(x \mid c)}\,\overset{\text{Class Prior Probability}}{P(c)}}{\underset{\text{Predictor Prior Probability}}{P(x)}}$$

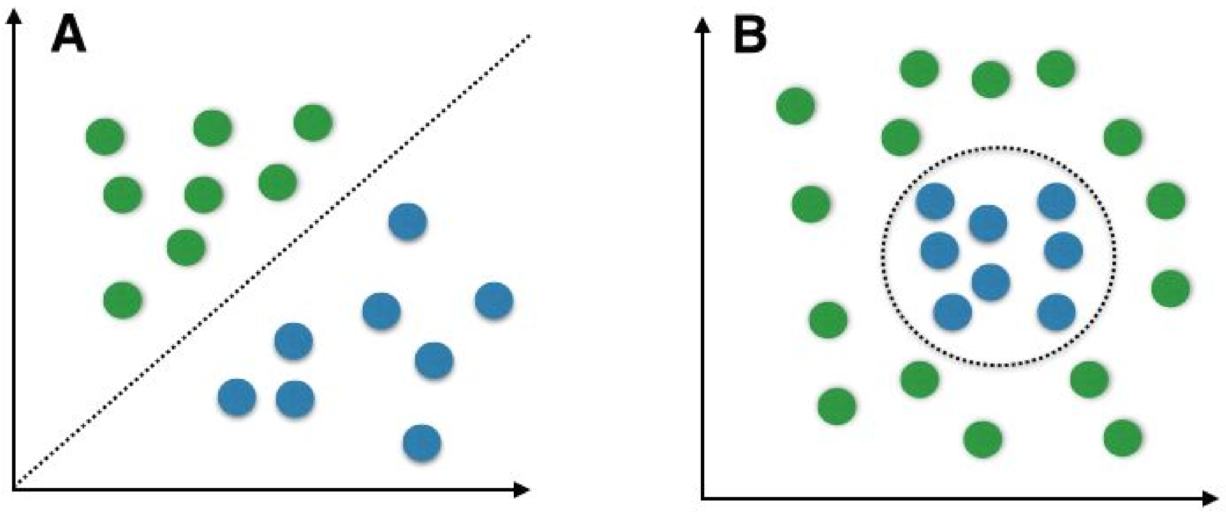
$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

The Naive Bayes Algorithm is used as an Supervised Classification for this Project. The Naive Bayes is particularly Recommended For Handling of Tokenizer as it it is Particulary well suited to Large Number of Features. The

Naive Bayes Algorithm are Family of Classes and encompases several Classification alorithm based on Bayes Probability

- 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. However, in practice, fractional counts such as tf-idf may also work.

- It estimates the conditional probability of a particular word given a class as the relative frequency of term t in documents belonging to class(c). The variation takes into account the number of occurrences of term t in training documents from class (c),including multiple occurrences.

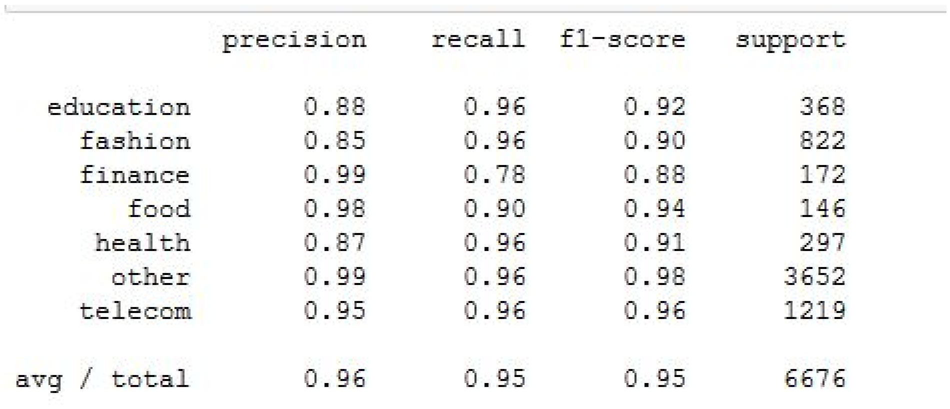


(Decision Boundary in Classification ML)

In Layman Terms given the Class it measures the Probability that a Class would be involved with a Particular word given that the particular Word is Found

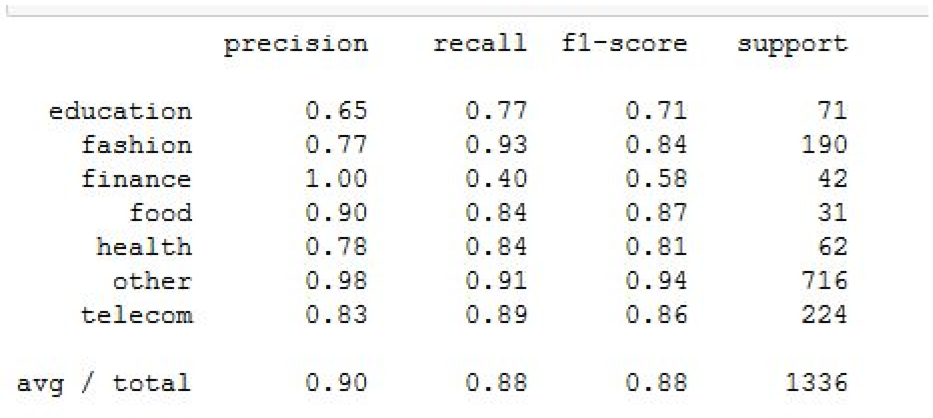
Benchmark and Prediction

A standard or point of reference against which things may be compared.



```
             precision    recall  f1-score   support

  education       0.88      0.96      0.92       368
    fashion       0.85      0.96      0.90       822
    finance       0.99      0.78      0.88       172
       food       0.98      0.90      0.94       146
     health       0.87      0.96      0.91       297
      other       0.99      0.96      0.98      3652
    telecom       0.95      0.96      0.96      1219

avg / total       0.96      0.95      0.95      6676
```

(Precision Score on Training Dataset)



```
             precision    recall  f1-score   support

  education       0.65      0.77      0.71        71
    fashion       0.77      0.93      0.84       190
    finance       1.00      0.40      0.58        42
       food       0.90      0.84      0.87        31
     health       0.78      0.84      0.81        62
      other       0.98      0.91      0.94       716
    telecom       0.83      0.89      0.86       224

avg / total       0.90      0.88      0.88      1336
```

(Precison score on Testing Dataset)

- **The Model Trained , perfomed very well on Training as well as on Testing Dataset , on Testing Dataset the Model performed upwards of 95 percent and about 90 percent accuracy on Testing Dataset.**

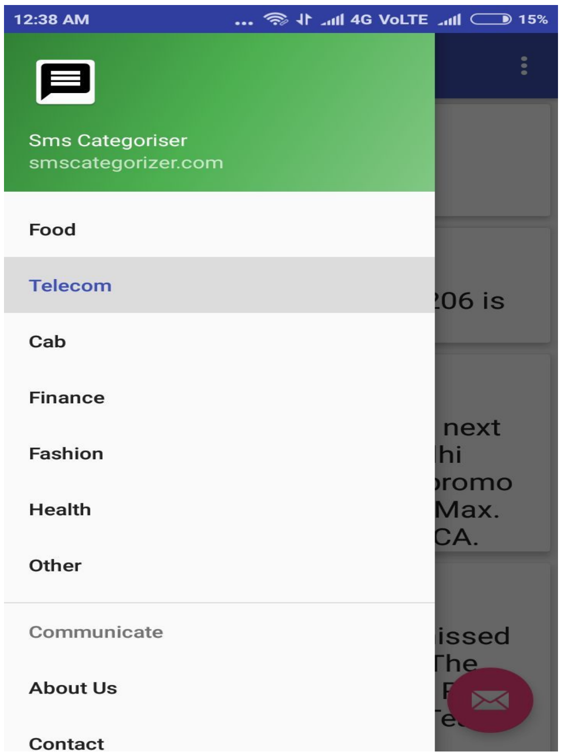
- **The almost Equal Score on Training and Testing dataset is an indication that the Model is not overfitting or underfitting**

- **Benchmark Prediction:Precision**

  The Precision calculates the Proportion that how accurately the Model Was able to classify the

messages belonging to different category. Higher the positive hit greater the precision

SMS Categorizer App

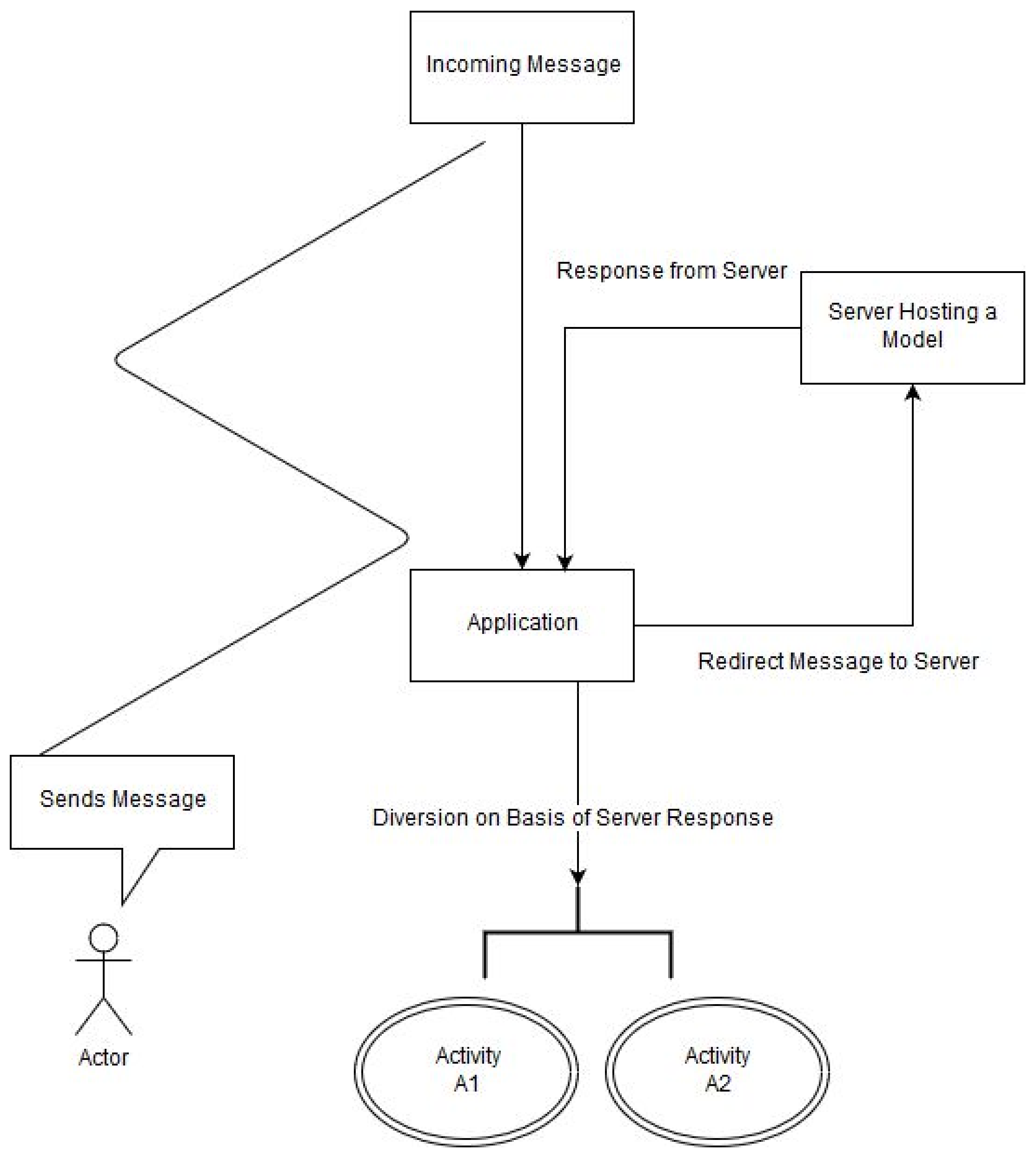


- A Sms Categorizer App was developed which Consist of Specific Activity for each of the Category which will Group the similar Messages

- App post message on to the Server Consist of deployed Model which will Predict the Given Message and returns The response in the Form of JSON Data . The App then

Redirect the message on to the specific activity based on the Response Received

Flowchart of Working of App



Future Scope

While The SMS categorizer was providing the Basic Functionality expected of SMS client. There are several things that can be expanded like

- **Inclusion of Tagging a multitude of messages as important and pinning them for specific duration**

- **Improving the model by dynamic Training by inclusion of new incoming messages**

- **Sync Messages to the Cloud**

- **Suggestion of Replying to the Incoming messages Depending upon the content like (thank you, Great see You soon)**

References

[1]“Scikit-learn-Documentation”-http://scikit-learn.org/stable/documentation.html

[2] “A Brief Explanation of Different Supervised Algos”-http://scikit-learn.org/stable/supervised_learning.html

[3] Ashraf M. Kibriya, Eibe Frank, “Multinomial Naive Bayes for Text. Categorization Revisited.”:*Walkato University*

[4] A McCallum."A Comparison of Event Models for Naive Bayes Text Classification”: *JustReach@com*

[5] “Natural Language Processing”-http://www.nltk.org/