

**COMBINING THE ENSEMBLE APPROACH AND THE POPULAR WORD  
VECTORIZATION/FEATURE ENGINEERING TECHNIQUES TO IMPROVE THE PERFORMANCE OF  
THE SENTIMENT CLASSIFIER AND EVALUATE THE PERFORMANCE OF THE CLASSIFIERS  
WITH THE CHANGE IN THE WORD VECTORIZATION TECHNIQUE-SENTIMENT  
CLASSIFICATION.**

**SHOBHANA HALAPETI  
SUPERVISOR- DIPAYAN SARKAR**

Final Thesis Report

FEBRUARY 2020

## **Abstract**

In machine learning, there are various feature engineering strategies for dealing with structured data. Feature Engineering is an art of the state for the optimization of the machine learning classifier. Text data is one of the most abundant sources of unstructured data. Text data usually consists of documents that can represent words, sentences, or even paragraphs of free-flowing text. The inherent unstructured and noisy nature of textual data makes it harder for machine learning algorithms to work on raw text data directly. Hence, the text data is converted to a one-hot encoded form, which machines can understand and learn. The ensemble approach on the base classifiers and the different word vectorization techniques are cited in many research papers discretely to improve the performance of the base classifiers. In this research, the change in the performance of the base classifiers trained on each feature subset created using different word vectorization/embedding technique and the corresponding ensemble model is evaluated. We have trained the base classifiers on each feature subset created using the different feature vectorization techniques and then combined the results of the base classifiers to get the ensemble model. The aim is to analyze the significance of the performance improvement of the base classifiers and the ensemble model with a change in the word vectorization technique. We have the base classifiers like the Logistic Regression and the Support Vector Machine performing better compared to the other base classifiers like the Naïve Bayes and the Decision Tree classifier. However, the Logistic and the Support Vector classifiers show low recall for Negative sentiments classification. The Ensemble Model performs well in terms of the recall across all feature subsets compared to the strong learners like the Logistic Regression and the Support Vector Machine. The performance of the ensemble model on the Word2Vec vector is on top of the performance of the ensemble model on the other feature subsets.

## **Keywords**

Base Classifier for Sentiment Analysis, Ensemble Classifier, Feature Extraction, Word Vectorization (Bag of words, TF-IDF model), Gini Index, Information Gain, Chi-Square method, Word2Vec, Naïve Bayes, Support Vector Machines, Logistic Regression, Decision Tree, Recall, Precision, Accuracy, F-Measure.

## **1.Introduction**

Sentiment Analysis is gaining popularity due to the enormous volume of the unstructured text produced through different social media platforms, e-commerce websites, blogs, online portals. Sentiment analysis enables the information extraction from this unstructured data, which in turn helps businesses to improve their products/services based on the customers' sentiments and helps the customers to know about the products before they buy. Sentiment Classification is an open research domain.

As proven in research studies (Ankit and Saleena, 2018) & (Sadhasivam and Kalivaradhan, 2019), the Ensemble Sentiment Classifier performs better when compared to the stand-alone classifier. Features engineering/ word vectorization/ word embedding is another area where research work is cited (Fang and Zhan, 2015) (Ghosh and Sanyal, 2018) & (Ma and Zhang, 2015). Efficient features selection techniques, word embedding, vector dimension reduction techniques are proven methods to improve the performance of the classifier

significantly. Still, research work is going on to find out better alternatives due to the importance of the sentiment classification in the Business Industry. It has been noticed that the base classifiers do not perform significantly well due to the complexity involved in text data in terms of noise, volume.

Lexical techniques like removing stop words, tokenization, stemming, lemmatization, along with few other pre-processing steps like spell corrections, result in reducing the number of redundant features. The Bag of words and TF-IDF word vectorization techniques create words vector, which can be used to train the sentiment classifier. However, this approach does not reduce the significant dimensionality issue. This large dimensionality vectors can be reduced using Factor-based Word embedding models such as SVD/ PCA or Prediction based embedding models. The lower-dimensional space is a much richer representation of the semantics of the word. However, Latent Semantic Analysis techniques like PCA/SVD has a drawback that the resulting dimensions are not interpretable. In the skip-gram approach to generate word vectors, the input is your target word, and the task of the neural network is to predict the context words (the output) for that target word. The input word is represented in the form of a 1-hot-encoded vector. Once trained, the weight matrix between the input layer and the hidden layer gives the word embeddings for any target word. In the CBOW model, instead of predicting a context word from a word vector, you predict a word from the sum of all the word vectors in its context.

The objective of this research is to evaluate the significance of the change in the performance of the base classifiers and the corresponding robust ensemble model with the change in features extraction/word vectorization techniques for sentiment classification. Most of the research papers have been cited where the base classifier is trained on the feature subset using the word vectorization/word embedding techniques like word2vec, TF-IDF, Bag of Words, or the feature importance discretely. Many research papers have also been cited where the Ensemble approach is applied to improve the performance of the base classifiers. A novelty in this research is the use of the most popular word vectorization/word embedding techniques in combination with the ensemble model to evaluate the significance of the change in the performance of the base classifiers and the ensemble model for each feature subset.

In this research, Mobile reviews data is used which is scraped from 'Pricebaba' and 'Mouth shut' publicly available websites. The dataset has 290,967 documents(rows) with the columns like 'Rating' and 'Review Updated.' A rating above and equal to 3 is considered as 'Positive' sentiments and rating below 3 is considered as 'Negative' sentiments. Here the reviews of the Mobile Products for the years 2016,2017, and 2018 are considered only if the corresponding product has at least 500 reviews. The text is further pre-processed using the Spacy library. The word vectorization techniques like Bag of words and TF-IDF are used for feature extraction. Another feature subset is created with the common features selected based on feature importance using the Gini Index, Chi-Square, and Information Gain methods. Word2Vec model is used for feature embedding to reduce the dimensionality of the vector and get an essential feature subset. Then Machine Learning Classifiers have been trained on each of these features subset, and the results of the base classifiers are combined to get the ensemble model for each feature subset. The research is aimed to evaluate the change in performance and accuracy of the base classifiers and the ensemble classifier with the change in word vectorization/embedding techniques. The ensembled model is built on each feature subset using the base classifiers like Decision tree, Naive Bayes, Support Vector Machine, and Logistic Regression.

## CHAPTER 2

### LITERATURE REVIEW

The below Literature review shows the importance of the Ensemble approach as well as feature extraction techniques; however, **the change in the performance level** of the base classifiers and the robust ensemble classifier with the change in Feature extraction technique can be considered as a research to improve the accuracy of the ensemble classifier further with the change in the Feature extraction/word embedding techniques.

#### 2.1 Xing Fang, Justin Zhan (2015)

**Word vectorization Techniques like frequently occurring word tokens and phrase tokens (negation-of-adjective and negation-of-verb)** - Xing Fang and Justin Zhan has proposed the general process for sentiment polarity categorization with detailed process descriptions.

Data used in this study by Author is online product reviews collected from Amazon.com. Experiments for both sentence-level categorization and review-level categorization are performed with promising outcomes. Authors have used POS taggers to filter the words tagged as nouns and pronouns. In this work, there are two types of phrases that have been identified by the Authors, namely negation-of-adjective (NOA) and negation-of-verb (NOV). The authors have selected 11,478-word tokens with each of them that occurs at least 30 times throughout the dataset. For phrase tokens, namely negation-of-adjective (NOA) and negation-of-verb (NOV), 3,023 phrases were selected of the 21,586 identified sentiment phrases, where each of the 3,023 phrases also has an occurrence that is no less than 30. With this approach, the base classifiers like the random forest, SVM and Naïve Bayes have a very high F1 Score for review level classification and sentence level classification. Here POS tags are used for features engineering. This approach has achieved an F1 score over 0.8 for the sentence level categorization and an F1 score of over 0.73 for review level categorization. The research is focused on reducing the variation of the words using Part of Speech tagging and, in turn, the effective features engineering or word vectorization techniques in order to optimize the performance of the classifier [4].

## **2.2 Monalisa Ghosh, Goutam Sanyal (2018)**

**Ensemble approach used to combine the top-ranked common Features** - Monalisa Ghosh and Goutam Sanyal have investigated the ability of the widely used features selection methods (IG, Chi-Square, Gini Index) individually as well as their combined approach on four machine learning algorithms. The authors have evaluated the proposed methods on three datasets viz. IMDb movie review, electronics, and kitchen products review dataset. The authors have initially selected the features subset using three different feature selection methods. After that, the statistical method UNION, INTERSECTION, and revised UNION method are applied to merge these different feature subsets to obtain all top-ranked, including commonly selected features. Finally, the authors have trained the individual classifier SMO, MNB, RF, and LR (logistic regression) with these features vector for classification of the review data set. The performance of the algorithm is measured by evaluation methods such as precision, recall, F-measure, and ROC curve. Experimental results show that the combined method achieved the best accuracy of 92.31 with classifier SMO, which is encouraging and comparable to the related research. Here, the authors have used the ensemble approach to combine the common top-ranked features extracted using IG, Chi-Square, and Gini Index techniques and trained the base classifiers on the combined features set. The research is focused on Feature engineering/features extraction techniques like chi-square, GI, and IG to optimize the performance of sentiment classifiers. This research shows another different and efficient technique of feature selection, which improves the performance of the base classifier significantly [7].

## **2.3 Ankit, Nabizath Saleena (2018)**

**Ensemble Classifier combining the results of base classifiers to improve the accuracy-** Ankit, Nabizath Saleena, have used the ensemble classifier combining the base classifiers Random Forest, SVM, Logistic Regression and Naïve Bayes to improve the accuracy for Twitter Sentiment Analysis. In this research study, the ensemble approach has not improved the accuracy significantly compared to the Base classifier. For the Twitter Sentiment Analysis, the accuracy is improved to 74.67%, whereas average accuracy for the base classifier is 72%. Here in this research, the Bag of Words technique is used for word vectorization, and the Ensemble classifier is used to improve the accuracy of the classifier [1].

#### **2.4 Jayakumar Sadhasivam, Ramesh Babu Kalivaradhan (2019)**

##### **Ensemble Classifier combining the results of base classifiers to improve the accuracy -**

Jayakumar Sadhasivam and Ramesh Babu Kalivaradhan has trained the ensemble classifier combining Support Vector Machine and Naïve Bayes classifiers on Amazon Products Review dataset. Here the ensemble classifier shows significant improvement in the accuracy (73% for Positives and 78% for Negatives) compared to base classifiers (Naïve Bayes- 32% Positives and 37% Negatives, SVM- 27% Positives and 33% Negatives) [12].

#### **2.5 Long Ma, Yanqing Zhang (2015)**

**Word2Vec to process big data text** - Long Ma and Yanqing Zhang have emphasized on word2vec model for Big data processing. Processing a massive data set is time-consuming, not only due to the volume of data but also from different data types and intricate structures. Currently, many data mining and machine learning technologies are being applied to deal with big data problems. Authors have emphasized on the use of the word2vec algorithm, which is capable of selecting useful features and also reduce the feature dimension. Word2Vec, proposed and supported

by Google, is not a special algorithm, but it consists of two learning models, Continuous Bag of Words (CBOW) and Skip-gram. By feeding text data into one of the learning models, Word2Vec outputs word vectors that can be represented as a large piece of text. In this paper, the authors first trained the data via the Word2Vec model and evaluated the word similarity. Also, they have clustered similar words together and use the generated clusters to fit into a new data dimension so that the data dimension is decreased. This research is focused on the use of the word2vec model for dimension reductions, especially for substantial data sets [9].

#### **2.6 Devika M D, Sunitha C, Amal Ganesh (2016)**

##### **Sentiment Analysis: A Comparative Study on Different Approaches** - Devika M D, Sunitha

C, Amal Ganesh have emphasized on Sentiment analysis, requires the usage of a training set for performance optimization. They have mentioned that the quality of trained data plays a significant role in the accurate evaluation of the text. The semantic analysis of the sentence also increases the meaning and accuracy of the result. POS tagging is helpful to users for understanding whether the review or comment corresponds to the relevant subject searched for. This research is focused on the comparison and consolidation of the three main approaches used in sentiment analysis: 1) **Machine Learning Approach** 2) **Rule-Based Approach** 3)

**Lexicon Based Approach** and shares the information of different machine learning classifiers in the form of advantages and disadvantages. Various sentiment analysis methods and different levels of analyzing sentiments have been studied in this paper. Machine learning methods like SVM, NB, Maximum Entropy methods were discussed here in brief, along with some other interesting methods that can improve the analysis process in one or the other way. Semantic analysis of the text is of great consideration. Research work is carried out for better analysis methods in this area, including the semantics by considering n-gram evaluation instead of word by word analysis [2].

## **2.7 Walaa Medhat, Ahmed Hassan, Hoda Korashy (2014)**

**Sentiment Analysis Algorithms and applications: A survey** - Walaa Medhat, Ahmed Hassan, and Hoda Korashy have provided brief details on the various sentiment analysis techniques and the related fields. The main contributions of this paper include the sophisticated categorizations of a large number of recent articles and the illustration of the recent trend of research in the sentiment analysis and its related areas. Fifty-four of the recently published and cited articles are categorized and summarized. These articles give contributions to many SA related fields that use SA techniques for various real-world applications. With the help of briefing on recent research work on Sentiment Analysis in this paper, the below information is discovered.

- 1) Naïve Bayes and Support Vector Machines are the most frequently used ML algorithms for solving SC problems. They are considered a reference model where many proposed algorithms are compared to.
- 2) The interest in languages other than English in this field is growing.
- 3) The most common lexicon source used is WordNet, which exists in languages other than English. Building resources, used in SA tasks, is still needed for many natural languages.
- 4) Information from micro-blogs, blogs, and forums, as well as a news source, is widely used in SA recently. This media information plays a significant role in expressing people's feelings or opinions about a specific topic or product. Using social network sites and micro-blogging sites as a source of data still need more in-depth analysis.
- 5) There are some benchmark data sets, especially in reviews like IMDB, which are used for algorithm evaluation.
- 6) In many applications, it is essential to consider the context of the text and the user preferences. That is why more research is needed on context-based SA. Using TL techniques,

related data to the domain can be used in question as training data. Using NLP tools to reinforce the SA process has attracted researchers recently and still needs some enhancements [10].

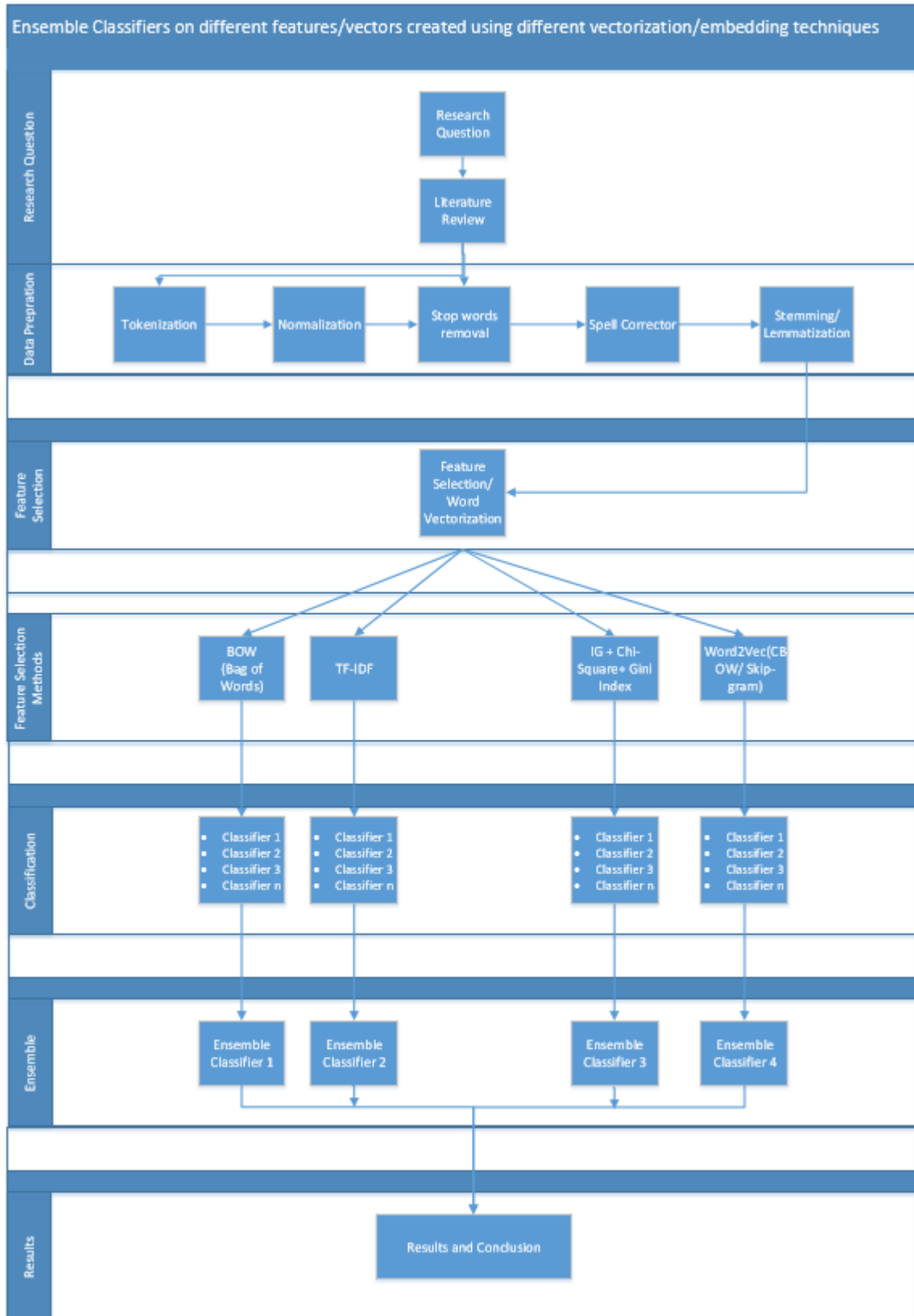
## **2.8 Doaa Mohey El-Din Mohamed Hussein (2018)**

**A survey on Sentiment Analysis Challenges** - Doaa Mohey El-Din Mohamed Hussein discusses the importance and effects of sentiment analysis challenges in sentiment evaluation based on two comparisons among forty-seven papers. The first comparison is based on the relationship between the sentiment review structure and the sentiment analysis challenges. The result of this comparison reveals another essential factor to recognize the sentiment challenges, which is domain-dependence. Moreover, the negation challenge became popular in all types of reviews structured just differs in implicit or explicit meaning. This comparison result provides a facility for the effects of each sentiment challenge on the review structure types. Authors conclude that the topic of nature and the review structure determine the suitable challenges for the evaluation sentiment reviews. Then the second comparison relies on the sentiment analysis challenges relevant to the accuracy rate. Their results present the importance of sentiment challenges in evaluating the sentiments and how to select the fitting challenge to improve accuracy. Authors find the relationship between the proportion of sentiment techniques usage in theoretical and technical types to solve sentiment challenges. Another result explains that the hot area of research is a theoretical type of sentiment challenge. That reflects on the results of the average of accuracy based on the number of researches in each challenge. The more the research in a sentiment challenge, the less the Average of accuracy rate. Overall, this research paper summarizes the keys of sentiment challenges concerning the type of review structure. It also divides the challenges into two types to ease to deal with them and focus on the degree of accurate meaning. This research discusses these sentiment challenges, the factors affecting them, and their importance [8].

Here in this research, Spacy Library is used to reduce word variation/word redundancies. The base classifiers (Naïve Bayes, SVM, Logistic Regression, Decision Tree) and the ensemble model is trained on different feature subset created using different word vectorization/ word embedding techniques. The objective of this research is to identify whether the change in features extraction/vectorization techniques improves the performance of the base classifiers and ensemble classifiers significantly. This study is helpful for the user to understand the significance of techniques like word vectorization/feature engineering and ensemble model for improvement in sentiment classification.



## The Framework of the Research Project



### 3.Methodology:







#### 3.1 The source of the data

Mobile Reviews data is scrapped from publicly available review websites Pricebaba and Mouthshut, using Beautiful Soup and urllib library. The scraped data parsed from the HTML file to CSV format. There are 290967 rows and 14 columns in the data set.

Created column is 'ReviewUpdated': Concat of 'Review' + 'Good'/'Bad' for *PriceBaba* + 'Title' for *Mouthshut*.

The main important columns considered for the analysis are Product, Review Updated, Rating, and Year.

#### Snapshot of the data in HTML format:

	000de216da73da3b8004668f7eb5c5be	07-07-2018 18:28	HTML File	530 KB
	00a9ffb9afa55818704d9bad8a1176eb	08-07-2018 00:15	HTML File	534 KB
	00a76f37bb42aa71d3808d94f722fd1f	07-07-2018 22:57	HTML File	545 KB
	00a3157cee36af03a263cb011b96927a	07-07-2018 23:34	HTML File	536 KB
	00ac62f5b722072b6ab98a9659c19352	07-07-2018 20:47	HTML File	533 KB
	00b2cddd4876614e7747d9c72f82b3af	07-07-2018 23:49	HTML File	534 KB

#### Snapshot of Data

Product	Review	Review Updated	Rating	Date	Title	Good	Bad	Website	GoodBad	TitleGoodBad	ProductPrice	ProductPriceRange	Year
Xiaomi Rei	Before starting the rei	Yet another Value	3	Jul 17, 2016 10:59 P	Yet another Value for Money prc			MouthShut		Yet another Valu	9999	Budget Range	2016
Xiaomi Rei	Marvelous Mobile just Awesome	Marvel	5	Jul 17, 2016 10:42 P	Awesome			MouthShut		Awesome	9999	Budget Range	2016
Xiaomi Rei	REDMI NOTE 3*====/ Power pack review		5	Jul 17, 2016 06:18 P	Power pack review on Redmi No			MouthShut		Power pack review	9999	Budget Range	2016
Xiaomi Rei	This phone looks verr	Flagship smartfon	4	Jul 17, 2016 04:44 P	Flagship smartfone under 13000i			MouthShut		Flagship smartfor	9999	Budget Range	2016
Xiaomi Rei	Master piece in such a	Boon for mobile	5	Jul 17, 2016 12:32 P	Boon for mobile			MouthShut		Boon for mobile	9999	Budget Range	2016
Xiaomi Rei	Very good phone havi	Best phone ever i	4	Jul 17, 2016 12:21 P	Best phone ever in the range of :			MouthShut		Best phone ever i	9999	Budget Range	2016
Xiaomi Rei	I have used many pho	Best mobile	5	Jul 17, 2016 08:54 A	Best mobile			MouthShut		Best mobile	9999	Budget Range	2016
Xiaomi Rei	Xiaomi Redmi Note 3 :Xiami redmi note		5	Jul 17, 2016 08:52 A	Xiami redmi note 3 sports a 2.2 i			MouthShut		Xiami redmi note	9999	Budget Range	2016

#### 3.2 Data Preparation

1. Reviews are classified as Positive (1) and Negative (0). The reviews having 1 Star and 2 Stars ratings are labeled to 0, i.e., Negative. Rating more than 2 Stars are labeled to 1, i.e., Positive. (Rating scale-1 to 5)
2. Data has 290967 rows. The data is filtered for the products which have at least 500 reviews and for the year 2016, 2017,2018.
3. Data is explored to have a general understanding of the data.

### Top Products by No of Reviews:



Figure 1

- **Minimum of 500 Reviews** over the 3 Years - 2016, 2017, and 2018.
- Xiaomi Redmi Note3 is having the highest No of Reviews (above 12000) followed by **Samsung Galaxy J2 6 2016 Edition**.

### Top Products by Highest Average Rating at Product level:

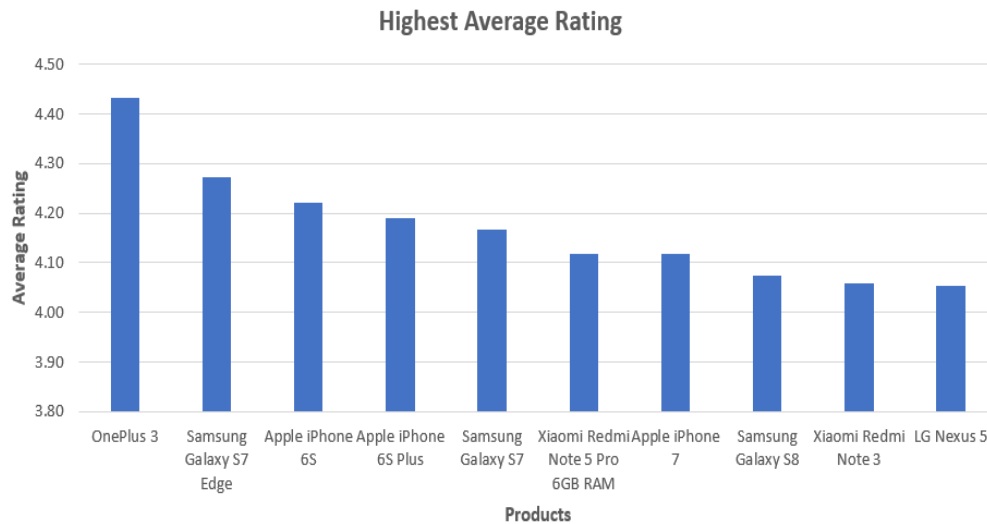


Figure 2

- **OnePlus3** is having the highest Average Rating, followed by **Samsung Galaxy S7 Edge**.

### Top Products by Lowest Average Rating at Product level:

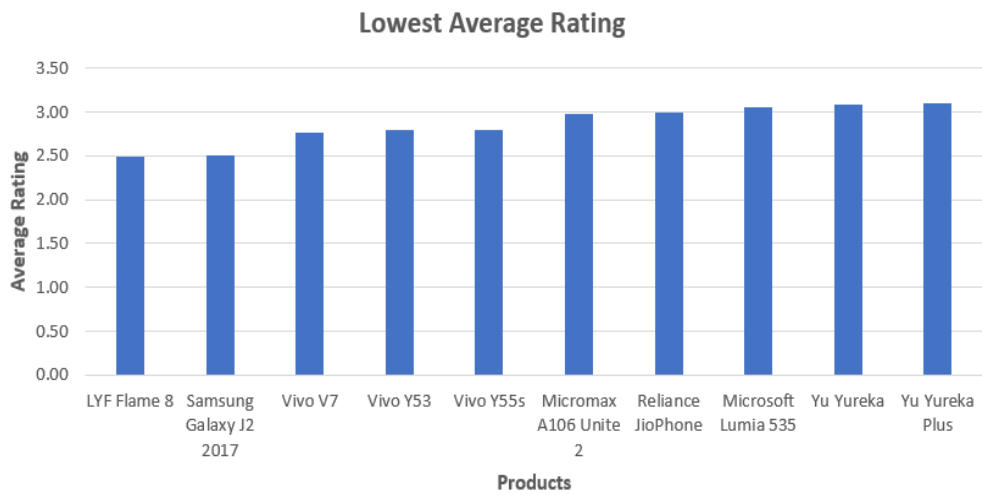


Figure 3

- **LYF Flame 8** is having the lowest Average Rating followed by **Samsung Galaxy J2 2017**

### Top Products by Highest Average Sentiments:

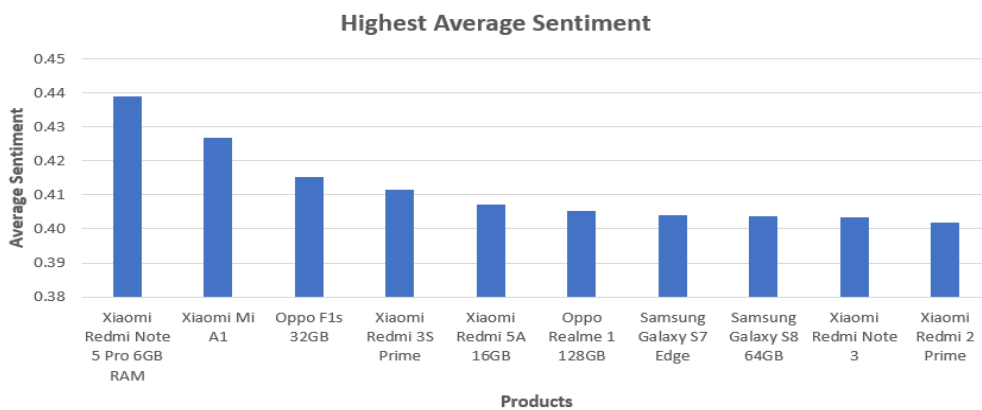


Figure 4

- **Xiaomi Redmi Note 5 Pro 6GB RAM** has the highest Average Sentiment followed by **Xiaomi Mi A1**.

### Top Products by Lowest Average Sentiments:

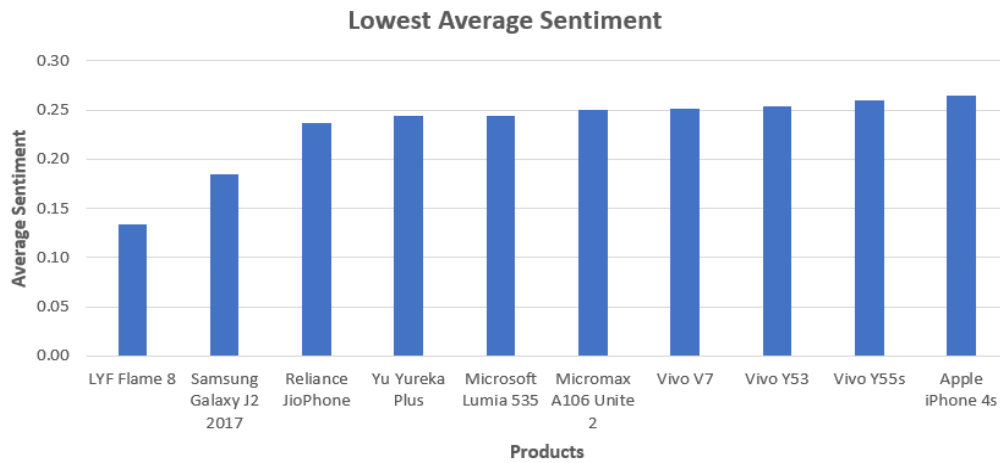


Figure 5

- **LYF Flame 8** has the lowest Average Sentiment, followed by **Samsung Galaxy J2 2017**.

### Top Products by Positive Sentiments (count):

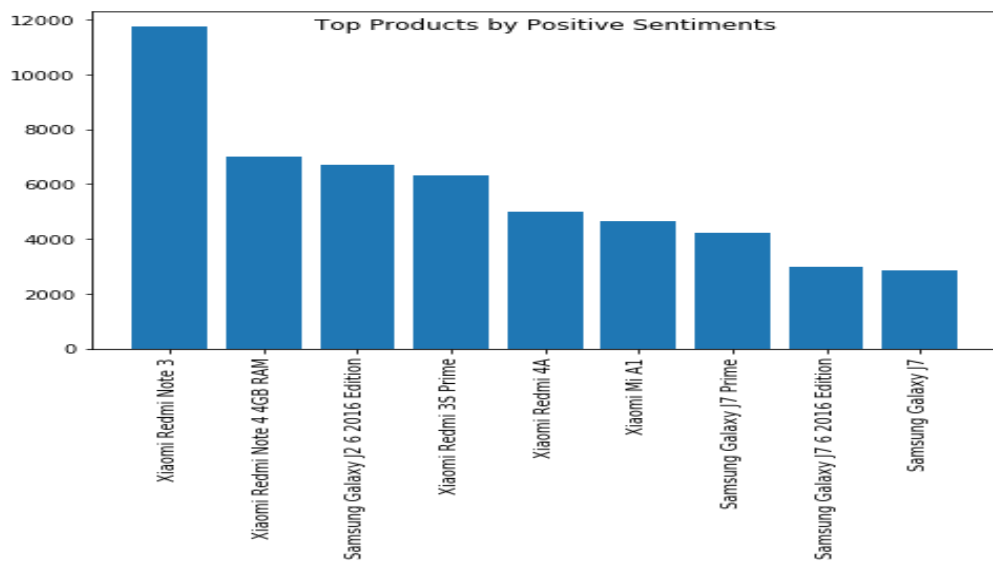


Figure 6

- **Xiaomi Redmi Note 3** has the highest No of Positive Sentiments, followed by **Xiaomi Redmi Note 4 4GB RAM**.

### Top Products by Negative Sentiments (count):

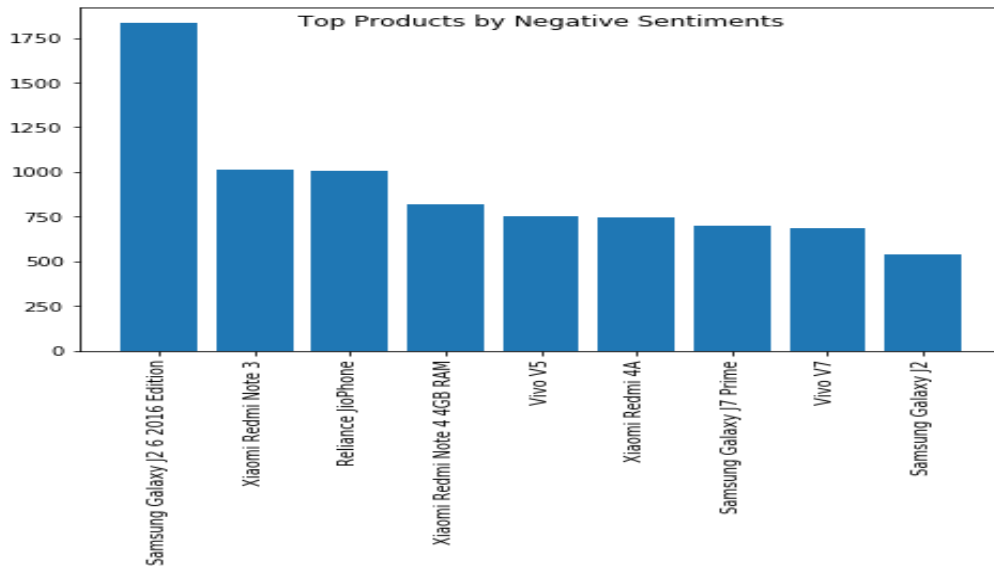


Figure 7

- **Samsung Galaxy J2 6 2016 Edition** has the highest No of Negative Sentiments followed by **Xiaomi Redmi Note 3**.
  - **Xiaomi Redmi Note 3** has a high no of **Positive** as well as **negative** sentiments.
4. I have used the "text blob" library to understand the sentiment Polarity of the text. Polarity lies in the range of  $[-1,1]$  where 1 means a positive statement, and -1 means a negative statement. As per the below figure, it is seen that average sentiment polarity is between 0.25 and 0.50 (high) for rating score "3", "4", and "5". Average Sentiment Polarity for rating score "1" and "2" is between 0 and 0.25 (moderate).

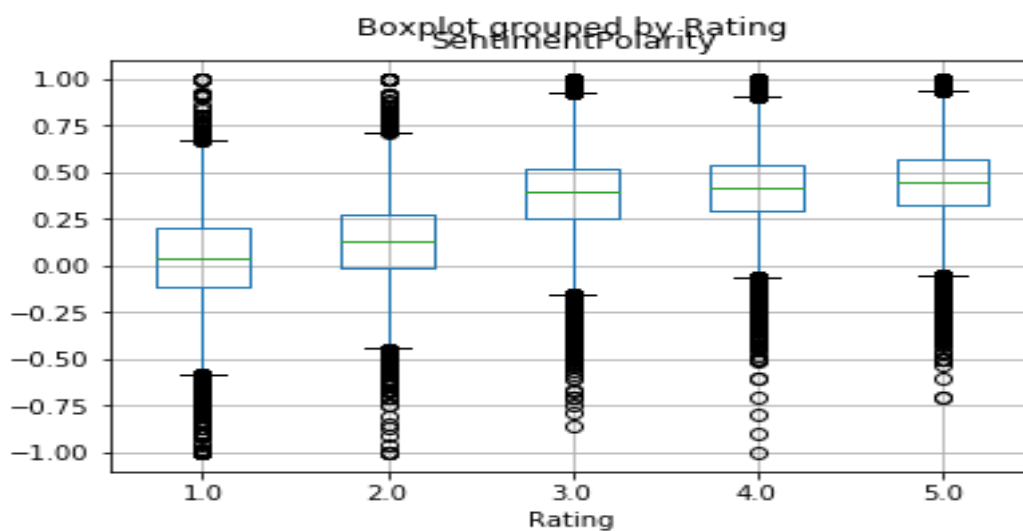


Figure 8

5. Spacy Library is used for tokenization, Part of Speech tagging, and Lemmatization. The data preprocessing tasks are performed to get the required clean filtered data for sentiment classification.
  - a. Tokenization - documents/text is split into a List of Tokens/words.
  - b. Tokens with POS tags like AUX, NUM, SYM, ADP, PRON are removed as they do not add any value for the sentiment analysis.
  - c. Lemma for tokens is used for further analysis. Lemmatization is the process of transforming all the tokens into the original form. Lemmatization is based on POS tagging. Here tokens are converted to their lemma using the Spacy library to reduce the word variance.
  - d. Stop words are common and high-frequency words that do not add any information about the context. Hence, stop words are removed along with new lines, extra white leading/trailing spaces, Special Characters to reduce the word variation. Numeric text is also removed as they do not add any value to the Sentiment Analysis.
6. Data is divided into Train and test split (70 and 30% respectively)

### 3.3 Features Selection/Word Vectorization

Text data is one of the most abundant sources of unstructured data. Text data usually consists of documents that can represent words, sentences, or even paragraphs of free-flowing text. The inherent unstructured and noisy nature of textual data makes it harder for machine learning algorithms to work on raw text data directly. Hence, the text data is converted to a one-hot encoded form that machines can understand and learn. In this research, multiple techniques like Bag of Words, TF-IDF, Chi-Square, Gini Index, and Information Gain are used for Features Engineering/selection/word vectorization.

A. **Bag of Words-** Bag of Words represents **the vocabulary of words**.

B. **TF-IDF Matrix-** The bag of words representation is a very naive way of representing text. It relies on just the word frequencies of the words of a document. TF-IDF matrix represents the word more smartly. It emphasizes how important the word is for the document in the corpus. It emphasizes the fact that some words in a document with high frequency do not add any specific information for the document as they are generic words appearing in most of the documents. In the Bag of words model, each word is assumed to be equally important.

C. **a) Information Gain (IG)-**

This statistical method is an effective solution for feature selection. In statistics, entropy is referred to like the randomness or the impurity in a group of classes. Information gain is a decrease in entropy. Information gain computes the difference between entropy (impurity) before splitting into the groups of classes and average entropy (impurity) after the split of the dataset/sample into the groups of classes based on a given attribute. Decision tree/Random Forest algorithm uses information gain for classification/Regression Problem. Information Gain offers a ranking of the features depending on their Information Gain score. A certain number of features can be selected quickly based on the ranking of Features (Ghosh and Sanyal, 2018). In this

research, this technique is used to filter the features based on feature importance calculated using the IG score.

#### **b) Chi-square ( $\chi^2$ )**

Chi-square ( $\chi^2$ ) is a statistical test, which can quantify the association between the feature and its related class. It tests a null hypothesis that the feature/word and class are wholly independent of each other. If the CHI value of feature for the class is high, then there is a closer relationship exists between the feature and class. The features with the highest  $\chi^2$  values for a category should perform best for classifying the documents (Ghosh and Sanyal, 2018). In this research, this technique is used to identify the features closely related to the target variable (Positive/Negative Sentiments)

#### **c) Gini Index (GI)**

Decision trees use the Gini Index of a feature/variable to decide to split a node into two or more sub-nodes. The decision tree splits the nodes on all available variables and then selects the split, which results in most homogeneous sub-nodes. Higher Gini Index value represents the higher homogeneity. The Gini Index measures the feature's ability to split the data into pure homogeneous groups of classes. Here in this research, the Gini index method is used for calculating the feature score in order to select features (Ghosh and Sanyal, 2018).

#### **d) Combined Features selection Method**

Each feature selection method applied with different rules to extract a feature subset, outcomes various feature subsets for the same dataset. The features from each feature subset are filtered features based on the feature importance threshold. These different feature subsets are merged by adopting the statistical method UNION, and then the common features, identified from the merged feature subset, are used as final features for the classifier (Ghosh and Sanyal, 2018).

- D. Word2Vec (Shallow Neural Network learns the embedding such that output words can be predicted using input words)** is a language modeling technique used for mapping words to vectors of real numbers. It reduces computational complexity along with vector dimensionality. It represents words in vector space with several dimensions. Word embedding represents the words that occur in a similar context in vector space based on the assumption that the words in a similar context are closer to each other. Word embeddings can be generated using **neural networks, co-occurrence matrix, probabilistic models**. In this research, Word2Vec from the genism model is tried for word embedding. Word2Vec consists of two models/architectures for generating word embedding 1) Continuous Bag of Words 2) Continuous Skip-gram model. These models are **shallow two-layer neural networks** having one input layer, one hidden layer, and one output layer (Ma, L., and Zhang, Y. ,2015).

**Continuous Bag of Words Model (CBOW)**- The CBOW model predicts the current word given context words within a specific window. The input layer contains the context words, and the output layer contains the current word. The hidden layer contains the number of dimensions in which we want to represent the current word present at the output layer (En.wikipedia.org. ,2020) & (GeeksforGeeks, 2020) & (Medium, 2020).



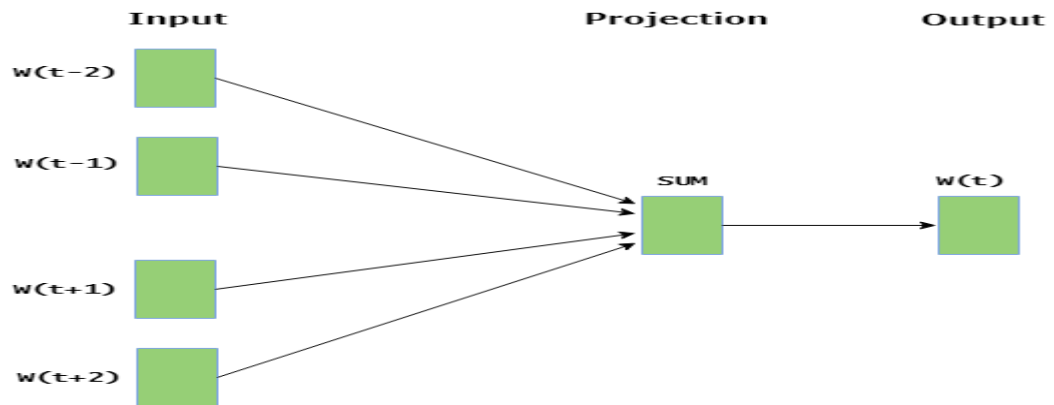


Figure 9

**Continuous Skip-gram Model-** The Skip-gram predicts the surrounding context words within a specific window given the current word. The input layer contains the current word, and the output layer contains the context words. The hidden layer contains the number of dimensions in which we want to represent the current word present at the input layer (En.wikipedia.org. ,2020) & (GeeksforGeeks, 2020) & (Medium, 2020).

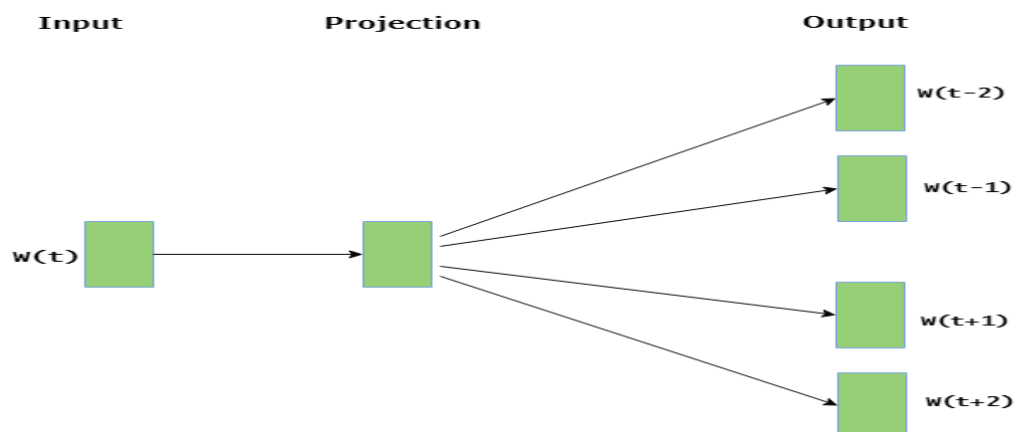
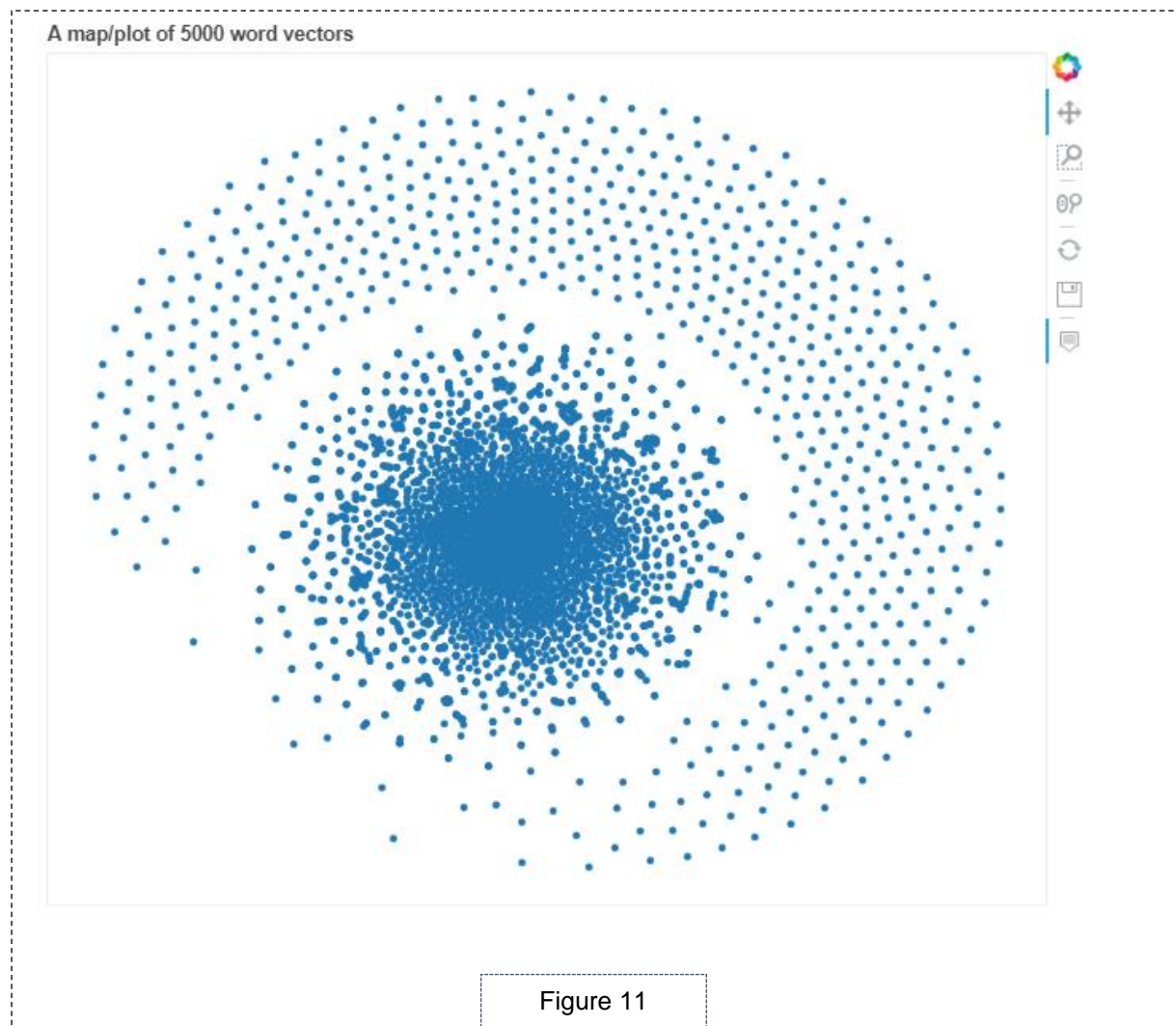


Figure 10



‘Bokeh’ library is used to get the plot of 5000 words vector. This plot shows how words of similar contexts are grouped in vector space.

### 3.4 Classification

Machine learning techniques are widely used in artificial intelligence and document classification. Extracted feature sets are used to train the classifier to classify the reviews as positive or negative. In this research, the Naïve Bayes, Logistic Regression, Decision Tree, and Support Vector classifiers are trained on Mobile Review dataset for each feature subset created using the mentioned word vectorization/ embedding techniques. Finally, the ensemble model is built by applying **weighted majority voting** on the base classifiers for each feature subset. In this research, the transition of the performance of the base classifiers and the ensemble model for sentiment classification is evaluated with the change in the word vectorization/embedding technique. The stability of each base classifier is evaluated using a 3-fold Cross-Validation technique. Weights are applied on base classifiers based on the stability factor and the performance of the base classifier. The results of the base classifiers are combined using weights assigned to get the ensemble model.

### **Naive Bayes (NB)**

Naïve Bayes classifier is based on the independence assumption where the joint probabilities of features and categories are used to calculate the probability score of categories of a given document. It is a probabilistic classifier. In this paper, we consider the Bio-nominal Naive Bayes classifier (Ghosh and Sanyal, 2018).

### **Support Vector Machine (SVM)**

Support Vector Machines (SVMs) are supervised learning models introduced for classification in both linear and nonlinear versions. If the data is linearly separable, then SVM uses hyperplanes to segregate the classes and will choose the hyperplane, which maximizes the distance (margin) between the nearest data points of either class. If datasets are non-linearly inseparable, then the SVM method tries to define a decision boundary with the hyper-planes in a high dimensional feature space. This high dimensional hyperplane separates the vectorized document into two classes as well as determines a result to make the decision based on this support-vector. The classical SVM seems to be able to separate the linear dataset with a single hyperplane, which can separate two classes. For the nonlinear dataset, kernel functions are used in that situation to layout the data to a higher dimensional space in which it is linearly separable (Ghosh and Sanyal, 2018).

### **Decision Tree**

The decision tree is the most popular tool for classification. It's easy to interpret and understand. As the name suggests, the decision tree is a tree-like structure. The decision tree tries to split the data into subsets of homogeneous classes, and every time chooses the attribute for splitting the data, which can create the most homogeneous subgroups. A tree can be *trained* by splitting the data set into subsets based on an attribute value test. The attribute with the highest Gini Index Score or Information Gain Score is chosen for the split. This process is repeated on each derived subset in a recursive manner. *The recursion/recursive partition* is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions (GeeksforGeeks, 2020). Decision trees can handle high dimensional data. In general, the decision tree classifier has good accuracy (Ghosh and Sanyal, 2018). However, the decision tree is prone to overfitting, which can be controlled by pruning.

### **Logistic Regression**

The logistic regression model consists of a set of classification rules extensively used for binary classification problems. In order to solve the multiclass problem, the model must be extended. This logistic function of this classifier extracts a set of weighted features from the input and estimates the correlation between the occurrence class and extracted features. Logistic regression becomes a suitable fit to the data by maximizing the log-likelihood function. Containing all predictor into a single model generally results in poor predictions. The proper variable selection makes the model more accurate and generalized (Ghosh and Sanyal, 2018).

### Ensemble Classifier

The stable classifiers are identified based on the performance of the K fold cross-validation technique for each of the feature subsets. Each of the base classifiers is assigned the weights based on performance and stability factor, and the results/predictions of these base classifiers are combined using the majority votes approach. The accuracy of these ensemble classifiers is evaluated for each feature subsets (Ankit and Saleena, 2018) & (Sadhasivam and Kalivaradhan, 2019).

## 4 Results

### Model Evaluation:

Models/Classifiers developed on each feature subset is evaluated based on three parameters: 1) Stability, 2) Accuracy (Ability of model to classify correctly, 3) F-Measure-which describes the ability of the model to classify the relevant class. The stability of each base classifier is evaluated by performing 3-fold cross-validation on data.

Accuracy: - In simple terms, accuracy represents the correct classification by Model.

Precision: - Precision determines the model capability in terms of predicting relevant class correctly out of a total no of predictions for the relevant class.

$\text{True Positives} / (\text{True Positives} + \text{False Positives})$

$\text{True Negatives} / (\text{True Negatives} + \text{False Negatives})$

Recall: - Recall determines the model capability to predict the relevant class correctly out of actual numbers of relevant class in the sample.

$\text{True Positives} / (\text{True Positives} + \text{False Negatives})$

$\text{True Negatives} / (\text{True Negatives} + \text{False Positives})$

F-Score:  $(2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$ - F-Score represents the arithmetic mean of Recall and Precision, which highlights the performance of the Classifier for the relevant class on top of overall performance in terms of accuracy.

Many times, Accuracy is not the only metric to rely on, especially when the data is not balanced. We need to verify the performance of the model with respect to relevant classes, which can be determined using metrics like Precision, Recall, and F-Measure.

In this research, different word vectorization techniques (Bag of Words, TF-IDF, Word2Vec model, Common Features selection using Chi-Square, Gini Index, and Information Gain) have been used on the same data. Different base classifiers (Logistic Regression, Support Vector Machine, Naïve Bayes and Decision Tree) and the ensemble classifier (combining the results of base classifiers by adding weights to the base classifiers based on the performance and stability factor) are trained on each feature subset. The objective of the research is to evaluate whether the performance of the base classifiers and the ensemble model significantly changes with the change in feature extraction /word vectorization technique. As seen below in table 4.1, 4.2, 4.3, and 4.4, the Logistic Regression and the Support Vector Machine performance is better and consistent compared to the other base classifiers for each feature subset. We have added 0.5 times weights for the weak

learners like the Naïve Bayes and the decision tree classifiers to get the Ensemble model based on the **weighted majority voting rule**.

### Naïve Bayes (NB)

**The average accuracy of the Naïve Bayes classifier across all feature subsets is 85%. However, the Naïve Bayes classifier performance is low in the case of the prediction of negative sentiments. The average precision and recall for Naïve Bayes classifier with respect to negative sentiments classification across all feature subsets is 61 and 60%, respectively. Naïve Bayes show low recall for the Bag of Words and TF-IDF vectors. It shows low Precision for the feature subset created based on feature importance (using Chi-Square, Gini Index and Information Gain), and the Word2Vec Vector.**

Naïve Bayes	Bag of Words		TF-IDF		Common Features selection using Gini Index, IG and Chi2		word2vec	
	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive
Precision	72%	91%	72%	91%	59%	93%	40%	93%
Recall	50%	96%	50%	96%	66%	91%	72%	79%
F-Measure	59%	93%	59%	93%	62%	92%	51%	86%
Accuracy	89%		89%		87%		78%	

Table 4.1

### Support Vector Machine (SVM)

In this research, the Linear Support Vector Classifier is trained on each feature subset created using different word vectorization techniques. **The average accuracy of the Linear SVM classifier across all word vectors/feature subsets is 90%.** However, the **Linear SVM classifier performance is low for the classification of all the negative sentiments. The average recall for the Linear SVM classifier with respect to the negative sentiments classification across all word vectors is 56.3%.** The Linear SVM Classifier **shows improvement in overall performance** with respect to recall, precision, and F-Measure **with the change in feature subset.** The Linear SVM Classifier **performs better for the word2vec** compared to the other word embeddings/vectorization techniques. Recall for SVM is improved to 59% for negative sentiments classification with the Word2Vec vector.

SVM	Bag of Words		TF-IDF		Common Features selection using Gini Index, IG and Chi2		word2vec	
	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive
Precision	70.0%	92.0%	77.0%	92.0%	81.0%	91.0%	80.0%	92.0%
Recall	58.0%	95.0%	57.0%	97.0%	51.0%	98.0%	59.0%	97.0%
F-Measure	63.0%	94.0%	66.0%	94.0%	63.0%	94.0%	68.0%	95.0%
Accuracy	89.0%		90.0%		90.0%		91.0%	

Table 4.2

### Decision Tree

In this research, the Decision Tree Classifier is trained on each feature subset created using different word vectorization techniques. **The average accuracy of the Decision**

**Tree classifier** across all word vectors/feature subsets is **84.3%**. However, the Decision Tree classifier **performance is low for the classification of all the negative sentiments. The average recall and precision for the Decision Tree classifier with respect to negative sentiment classification** across all word vectors is **48.3% and 51%, respectively**. The Decision Tree does not show improvement in overall performance (including recall, precision, and F-measure) with the change in the feature subset.

Decision Tree	Bag of Words		TF-IDF		Common Features selection using Gini Index, IG and Chi2		word2vec	
	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive
Precision	57.0%	90.0%	53.0%	90.0%	50.0%	90.0%	44.0%	89.0%
Recall	49.0%	93.0%	49.0%	91.0%	49.0%	91.0%	46.0%	89.0%
F-Measure	53.0%	92.0%	51.0%	91.0%	50.0%	90.0%	45.0%	89.0%
Accuracy	86.0%		85.0%		84.0%		82.0%	

Table 4.3

### Logistic Regression

In this research, the Logistic Regression Classifier is trained on each feature subset created using different word vectorization techniques. **The average accuracy of the Logistic classifier** across all word vectors/feature subsets is **90.3%**. However, the Logistic classifier performance is low for the classification of all the negative sentiments. **The average recall of the Logistic classifier for negative sentiments classification** across all word vectors is **55.8%**. The Logistic Classifier **shows improvement in overall performance** for recall, precision, and F-Measure **with the change in feature subset**. The Logistic Classifier **performs better for the word2vec** vector compared to the other word embeddings/vectorization techniques. Recall for the Logistic Regression is improved to 60% for negative sentiments classification with the Word2Vec vector.

Logistic Regression	Bag of Words		TF-IDF		Common Features selection using Gini Index, IG and Chi2		word2vec	
	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive
Precision	76.0%	92.0%	81.0%	92.0%	81.0%	91.0%	79.0%	93.0%
Recall	58.0%	97.0%	54.0%	97.0%	51.0%	98.0%	60.0%	97.0%
F-Measure	66.0%	94.0%	65.0%	94.0%	62.0%	94.0%	68.0%	95.0%
Accuracy	90.0%		90.0%		90.0%		91.0%	

Table 4.4

The logistic regression and the Support Vector Classifiers are the strong learners based on the performance as per Table 4.1, 4.2, 4.3, and 4.4, and the stability factor assessed using the 3-fold cross-validation technique.

### Ensemble Classifier

**The average accuracy of the Ensemble classifier** across all feature subsets is **91%**. However, the **Ensemble classifier performance is improved for the classification of**

the negative sentiments in comparison to the base classifiers. The average recall for the Ensemble Classifier for negative sentiments classification across all word vectors is 58%, whereas the average recall for the strong learners like the Logistic Regression and the Support Vector classifier is 55.8 and 56.3% respectively. The average precision for the Ensemble Classifier for negative sentiments classification across all word vectors is 78%, whereas the average precision for the strong learners like Logistic Regression and SVM is 79.3 and 77%, respectively. Ensemble Classifier, which combines the results of the weak and the strong learners, has improved the F-Measure for negative sentiments classification compared to the strong learners like Logistic Regression and SVM across all feature subsets. You also see the improvement in the results of the ensemble classifier with the change in feature subsets. **Ensemble Classifier performs better for word2vec compared to other word embeddings/vectorization techniques.**

Ensemble	Bag of Words		TF-IDF		Common Features selection using Gini Index, IG and Chi2		word2vec	
	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive
Precision	75%	93%	79%	92%	80%	91%	79%	93%
Recall	61%	96%	57%	97%	53%	97%	60%	97%
F-Measure	67%	94%	66%	95%	64%	94%	68%	95%
Accuracy	90%		91%		90%		91%	

Table 4.5

SVM	Bag of Words		TF-IDF		Common Features selection using Gini Index, IG and Chi2		word2vec	
	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive
Precision	70.0%	92.0%	77.0%	92.0%	81.0%	91.0%	80.0%	92.0%
Recall	58.0%	95.0%	57.0%	97.0%	51.0%	98.0%	59.0%	97.0%
F-Measure	63.0%	94.0%	66.0%	94.0%	63.0%	94.0%	68.0%	95.0%
Accuracy	89.0%		90.0%		90.0%		91.0%	

Table 4.2

Logistic Regression	Bag of Words		TF-IDF		Common Features selection using Gini Index, IG and Chi2		word2vec	
	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive
Precision	76.0%	92.0%	81.0%	92.0%	81.0%	91.0%	79.0%	93.0%
Recall	58.0%	97.0%	54.0%	97.0%	51.0%	98.0%	60.0%	97.0%
F-Measure	66.0%	94.0%	65.0%	94.0%	62.0%	94.0%	68.0%	95.0%
Accuracy	90.0%		90.0%		90.0%		91.0%	

Table 4.4

As per Table 4.1,4.2,4.3 and 4.4, **the logistic and the Support Vector classifier** are the strong learners compared to the Decision tree and the Naïve Bayes classifiers. The logistic and the Support Vector classifiers are low on recall measure for negative



sentiments classification across all feature subsets. However, the recall of the logistics and SVM classifier is improved to 60% and 59% respectively for word2vec compared to Bag of Words (58% both), TF-IDF (LR-54% and SVM-57%) and the feature subset created based on feature importance (LR-51% and SVM-51%). As a result, the F-Measure of the logistics and SVM Classifier is improved to 68% (both) for word2vec compared to Bag of Words (LR-66% and SVM-63%), TF-IDF (LR-65% and SVM-66%), and the feature subset created based on feature importance (LR-62% and SVM-63%).

In this research, we have used 4 base classifiers -1) the Naïve Bayes, 2) the Decision Tree, 3) the Logistic Regression, and 4) the Support Vector Machine. The stability of each classifier is tested using a 3-fold cross-validation technique for all feature subsets. The Logistic regression and the Support vector classifier are identified as the strong learners based on the performance of the base classifiers with respect to measures like accuracy, recall, and precision. We have used the weighted majority voting approach to combine the results of the base classifiers for the Ensemble model. We have added 0.5 weights for each weak learner (Naïve Bayes and Decision Tree Classifier). The Ensemble model will classify the text sentiment as Positive Sentiment 1) if two strong classifiers (the Logistic Regression and the Support Vector Machine) classify the text sentiment as 'Positive' or 2) if the one strong learner (the Logistic Regression or the Support Vector Machine) and two weak learners (the Naïve Bayes and the decision tree classifier) classify the text sentiment positive. Similarly, the Ensemble model will classify the text sentiment as Negative Sentiment in two scenarios. 1) if two strong classifiers (the Logistic Regression and the Support Vector Machine) classify the text sentiment as 'Negative' or 2) if the one strong learner (the Logistic Regression or the Support Vector Machine) and two weak learners (the Naïve Bayes and the decision tree classifier) classify the text sentiment Negative. As shown in Table 4.6, the precision of the Ensemble Classifier for positive and negative classification is like the strong learner 'Logistic Regression' for each feature subset. However, as shown in Table 4.7, the recall of the ensemble Classifier for negative classification is improved significantly for each feature subset (Bag of Words-61%, TF-IDF-57%, the feature subset based on features importance-53% and Word2Vec-60%) compared to the strong learner - the Logistic Classifier. The recall of the Logistic classifier (the strong learner) for negative classification is 58% for Bag of Words, 54% for TF-IDF vector, 51% for feature subset based on features importance, and 59% for the word2vec vector. As a result, you see the improved accuracy of the Ensemble model for each feature subset compared to the base classifiers in Table 4.9 (Bag of Words-90%, TF-IDF-91%, the feature subset based on features importance-90% and word2vec-91%).



Negative Sentiments					Positive Sentiments				
Precision	Bag of Words	TF-IDF	Common Features selection using Gini Index, IG and Chi2	word2vec	Precision	Bag of Words	TF-IDF	Common Features selection using Gini Index, IG and Chi2	word2vec
Support Vector Machine	70.0%	77.0%	81.0%	80.0%	Support Vector Machine	92.0%	92.0%	91.0%	92.0%
Logistic Regression	76.0%	81.0%	81.0%	79.0%	Logistic Regression	92.0%	92.0%	91.0%	93.0%
Naive Bayes	72.0%	72.0%	59.0%	40.0%	Naive Bayes	91.0%	91.0%	93.0%	93.0%
Decision Tree	57.0%	53.0%	50.0%	44.0%	Decision Tree	90.0%	90.0%	90.0%	89.0%
Ensemble	75.0%	79.0%	80.0%	79.0%	Ensemble	93.0%	92.0%	91.0%	93.0%

Table 4.6

Negative Sentiments					Positive Sentiments				
Recall	Bag of Words	TF-IDF	Common Features selection using Gini Index, IG and Chi2	word2vec	Recall	Bag of Words	TF-IDF	Common Features selection using Gini Index, IG and Chi2	word2vec
Support Vector Machine	58.0%	57.0%	51.0%	59.0%	Support Vector Machine	95.0%	97.0%	98.0%	97.0%
Logistic Regression	58.0%	54.0%	51.0%	60.0%	Logistic Regression	97.0%	97.0%	98.0%	97.0%
Naive Bayes	50.0%	50.0%	66.0%	72.0%	Naive Bayes	96.0%	96.0%	91.0%	79.0%
Decision Tree	49.0%	49.0%	49.0%	46.0%	Decision Tree	93.0%	91.0%	91.0%	89.0%
Ensemble	61.0%	57.0%	53.0%	60.0%	Ensemble	96.0%	97.0%	97.0%	97.0%

Table 4.7

Negative Sentiments					Positive Sentiments				
F1-Measure	Bag of Words	TF-IDF	Common Features selection using Gini Index, IG and Chi2	word2vec	F1-Measure	Bag of Words	TF-IDF	Common Features selection using Gini Index, IG and Chi2	word2vec
Support Vector Machine	63.0%	66.0%	63.0%	68.0%	Support Vector Machine	94.0%	94.0%	94.0%	95.0%
Logistic Regression	66.0%	65.0%	62.0%	68.0%	Logistic Regression	94.0%	94.0%	94.0%	95.0%
Naive Bayes	59.0%	59.0%	62.0%	51.0%	Naive Bayes	93.0%	93.0%	92.0%	86.0%
Decision Tree	53.0%	51.0%	50.0%	45.0%	Decision Tree	92.0%	91.0%	90.0%	89.0%
Ensemble	67.0%	66.0%	64.0%	68.0%	Ensemble	94.0%	95.0%	94.0%	95.0%

Table 4.8

Accuracy	Bag of Words	TF-IDF	Common Features selection using Gini Index, IG and Chi2	word2vec
Support Vector Machine	89.0%	90.0%	90.0%	91.0%
Logistic Regression	90.0%	90.0%	90.0%	91.0%
Naïve Bayes	89.0%	89.0%	87.0%	78.0%
Decision Tree	86.0%	85.0%	84.0%	82.0%
Ensemble	90.0%	91.0%	90.0%	91.0%

Table 4.9

Accuracy of the Support Vector Machine, the Logistic Regression, and the Ensemble model is high for the word2vec vector compared to all the other feature subsets.

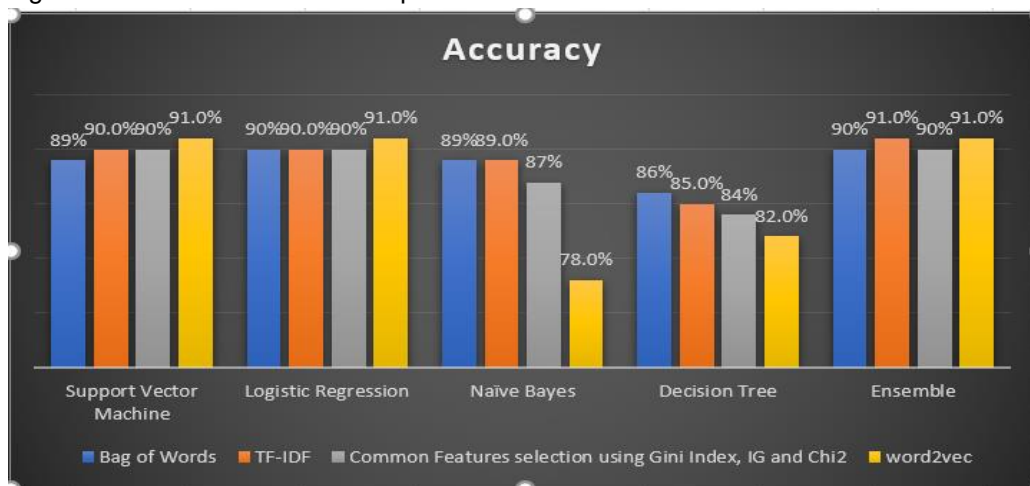


Figure 12

The Support Vector Machine, the Logistic Regression, and the Ensemble model have a relatively high recall for Positive sentiments classification across all feature subsets.

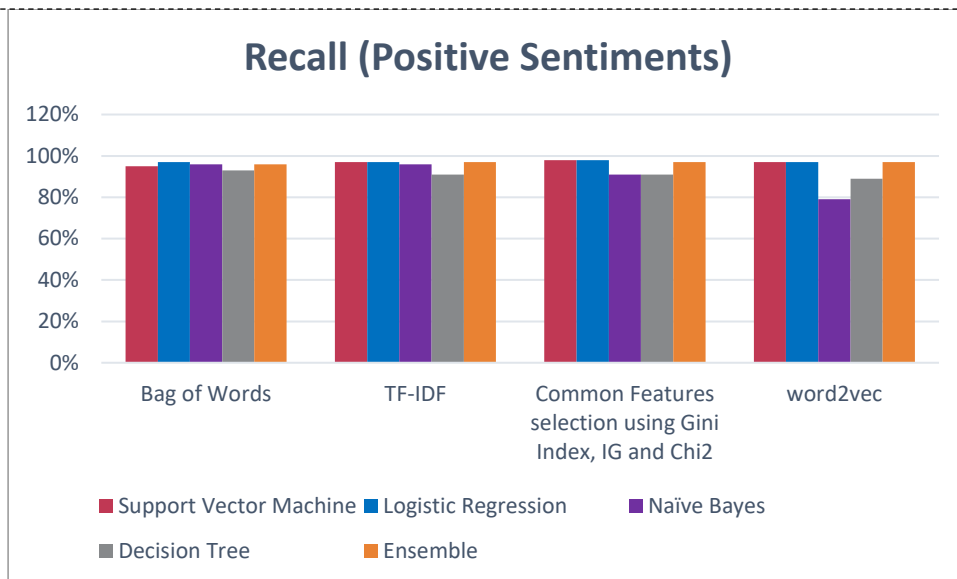


Figure 13

The strong learners like the Support Vector Machine, the Logistic Regression, and the Ensemble model has a relatively high recall for negative sentiments classification on the word2vec vector compared to the other feature subsets.

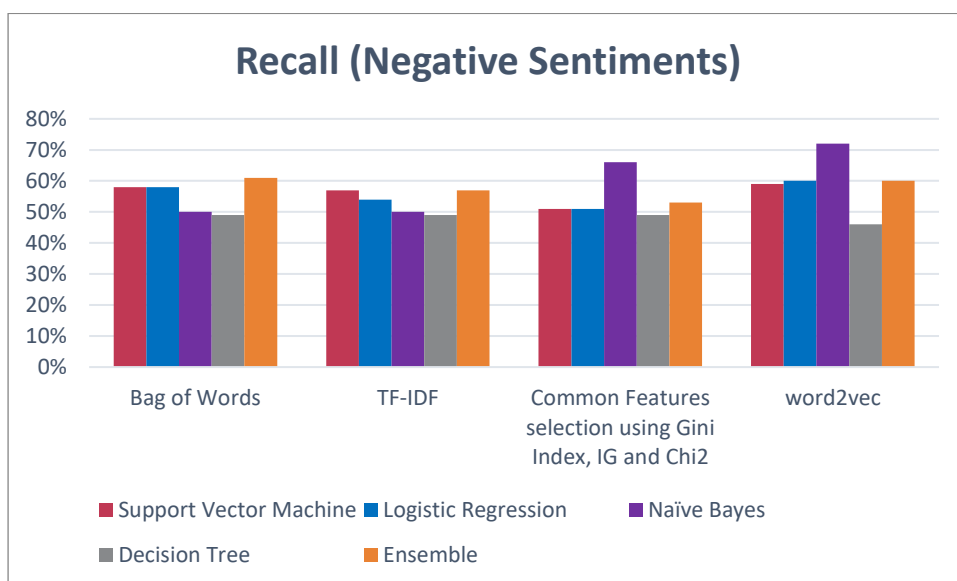


Figure 14

The Support Vector Machine classifier, the Logistic Regression classifier, and the ensemble model have an improvement in the precision for positive and negative classification with the change to the word2vec model.

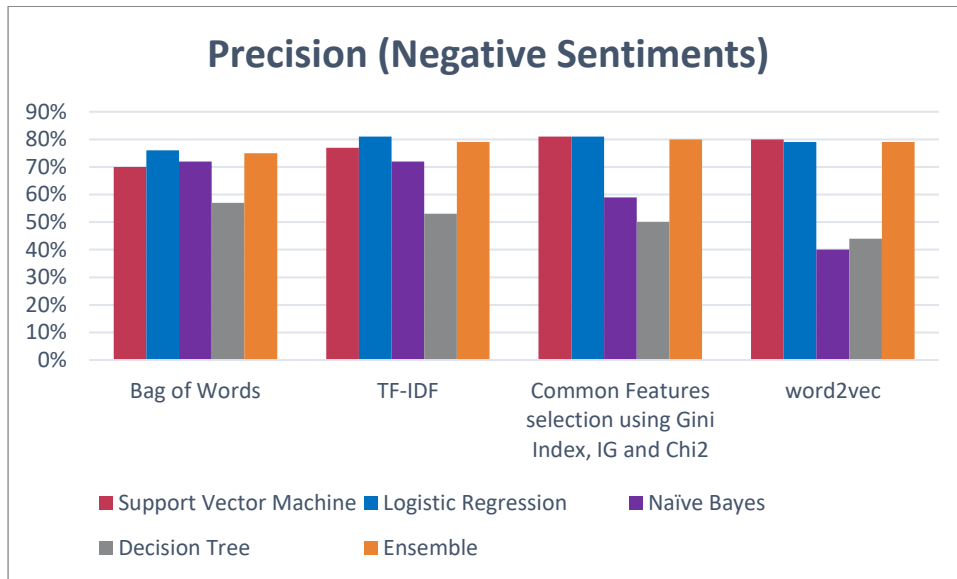


Figure 15

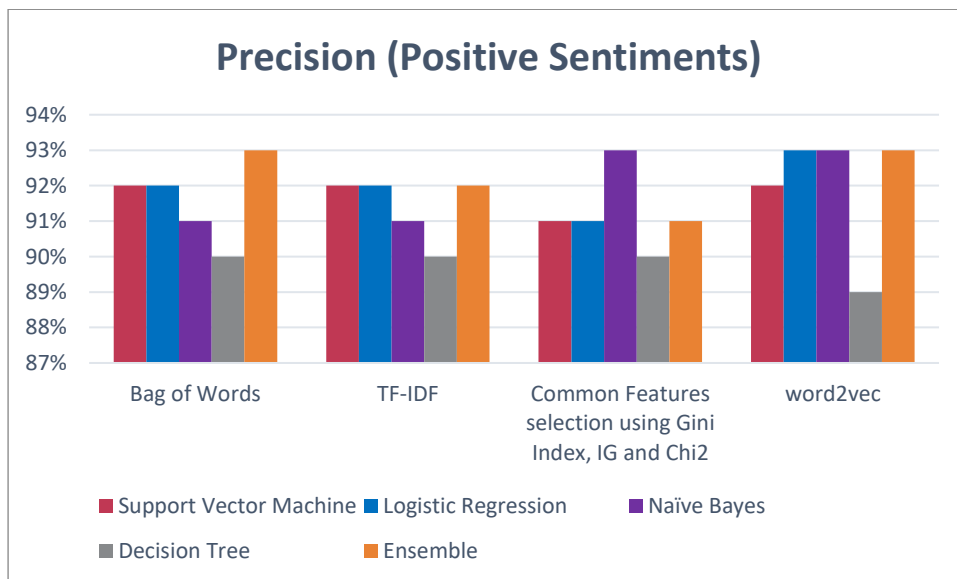


Figure 16

The Support Vector Machine classifier, the Logistic Regression classifier, and the ensemble model have an improvement in the F-Measure for positive and negative classification with the change to the word2vec model.

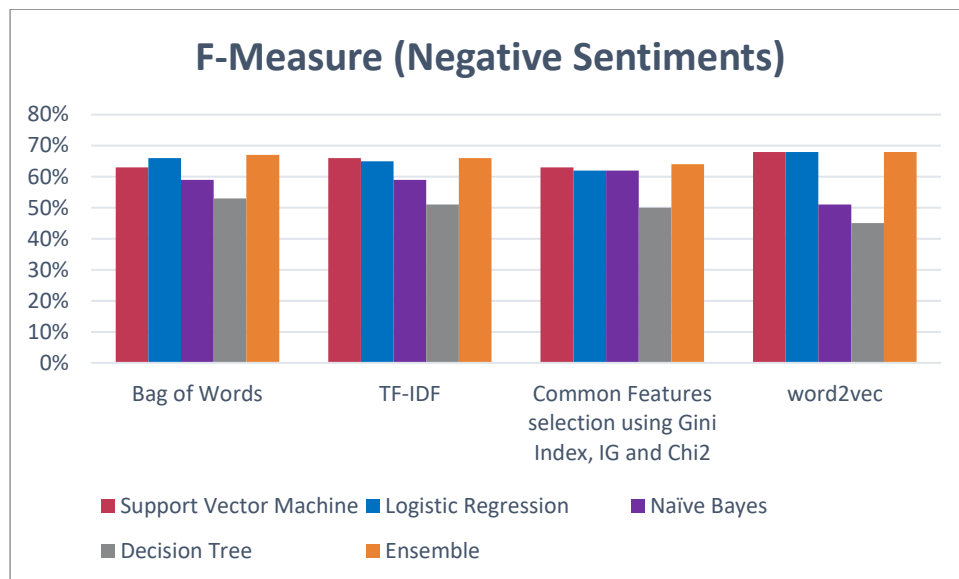


Figure 17

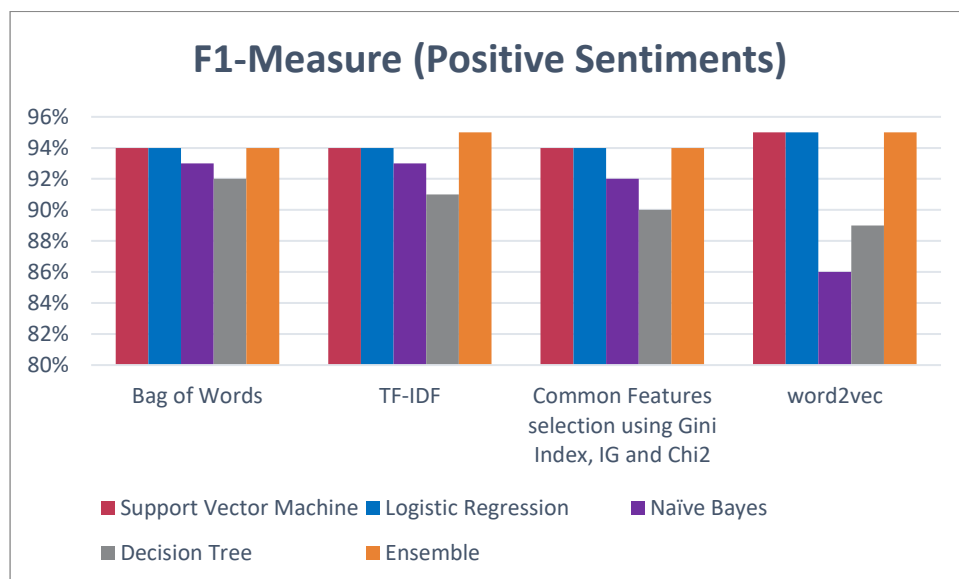


Figure 18

## 5 Conclusion:

### Summary of results and contribution

- In many research papers, an ensemble classifier is identified as one of the approaches to improve the performance of the base classifiers. Bag of Words, TF-IDF, Features selection based on Features Importance (using Gini index, Information gain, and Chi-Square methods) are the feature vectorization techniques cited in many research papers to improve the performance of the base classifiers. Recently word embedding technique like Word2Vec is used to improve the performance of the base classifier in many research papers. Word Embedding is a language modeling technique that reduces computational complexity along with vector dimensionality. It represents words in vector space with several dimensions. In short, there are research papers where the Ensemble model built on the base classifiers and these base classifiers are trained on a single feature subset to improve the performance of the base classifiers. There are research papers where word vectorization techniques like creating a common feature subset based on feature importance are used on different datasets to analyze the performance of the different base classifiers. In many research papers, the Word2Vec technique (a shallow neural network) is tried to analyze the performance of the base classifier for the sentiment classification. In many research papers, the ensemble model or the use of word vectorization/word embedding technique is cited as one of the approaches to improve the performance of the classifier for sentiment classification. However, these two popular approaches (ensemble model and the feature engineering) are not tried together.

In this research, the combination of the ensemble approach and different word vectorization/embedding techniques is tried to evaluate the significance of the change in performance of sentiment classification with the change in feature extraction techniques. We have also evaluated the performance of the respective ensemble model, where the results of all the base classifiers for each feature subset are combined. It is observed that there is an improvement in overall performance (including recall, precision, accuracy) of the strong classifiers/learners with word2vec vector compared to the other feature subsets. As observed, there is a significant improvement in the negative sentiment classification (recall) with the combination of the strong base learners and the Word2Vec-word vectorization technique. The accuracy of the ensemble classifier is in line with the accuracy of the strong classifier. In contrast, there is an improvement in the recall for negative classification of the ensemble classifier compared to the strong base learner -the Logistic Regression. It is seen that there is a significant improvement on low performance parameters of the strong base classifiers with the change in the word vectorization technique Word2Vec. There is also the improvement in the low performance parameters of the base classifiers with the ensemble model. The combination of the word2vec model and the ensemble model shows the best performance parameters for the sentiment classification on top of the combination of the other word vectorization techniques with the base classifiers and ensemble approach.

However, the improvement in the performance of the sentiment classifier is not significant with the change to the word2vec vector and the ensemble approach.

## **Future Work**

Recently Deep Learning is an evolving technique in NLP. Deep learning is a type of machine learning, where features are not created manually. Instead, the data is fed into the deep learning algorithm, and it automatically learns the features most useful to determine the output. Deep Learning works on the principle of the neural network, which resembles a human brain and automatically learns what features are useful. Standard deep learning algorithms include convolutional neural networks (CNNs), recurrent neural networks, and deep Q networks.

In this research, the combination of the different word vectorization techniques (Bag of Words, TF-IDF, Features selection based on feature importance and Word2Vec architecture) and the ensemble model (combined results of the base classifiers based on weighted majority voting) are used. However, the improvement in the performance of the sentiment classifier is not significant with the change to the word2vec vector and the ensemble approach. In future work, the Deep Learning applications can be implemented to see whether the performance improves significantly compared to the performance of the combination of the word2vec model and the ensemble approach as used in this research paper.

## REFERENCES







1. Ankit and Saleena, N. (2018). An Ensemble Classification System for Twitter Sentiment Analysis. *Procedia Computer Science*, 132, pp.937-946.
2. Devika, M., Sunitha, C., and Ganesh, A. (2016). Sentiment Analysis: A Comparative Study on Different Approaches. *Procedia Computer Science*, 87, pp.44-49.
3. En.wikipedia.org. (2020). *Word embedding*. [online] Available at: [https://en.wikipedia.org/wiki/Word\\_embedding](https://en.wikipedia.org/wiki/Word_embedding) [Accessed 10 Feb. 2020].
4. Fang, X. and Zhan, J. (2015). Sentiment analysis using product review data. *Journal of Big Data*, 2(1).
5. GeeksforGeeks. (2020). *Decision Tree - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/decision-tree/> [Accessed 10 Feb. 2020].
6. GeeksforGeeks. (2020). *Python / Word Embedding using Word2Vec - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/python-word-embedding-using-word2vec/> [Accessed 10 Feb. 2020].
7. Ghosh, M., and Sanyal, G. (2018). An ensemble approach to stabilize the features for multi-domain sentiment analysis using supervised machine learning. *Journal of Big Data*, 5(1).
8. Hussein, D. (2018). A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, 30(4), pp.330-338.
9. Ma, L., and Zhang, Y. (2015). Using Word2Vec to process big text data. *2015 IEEE International Conference on Big Data (Big Data)*.
10. Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), pp.1093-1113.
11. Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), pp.1093-1113.
12. Medium. (2020). *NLP 101: Word2Vec—Skip-gram and CBOW*. [online] Available at: <https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314> [Accessed 10 Feb. 2020].
13. Sadhasivam, J., and Kalivaradhan, R. (2019). Sentiment Analysis of Amazon Products Using Ensemble Machine Learning Algorithm. *International Journal of Mathematical, Engineering and Management Sciences*, 4(2), pp.508-520.



## 7. Appendices:

Code Repository:- <https://github.com/halapets/Sentiment-Analysis-LJMU>

### Snapshot of the data in HTML format:

 000de216da73da3b8004668f7eb5c5be	07-07-2018 18:28	HTML File	530 KB
 00a9ffb9afa55818704d9bad8a1176eb	08-07-2018 00:15	HTML File	534 KB
 00a76f37bb42aa71d3808d94f722fd1f	07-07-2018 22:57	HTML File	545 KB
 00a3157cee36af03a263cb011b96927a	07-07-2018 23:34	HTML File	536 KB
 00ac62f5b722072b6ab98a9659c19352	07-07-2018 20:47	HTML File	533 KB
 00b2cddd4876614e7747d9c72f82b3af	07-07-2018 23:49	HTML File	534 KB

```
[!DOCTYPE html]<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:fb="http://www.facebook.com/2008/fbml" xml:lang="en" lang="en" class="no-js
s1280"><head><meta name="dcterms.rightsHolder" content="Pricebaba.com, Copyright (c)
2018"/><meta name="pb.intz" content="5th July 2018 07:38:43 PM"/><link rel="manifest"
href="/manifest.json"><meta name="msapplication-config" content="none"/><link rel="search"
type="application/opensearchdescription+xml" href="/opensearchdescription.xml" title="Pricebaba
Search: Find the Best Prices for Mobiles, Laptops & other Electronics in India"/><meta http-
equiv="X-UA-Compatible" content="IE=edge,chrome=IE8"><link rel="chrome-webstore-item"
href="https://chrome.google.com/webstore/detail/niklpdinbckehpkpopjgfcjgjhjdch"><!--[if lt IE
9]><link rel="stylesheet"
href="https://passets.pricebaba.com/assets/components/ie/ie.css"><![endif]--><!--[if lt IE 9]><script
src="http://html5shiv.googlecode.com/svn/trunk/html5.js" ></script><![endif]--><meta
property="fb:app_id" content="378589355588183"/><meta property="fb:admins"
content="1789882767"/><title>Micromax X279i Price In India, Buy at Best Prices Across Mumbai,
Delhi, Bangalore, Chennai & Hyderabad</title><link rel="shortcut icon" type="image/png"
href="https://passets.pricebaba.com/assets/application/images/favicon/favicon.dcaca7ec6338.ico"/
><meta name="keywords" content="Micromax X279i, Micromax X279i price, Micromax X279i
review, compare Micromax X279i, Micromax X279i specifications, Micromax mobile phones,
Micromax X279i features"/><meta name="description" content="Pricebaba brings you the best
price & research data for Micromax X279i. Look at latest prices, expert reviews, user ratings, latest
news and full specifications for Micromax X279i. You can also compare Micromax X279i with other
mobiles, set price alerts and order the phone on EMI or COD across Bangalore, Mumbai, Delhi,
Hyderabad, Chennai amongst other Indian cities. Details for this Micromax phone were last updated
on 5th July 2018."/><link rel="canonical" href="https://pricebaba.com/mobile/micromax-
x279i"/><link rel="amphtml" href="https://pricebaba.com/lite/mobile/micromax-x279i"/><meta
name="og:title" property="og:title" content="Micromax X279i Price In India, Buy at Best Prices
Across Mumbai, Delhi, Bangalore, Chennai & Hyderabad"/><meta name="og:site_name"
property="og:site_name" content="https://pricebaba.com"/><meta name="og:image"
property="og:image"
content="https://md3.pricebaba.com/images/product/mobile/1366/micromax-x279i-
raw.jpg"/><meta name="og:url" property="og:url"
content="https://pricebaba.com/mobile/micromax-x279i"/><meta name="og:description"
```

## Snapshot of the code to convert HTML data to CSV format:

```
''' WHOLE DATA '''

pricebabaFilesHTML = glob.glob('D:/EPBAPProject/Pricebaba/PricebabaHTMLData/*.HTML')

''' Read the contents of the reviews into a list . Note encoding should be mentioned as utf-8 otherwise gives an error. Note 'r' needs to
be included as well otherwise gives an error. '''

pricebaba_html = []
for filename in pricebabaFilesHTML:
    with open(filename, 'r', encoding='utf-8') as f:
        pricebaba_html.append(f.read())

''' Reviews - Getting 6470 reviews. '''

PriceBabaReviewsHTML = re.findall(r'<div class="p-h-m"><span>(.*?)</span>', str(pricebaba_html))

''' ProductName. '''

PriceBabaProductHTML = re.findall(r'User Reviews for (.*?) \(', str(pricebaba_html))

''' Rating '''

PriceBabaRatingHTML = re.findall(r'<div class="usr-rvw__rtng-wrpr"><div class="cui-rating" title="Rated (.*?) out',
str(pricebaba_html))

PriceBabaRatingHTML = [ int(x) for x in PriceBabaRatingHTML]

''' Date - Name of the reviewer is also included. '''

PriceBabaDateHTML = re.findall(r'<div class="m-t-m \"><span>(.*?)</span>', str(pricebaba_html))

''' Combining Product Name,Rating,Date, Review with the help of | in Regex'''

PriceBabaCombined = re.findall(r'User Reviews for (.*?) \(\<div class="usr-rvw__rtng-wrpr"><div class="cui-rating" title="Rated (.*?)
out\<div class="m-t-m \"><span>(.*?)</span>\<div class="p-h-m\><span>(.*?)</span>', str(pricebaba_html))

''' Converting tuple to dataframe '''

PriceBabaAllCombinedDF= pd.DataFrame(PriceBabaCombined)
```

## Snapshot of the data in CSV format:

Product	Review	Review Updated	Rating	Date	Title	Good	Bad	Website	GoodBad	TitleGoodBad	ProductPrice	ProductPriceRange	Year
Xiaomi Rei	Before starting the rei	Yet another Value	3	Jul 17, 2016 10:59 P	Yet another Value for Money prc	MouthShut				Yet another Valu	9999	Budget Range	2016
Xiaomi Rei	Marvelous Mobile just	Awesome Marvel	5	Jul 17, 2016 10:42 P	Awsome			MouthShut		Awsome	9999	Budget Range	2016
Xiaomi Rei	REDMI NOTE 3*====/	Power pack review	5	Jul 17, 2016 06:18 P	Power pack review on Redmi No	MouthShut				Power pack revie	9999	Budget Range	2016
Xiaomi Rei	This phone looks verr	Flagship smartfon	4	Jul 17, 2016 04:44 P	Flagship smartfone under 13000i	MouthShut				Flagship smartfor	9999	Budget Range	2016
Xiaomi Rei	Master piece in such a	Boon for mobile	5	Jul 17, 2016 12:32 P	Boon for mobile			MouthShut		Boon for mobile	9999	Budget Range	2016
Xiaomi Rei	Very good phone havi	Best phone ever i	4	Jul 17, 2016 12:21 P	Best phone ever in the range of :	MouthShut				Best phone ever i	9999	Budget Range	2016
Xiaomi Rei	I have used many pho	Best mobile I hav	5	Jul 17, 2016 08:54 A	Best mobile			MouthShut		Best mobile	9999	Budget Range	2016
Xiaomi Rei	Xiaomi Redmi Note 3 :	Xiami redmi note	5	Jul 17, 2016 08:52 A	Xiami redmi note 3 sports a 2.2 i	MouthShut				Xiami redmi note	9999	Budget Range	2016