Predicting Usefulness Of Justdial Reviews Using Machine Learning Techniques

Aditi Dubey B.E. (I.T.)

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Data Science)

Supervisor: Dr.Bahman Honari

August 2019

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Aditi Dubey

August 12, 2019

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Aditi Dubey

August 12, 2019

Acknowledgments

I am very much thankful to my supervisor Dr.Bahman Honari from School of Computer Scinece and Statistics at Trinity College Dublin (IE) who has been always very helpful and motivating during this whole journey. I am very grateful to him for his expert guidance all the time and polishing my knowledge.
I would also like to thank Prof. John Dingliana and Prof. John Waldron from Trinity College Dublin as the second reader of my thesis and I am very grateful for their valuable comments on my thesis which gave me a way to improvise.
I am very thankful to my parents for always supporting me and making me the person i am this day. I am very lucky and thankful to my friends who are my family away home, Pinki and Nakul, who continuously motivated and supported me during this whole journey of my degree.
Last but not the least, I am thankful to God for blessing me with good mental and physical strength to achieve my goals.

Aditi Dubey

University of Dublin, Trinity College
August 2019

Predicting Usefulness Of Justdial Reviews Using Machine Learning Techniques

Aditi Dubey, Master of Science in Computer Science

University of Dublin, Trinity College, 2019

Supervisor: Dr.Bahman Honari

Online platform for reviews and feedback plays a very important role for any business growth in this era of internet world. Justdial is a local search engine from India which enables users to search and use services of various businesses like AC repairers, automobile shops, book shop, movie tickets, doctor, floweriest, Chinese restaurant etc. on the basis of users location or search preferences. It also enables users to write the reviews and provide the ratings to the businesses which they have used as per their experience. On the basis of these reviews and ratings, other users get the help in making the decision to choose a service provider. If the next user likes the service of the business as stated by the review of first user and that review proved helpful to him, it will give like to that review and further more users will access those likes and refer that review to choose a better service for themselves. Review with good number of likes (usefulness) will play a very important role in businesss growth as well as for users. Justdial or businesses would like to promote these type of reviews to increase their businesses among consumers. However it might take ample number of days or months to get the likes (usefulness feedback) on the reviews hence there is a need to automatically predict the usefulness of the reviews.

This work is mainly focused on predicting the usefulness of Justdial reviews on the basis of various factors such as review texts, date, star ratings, usefulness (likes

frequency) using various machine learning algorithms like RandomForest, Gradient Boosting Model, Extreme Gradient Boosting Model, Light GBM and Long short term memory (deep learning algorithm) . This research will analyze the performance of these various machine learning models on this type of business problems. This piece of work also carried out some pre-processing with the data and attempt to analyze the correlation between various factors to get a better understanding of significance of these factors on our models. Implementation of natural language processing in python language has been done to analyze the text of the reviews before applying the machine learning models and have attempted to train the models with better accuracy. The results of this research has been evaluated using precision, recall, F1 score and Root Mean Square Error (RMSE). After training various models, it can be concluded that XGBM gave the best results among all with 78.20% accuracy, 78% precision, 99% recall and 88% F1 score.

Summary

Justdial being Indias no.1 search engine serves as a vast ocean of research in the field of data analysis. Review can change the decision of a consumer and hence a future of a business can grow or fall on the basis of reviews, ratings and feedbacks.

Online platforms are excellent method of advertisement as well as to reaching out directly to the consumer in this era of online shopping and usage of online services in almost every domain around the world. This field serves as a treasure to the researchers who are working on data analysis domain as well as machine learning field. This enables and encourage students and researchers to deep dive in this area and try to resolve these type of business problems with the help of machine learning models in an innovative way. These type of datasets and researches provides researchers to do data mining and visualize the data in a way on which business decisions can be made. Several reports can be generated on the basis of these type of datasets which might be very helpful in making big business decisions.

Justdial enables the user to upvote or downvote a review if it is helpful or not helpful respectively. However a user can read a limited number of reviews before making his or her decisions. Sometimes relevant reviews could not get much visits and upvotes and hence they skip out of the many users attention [1]. A research done by [2] reveals that if these usefulness criteria influence the display order of the reviews on a website

in such a way that good reviews are among top 25% list of the reviews display order for a business and hence would be promoted can be extremely helpful for building a business reputation and its expansion. However, for some least searched businesses or some new reviews does not have much upvotes and hence they are considered outliers in our dataset.

In this research, usefulness of a review is predicted on the basis of weighted useful score. Usefulness score is calculated on the basis of number of days from the current date to the review posting date. This is for the normalization of the usefulness in terms of time considered. The length of the review as well as bag of good and bad words is also taken into picture to determine the sentiments of the review writer. Correlation of star rating with usefulness is also considered as an important factor for this research.

NLTK library of python is used for natural language processing of the review text. After pre-processing of data such as removing bigrams, special characters, stop words, invalid letters, smiley emotions, tokenizing of words, lemmatize the word (eat, ate, eaten) etc. a new set of data is prepared and various machine learning models are applied. In the dataset, if a review has zero upvotes, then it is considered as not useful.

In this dissertation, Random Forest Classifier, Gradient Boosting Model, Extreme Gradient Boosting Model, Light Gradient Boosting model and Long Short Term Memory Model (LSTM) has been implemented. Evaluation of these machine learning models are done using precision, recall and F1 score and Root Mean Square Error (RMSE). Results are concluded on the basis of accuracy percentage of the all the models.

This research will help researcher to understand, analyze and visualize the data for such type of business problems and their expansion as well as to evaluate the per-

formance of these machine learning models on such type of business data and problems.

This research will also be beneficial to the business owners for promoting their excellent reviews in top 25% reviews and influencing the users decision. Consumers will get the benefit of going through direct feedback, review and rating from another consumer and hence choosing a better service for himself.

Contents

List of Tables

List of Figures

Chapter 1

State of the Art

1.1 Project Background and motivation

Justdial is Indias no. 1 local search engine and it is being used by unlimited number of youths/all age groups people these days. Justdial is experiencing exponential growth in its users. It provides services through android/iOS application, website and phone call center (+91-8888888888). It enables users to get the information immediately on their cell phone through a text message even if user dont have internet access.

- Justdials services connect sellers of products and services with potential buyers/users.
- 131.3 million quarterly unique visitors in Q2FY19.
- High user engagement, 88 million ratings and reviews.
- Database of 23.8 million listings.
- Approximately 470,620 active paid campaigns [3]

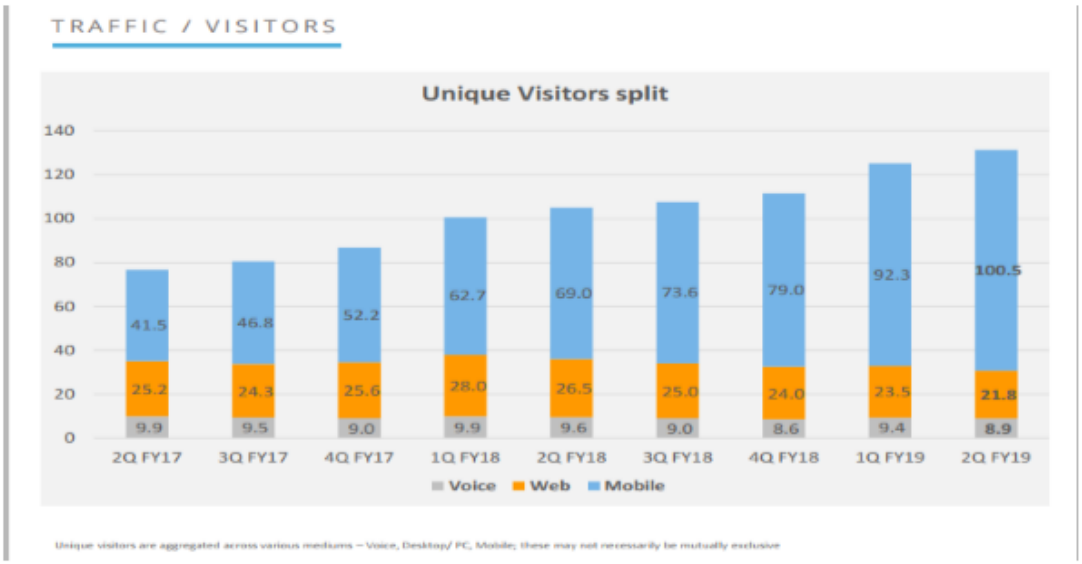


Figure 1.1: Unique visitors of JustDial in the given duration [3]

With the increasing use of smart phones and internet around the world, services and service providers on the internet are also growing exponentially. Internet is the best option to target the potential and genuine buyers. Internet is used widely for advertisement purposes. Everybody is just a little search away from their desirable services. When more and more people are using services online, it is very critical to get the best-fit out of so many options available online as well as avoiding fake, spam or vague services.

Reviews and ratings of other users play a very important role to get the best-fit for a consumer. In the dissemination of information, online reviews play a very essential role and influence user choice. However, before making a choice, a user can only read a limited number of reviews. This is very important aspect for the success of a rating and review site such as Justdial that to identify which reviews are useful to promote them.

Justdial introduced the feature of upvote and downvote a review to facilitate its users to get the best services from their genuine experience and direct feedback from other users. Using this upvote feature on the reviews, Justdial can promote these re-

views and reorder and place them on the top 25% reviews for the users and assist users to get the best services. This information will also benefit the business and service providers as well as Justdial will get more and more users. However, for some least searched businesses or latest reviews, there are no upvote or downvote information and hence prediction of usefulness of a review can be a game-changer. Prediction of usefulness is an interesting business problem solution for all the sources who are online service providers like Justdial.

Online communities take a lot of choices about their daily operations in this internet era, such as shopping, traveling, eating, etc. To this end, their choices rely strongly on ratings and reviews based on confidence. However, reviews from a large number of different users often result in a huge amount of variance in the quality of reviews. This could adversely influence the user's general experience and may even affect the community's trustworthiness [7]. Therefore, controlling the reviews will be presented and ordering to customers becomes a significant element. Also, as time goes by, the usefulness of a review also decreases. This study may assist Justdial India evaluate its company outlook. So, having a way to predict and rank the usefulness of reviews is essential in this internet world.

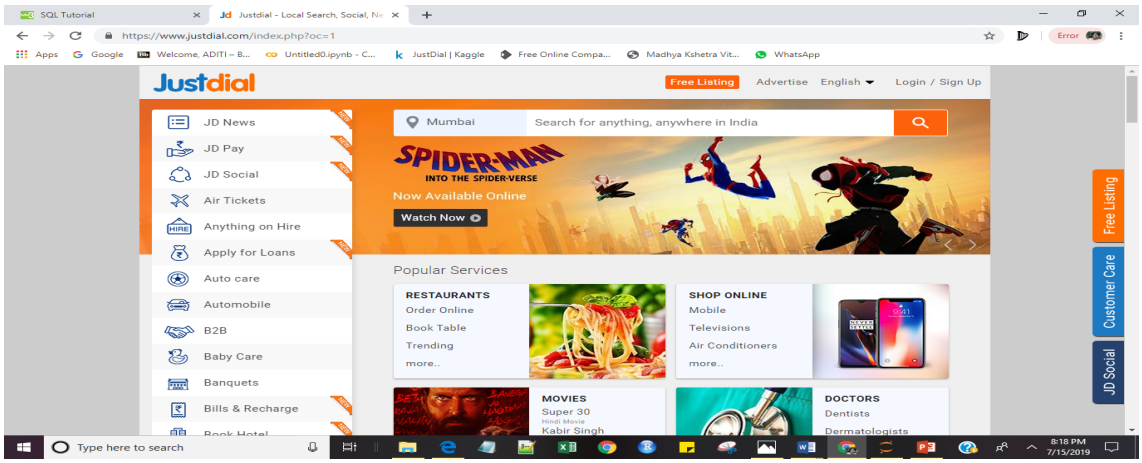


Figure 1.2: Website home page

Figure 1.3: Example of rating on website

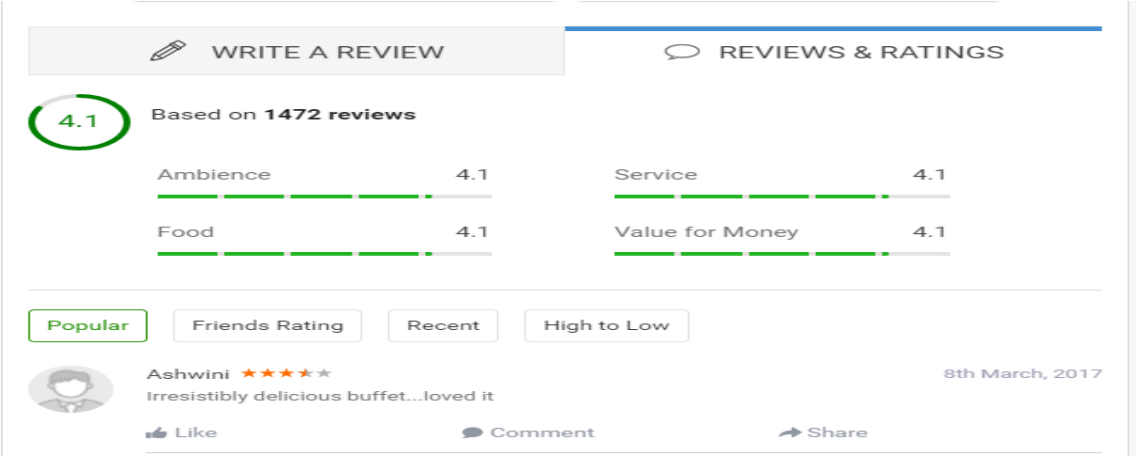


Figure 1.4: Example of review posted by a user

This research is an attempt to analyze and predict the usefulness or Upvotes on the Justdial reviews which have been written by users. Various techniques like polarity score, negative positive emotions extraction, usefulness per day (weighted usefulness), and machine learning algorithms have been used in this work. Features are also created wherever necessary. Various classification and regression algorithms are explored like Random Forest, Gradient boosting model , Extreme Gradient boosting model, Light gradient boosting model and Long Short Term Memory (Deep learning model) to predict how many Useful votes a review can receive. Root Mean Square Error, R

square and confusion matrix are used to analyze the model accuracy. Precision, recall and F1 score are used to analyze the results.

1.2 Research aim

The aim of this research is to predict the usefulness of reviews of the Justdial dataset by analyzing the impact on accuracy when applying machine learning algorithms in this type of business problem. Objectives of this research are:

- To evaluate the machine learning algorithm for this type of dataset and business problems.
- To develop a model which predict the reviews usefulness with better accuracy.
- To propose the resolution for such type of business problems and their direct impact on businesses across.

1.3 Research question

'Can a model be developed to predict the usefulness of reviews on the Justdial dataset using machine learning algorithms with better accuracy?'

Chapter 2

Literature review

Various methods and techniques have been previously applied to analyze the reviews, stars and their usefulness. Different classification and regression models are applied and analysis of the results have been done to predict the review usefulness and performance evaluation of machine learning models in this type of business problems has been also done.

Depending on the previous work done on this topic, analysis and conclusions have been made regarding the techniques used and implemented in this work. Advantages and disadvantages of using these models are also discussed further.

[8] used 3 algorithms, i.e. Logistic regression, K Nearest Neighbors (KNN), Support Vector Machine (SVM) and Random Forest. They found that the most efficient way to predict is to combine various features. Removing the KNN model's TF-IDF function does not greatly influence output. KNN's prediction was no better than logistic regression, but the prediction of SVM enhanced in comparison with the logistic regression. For better performance measurement, features with TF-IDF and without TF-IDF were experimented during the application of RandomForest. Accuracy of RandomForest model was comparatively better than the accuracy of Logistic Regression but TF-IDF could not improve the model. The best accuracy achieved is from RandomForest model without using TF-IDF features.

[2] used Quadratic Discriminant Analysis (QDA), Linear Discriminant Analysis (LDA), Logistic regression, Support Vector Machine (linear), Logistic regression (Lasso), Support Vector Machine (RBF) models, Nave Bayes, and then concluded that Logistic Regression (with Lasso) and SVM algorithms can be used to predict the usefulness of Yelp reviews. It also enables to fix the issue of overfitting and improved forecast by using PCA or the Lasso process.

[1] used Support for Vector Machine-Models of Linear, Polynomial, Radial Kernel and Random Forest. SVMs have been demonstrated to be susceptible to noisy data. Metadata characteristics and syntactic features didn't improve the SVM models, but these features worked well with the Random Forest, and lexical features didn't help either of the 2 models.

[9] implemented Random Forest Regression, LASSO regression, Negative Binomial Regression and zero inflated negative binomial regression. Author also implemented 4-fold cross validation on before training the model on training dataset. Random Forest model with tree depth 20 was optimum solution as more than 20 depth was overfitting the model and less than 20 were unfitting. In this work, TF-IDF were used and improved the accuracy of both the models however not by great differences. Best results achieved with LASSO regression using TF-IDF feature.

[7] applied various feature selection approaches like bag of words, bag of words with TF-IDF feature, bag of word with removed stop words and with stemming, bag of word with part-of-speech tagging and other linguistic feature in text. Linear regression with Ridge regression and LASSO regression models and SVM models were applied. Best results were achieved using Ridge regression. Various combinations of linguistic techniques performed really well and hence concluded as bag of words can be removed from further research.

[10] trained a Support Vector Machine model for the analysis and prediction of usefulness of the reviews. Author also did a detailed analysis on various features of the dataset and concluded that star rating, unigrams, bigrams, text length has a direct correlation with the usefulness. Author also removed stop words which proved benefi-

cial for the model and provided better accuracy. Structural features except length of the text were not proved so fruitful for this research.

This paper [11] displays a basic unsupervised learning algorithm for recommending reviews as useful or not useful. The classification and prediction is anticipated by the semantic orientation of the phrases in the review that contain adjectives or adverbs. An expression has a positive semantic orientation when it has great affiliations (e.g., "awesome") and a negative semantic orientation when it has bad affiliations (e.g., "exceptionally arrogant"). In this paper, the semantic orientation of a review is determined on the basis of calculations between given review with good and bad word. A review is considered as useful or recommended if the average semantic orientation is positive. The model accomplishes a prediction of up to 84% accuracy when research done on 410 reviews of 4 different domains. Calculation is based on PMI-IR (Pointwise Mutual Information and Information Retrieval) algorithm to determine the semantic orientation of the review text.

[12] implemented 3 standard machine learning algorithms which are Nave Bayes, Maximum Entropy and Support Vector Machine. Author used k-fold validation method and extracted various features like unigrams, bigrams, unigrams+bigrams, adjectives etc. Equal distribution (700 reviews) of negative and positive reviews were taken to implement the models. Unigrams performed really well. No stemming and stop were used in this research which should be used for better results. Author also suggested that bag-of-words did not perform as good as human can analyze the sentiments. This research concludes that Nave Bayes did not perform well however SVM was best but difference was not much.

[13] used IMDB dataset for implementation of long short term memory model (LSTM) for review classification. Most frequent words are used to train and analyze the model to avoid rare words for better training of model. LSTM model used input layer, embedding layer, LSTM layer and output layer which were set as length 1, length 32, 100 neurons and 1 neuron respectively. Loss and valloss as well as accuracy and valaccu has measured and plotted in a graph. The final model is trained till fifth epoch and the loss and accuracy of the final model on the validation set were 0.4366

and 0.8675, respectively. The results using the validation set were pretty good with the 86.75% accuracy.

2.1 Advantages of using various Machine learning model

Machine learning is one of the technique which is being used to solve various classification and regression problems in our day to day life. Machine learning models are being used for predictions in almost every domain all over the world. Best advantage of these machine learning models is to extract information out of large datasets which is impossible for a human eye to analyze large data without machine. It enables the programmer to explore various aspects in the data and compare and conclude as per the business needs. There are 2 types of machine learning models i.e. Supervised and Unsupervised model. Supervised models are very efficient in accuracy and can be implemented on small and large datasets. Algorithms such as Logistic regression, random forest, decision tree, gradient boosting etc. have been considered best in terms of accuracy.

Random Forest classifier which is also known as random decision forest model, is a common ensemble technique that can be used to construct predictive models for issues of classification and regression problems both. Ensemble techniques use various learning models to achieve better predictive outcomes in the case of a random forest model, the model produces a whole forest of random uncorrelated decision trees to get the best possible accuracy. [14]

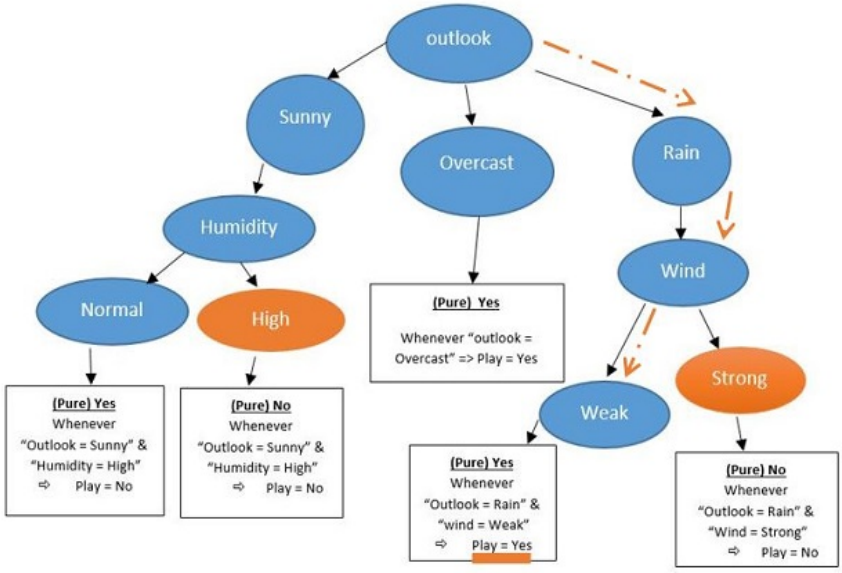


Figure 2.1: Random Forest Architecture

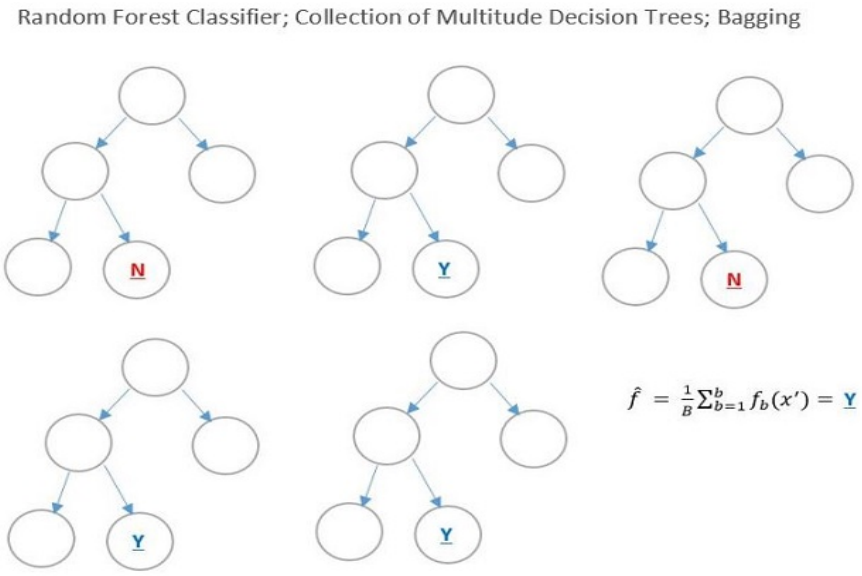


Figure 2.2: Random Forest Architecture

Gradient Boosting Machines (GBM) trains the model in a gradual manner. It gives more weight to weak learner tree and less weight to strong learner tree. When

the next tree gets trained, it takes the weight into consideration and hence GBM helps the model to optimize and get better accuracy by reducing loss function. Probably the greatest inspiration of utilizing Gradient boosting is that it enables one to advance a user defined determined cost function, rather than a loss function that generally offers less control and does not basically compare with real time applications. [15]

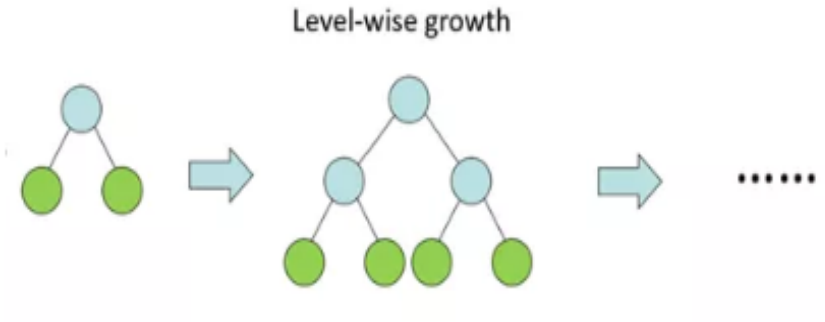


Figure 2.3: Gradient Boosting Architecture

XGB is demonstrated to be adaptable in almost every application and is particularly quicker than other machine learning algorithms. It additionally helps in dealing with inadequate information. It is an advanced form of GBM algorithm however is considerably more proficient as far as assets utilized and can likewise defeat the issue of overfitting without having negative impact on its productivity. [16]

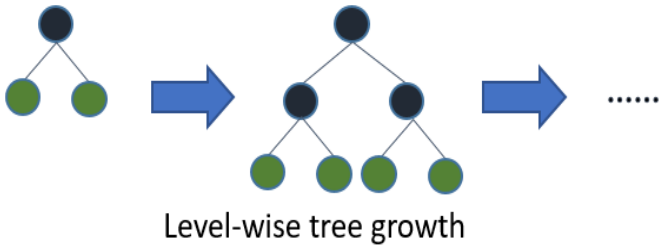


Figure 2.4: Extreme Gradient Boosting Architecture

Light GBM is a gradient boosting machine learning algorithm that utilizes tree based learning algorithms. Light GBM develops tree vertically while other algorithms develops trees on a level wise means that Light GBM develops tree leaf-wise while other algorithms develops level-wise. It will pick the leaf with max delta loss to develop. When developing a similar leaf, Leaf-wise algorithm can decrease more loss than a level-wise algorithm. The size of information or data is expanding day by day and it is getting to be hard for traditional data science algorithms to give quicker and accurate outcomes. Light GBM is prefixed as 'Light' in view of its fast. Light GBM can deal with the huge size of information and takes lower memory to run. Another reason of why Light GBM is well known is on the grounds that it focuses around the accuracy of results. LGBM also supports GPU learning and in this way researchers are broadly utilizing LGBM for data science application advancement. [17]

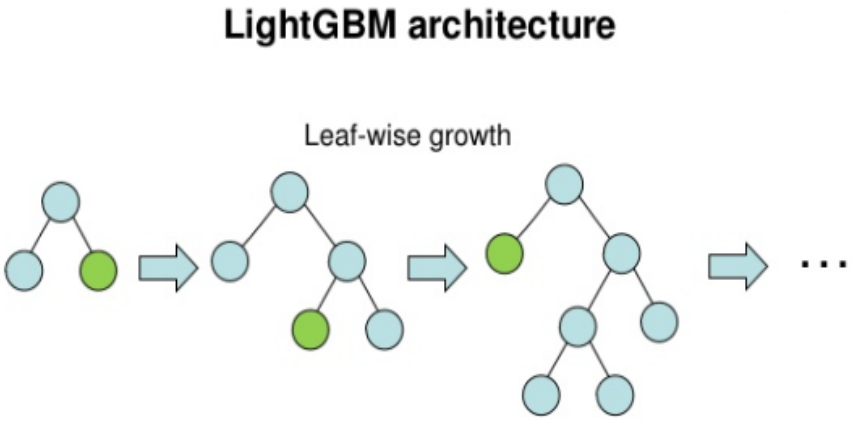


Figure 2.5: Light Gradient Boosting Architecture

LSTM: Long short term memory

Sequence prediction problems have been around for quite a while. They are considered as probably the most difficult issue to explain in the data science industry. These incorporate a wide scope of issues; from foreseeing deals to discovering patterns in stock markets, from understanding movie plots to perceiving your method for discourse, from language interpretations to anticipating your next word on your iPhone's or Android console. With the ongoing achievements that have been going on in data science, it is discovered that for almost all pattern prediction problems, Long short term Memory model, also known as LSTMs have been seen as the best and efficient resolution. LSTMs have an edge over traditional feed-forward neural systems and RNN from multiple points of view. This is a result of their property of specifically recollecting examples and patterns for long spans of time. [13]

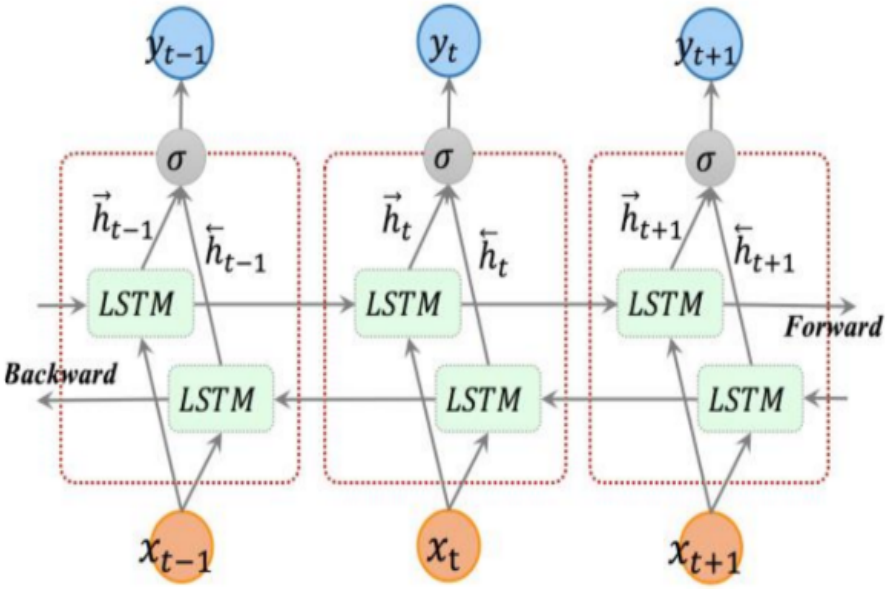


Figure 2.6: Long short term memory Architecture

Information extraction and sentiment analysis is very important these days as online platforms to express opinions are tremendously growing day by day. Analyzing the online reviews actually help to understand the mindset of general public and vibes. Blogs, tweets and review sites are the mirror of the society. These opinions can influence the decisions of the online buys and sellers.

In [18] it is expressed that Recurrent Neural Networks (RNN) are equipped for managing short-term dependencies in a sequence of data. However, RNNs experience difficulty when managing long term dependencies. These long term have an incredible inuence on the significance and polarity score of a dataset. Long Short Term memory models (LSTM) can handle this long term dependencies issue by bringing a memory into the system.

Twitter dataset is a rich source to deep dive for analysis of opinion of different sorts of occasions and products. [19] illustrates that identifying the assessment of these small texts and tweets is a difficult errand that has pulled in expanded research enthusiasm for ongoing years. The paper expresses that the conventional RNNs are not capable enough to manage complex estimation of sentiment analysis, in this way a LSTM system came into picture for classification and the assessment of tweets. A model is trained on the Twitter Sentiment corpus, a dataset containing 800,000 positive marked tweets and 800,000 negative named tweets. The outcomes demonstrate that the LSTM system beats all different classiers including RNN, Naive Bayes and Support Vector Machine.

One more sentiment analysis work has been done on 4 huge datasets is displayed in [20]. Three datasets are of caf reviews from Yelp for years 2013, 2014 and 2015. A rating between 1 to 5 stars is given to each restaurant. IMDB dataset has also taken which has reviews from movies and divided as negative or positive review. In the analysis, a LSTM model and other models were compared and LSTM model yields the best results on each of the four datasets.

The writing study demonstrates that the LSTM model is an exceptionally groundbreaking classier for sentiment analysis since it utilizes a memory in the network. Having a memory in the system is helpful on the grounds that when managing sequenced information, for example, a content, the importance of a word relies upon the setting of the past content.

Table 2.1: Related work at a Glance

Research	**Author**	**ML-techniques used**	**Best results**
Reviews Usefulness Prediction for Yelp Dataset	Zhang, H., Liu, X. and Ying, K., (no date)	Logistic regression, K Nearest Neighbors (KNN), Support Vector Machine (SVM) and Random Forest	Random Forest
Predicting Usefulness of Yelp Reviews	Liu, X., M. Schoemaker. And there's Zhang, N. (2012)	Quadratic Discriminant Analysis (QDA), Linear Discriminant Analysis (LDA), Logistic regression, Support Vector Machine (linear), Logistic regression (Lasso), Support Vector Machine (RBF) models, Nave Bayes Machine (SVM) and Random Forest	Logistic Regression (with Lasso) and SVM algorithms
Prediction of Useful Reviews on Yelp Dataset - Final Report	Li, Y. et al. (no date)	Support for Vector Machine-Models of Linear, Polynomial, Radial Kernel and Random Forest	Random Forest
Extracting Useful Features with Text Mining and Exploring Regression Techniques for Count Data	Unknown Author	Random Forest Regression, LASSO regression, Negative Binomial Regression and zero inflated negative binomial regression	LASSO regression using TF-IDF feature
Predicting Usefulness of Yelp Reviews with Localized Linear Regression Models	Unknown Author	Linear regression with Ridge regression and LASSO regression, bag of words with TF-IDF feature	Ridge regression
Automatically assessing review helpfulness	Soo-Min Kim, Patrick Pantel, Tim Chklovski, Marco Pennacchiotti (2007)	Support Vector Machine	Support Vector Machine

Chapter 3

Research Methodology

The main aim of this research is to predict the usefulness of the reviews of Justdial data and to analyze the factors responsible for determining the usefulness of reviews. For this, a standard and effective data mining method has been concluded as best suited and applied to achieve the desired output. Knowledge Discovery Databases (KDD) is the data mining process which is used for this research. [21]

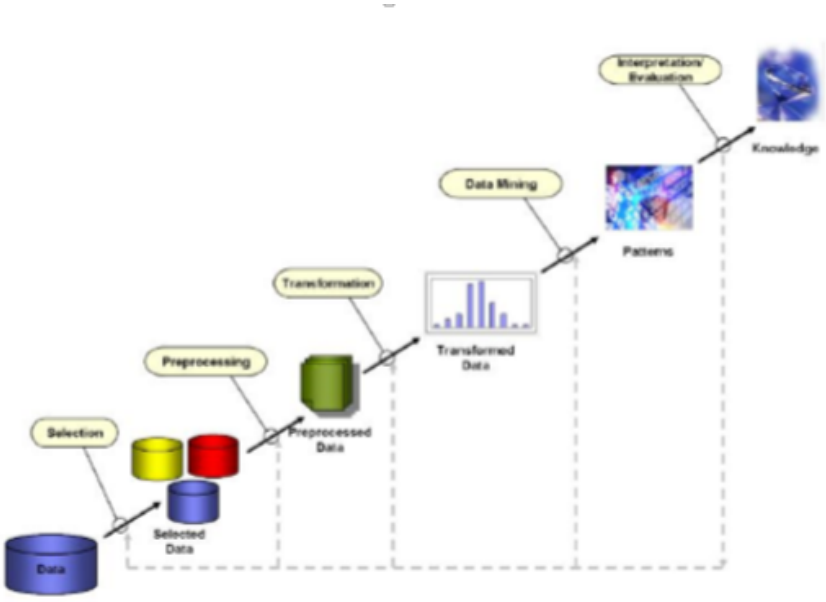


Figure 3.1: KDD process diagram [4]

There are following stages of KDD which comes first and are related to the data pre-processing, i.e., data collection, selection of data, data pre-processing and transformation of data. Next stages consist of implementation of the data mining methods, analyzing and evaluating the outcomes and at end concluding the information and knowledge for application on the real-life scenario problems.

Data collection is the foremost step of this process. This consist of looking out for a dataset which is suitable for the model implementation and analysis and then achieving the outcomes as expected. Basic description of the dataset and its attributes can be introduced at this stage.

Data selection is a very essential in the data mining methods. It is necessary to choose appropriate features and variables for analysis and model training as not correlated or unnecessary data features can impact the model negatively and can under-fit or over-fit the machine learning model. In this research, some selected variables have been considered from the dataset and some have been calculated and added as required. The attribute useful_tag is the target variable which is to be predicted useful or not useful. 0 means review is not useful and 1 means review is useful.

Data pre-processing means to clean the data like removing NA values, special characters, removing noisy data for example outliers etc. Data Imputation is also comes under this stage. Imputation of data means to populate the data where there are missing values in the records. There are many standard methods which can help statistically to impute the values in the dataset. In this research, many pre-processing tasks like removing stop words, adding polarity score, managed user mentions, bigrams, positive negative emotions, removing punctuations, special characters, handling spaces, used tokenizer of Keras dictionary (part of natural language processing) has to be implemented for applying the model finally. Some data visualization techniques have been used like word cloud, correlation plot and box plots for detecting outliers.

Data transformation is the approach to convert the data in an appropriate format so that model can understand and be trained. Different models demands different type of data formats. If model gets the data in the format as required, it can under-

stand the data accurately and give better results. There were 4 new features added that are max_date, days_to_review, useful_per_day and updated useful_tag (on the basis of weighted usefulness per day). All these features are described in the further sections. SMOTE (Synthetic Minority Over-Sampling Technique) is also used to handle the imbalance data as in this dataset, not-useful samples were in majority and useful were very few.

Data mining is the next level of the KDD process where algorithm implementation can be performed to get the expected results. This process also consist of visualizing the data to understand the important pattern which can be interpreted for concluding the results in a better way. Random Forest, GBM, XGBM, LGBM, LSTM models are trained. Train and Test set split is 80% and 20%.

Interpreting and evaluating process discusses about the results and to understand and compare the results of various algorithms in order to conclude the best one. Evaluation can be critical for future improvement scope. Root mean square error, mean absolute error, precision, recall and F1 score are the techniques used to evaluate the results of the model.

Knowledge is the final stage of the KDD process where analyzed results and conclusions of the models can be applied on real-time problems to take decisions in order to gain the insights to resolve the problems.

The dataset for this research has been taken from a public website that is Kaggle.com. This dataset contains the records of reviews which are marked as useful 1 and not useful 0. There are 9860 samples of reviews and 12 features (attributes) present in the dataset.

Chapter 4

Design Specification

This entire research code has been developed in Python programming language and ran on Google Colab engine online. Google Colab is free and open source online platform on which one can run the python code without installing any libraries. There were some version specific libraries used in the code which are not supported by CPU systems for example tenserflow, Keras etc. as they run on high computational GPU hence Google Colab is the efficient platform to deal with. Google Colab provides the built in libraries which are used in data mining, data visualization, statistical analysis etc. which can be simply installed and used. There is no impact of local machine configuration on the speed of Colab online. In this research, GPU was explicitly selected on Colab notebook from runtime dropdown and whole code was run on python 3.

Below figure shows the flow chart of implementation work done in this research. It illustrates that from the first step of problem statement to the last step of conclusion, various methods were implemented on data. Data imputation, transformation, basic analysis with the help of data visualization plots and data pre-processing has been done. After all the pre-processing, a processed data will be ready on which models are trained. Evaluation, analysis and conclusions are discussed in later stages.

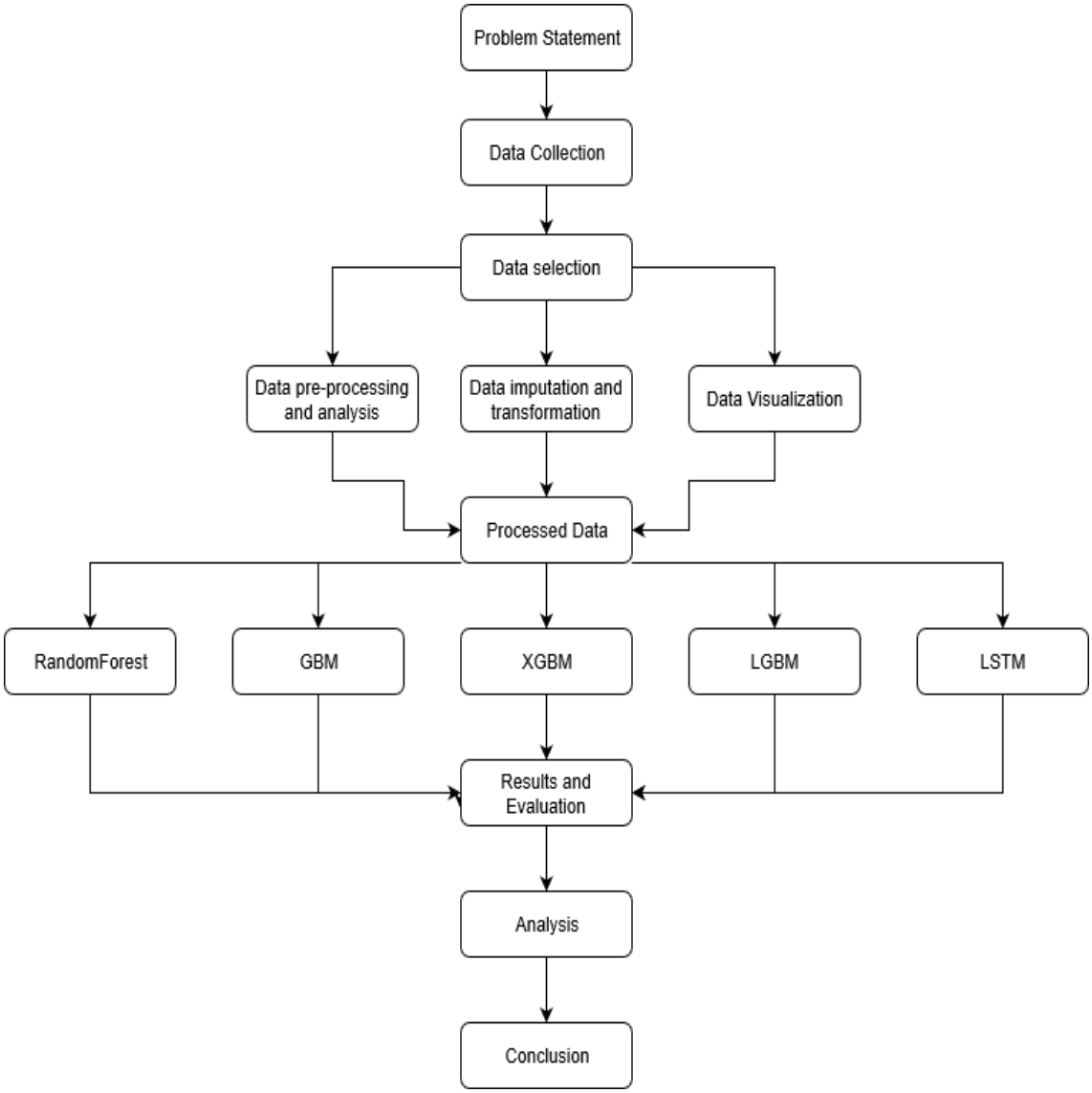


Figure 4.1: Flow chart

Chapter 5

Implementation

The main aim of this research is to predict the usefulness of the reviews in the justdial data set. Data is downloaded from Kaggle which is an open source website for data. Data consist of records from year 2003 to Dec 2016. The attributes of the data and their respective description are mentioned below in the Table 5.1.

Table 5.1: Data Description

Attribute	Data description	Data type
Business_id	Unique ID for each business	Alphanumeric (Unique ID)
Date	Date when review were posted	Date
Review_id	Unique ID for each review	Alphanumeric (Unique ID)
Stars	Rating for business (1 to 5)	Categorical
Text	Review text	String
Type	Review type	Categorical
User_id	Unique ID for each User	Alphanumeric (Unique ID)
Useful	Frequency of likes on each review	Numerical
Max_Date	Maximum Date in Dataset	Date
Days_to_review	Number of days to calculate that since how long the review is posted	Numerical
Useful_per_day	Weighted Usefulness	Numerical
Useful_tag	Useful is 1 else 0	Categorical

5.1 Pre-processing of Data

In the above mentioned dataset, following columns were present already i.e. Business_id, Date, Review_id, stars, text, type, user_id and useful. Rest four have added for this work.

Dataset consist of:

- **Businesses:** Dataset consists of 4132 unique business IDs.
- **Reviews (text):** 9858 unique review texts. Data contains full review (for NLP processing).
- **Reviews:** 9858 unique review IDs hence data have unique texts posted.
- **Users:** 6329 unique users who have actively posted different reviews for different businesses.

After pre-processing of the data and visualizing it at the early stage of this research, this came into light that there are some reviews which are very old and have received more number of likes while there are some reviews which are posted recently and have less number of likes as obvious. In some scenarios, some reviews which are new and so are less popular have received less likes but it doesnt mean that they are less useful. There are chances that these new reviews are more useful and get more number of likes in less number of days. This could mis-train the models to predict the usefulness of any review hence to handle the weightage of review usefulness per day, extra 4 columns were added which are explained below:

Max_date This date is the latest date in the data on which some reviews might have posted and it is assumed that data would have extracted on this date.

Days_to_review It is calculated as the difference of Max_date and review posting date. This difference means the number of days since review has been posted on the site from the latest date.

Useful_per_day - It is calculated as the useful (likes) divided by Days_to_review. This will calculate a value which will be considered as usefulness of a review per day since the day it has been posted. If there are no likes on a review that means the value of useful column will be 0, it is considered as not-useful review. If a review has received even a single like, it is considered as useful and value assigned as 1.

Useful_tag- This column contains the review usefulness after considering the weightage as per the number of days. Value will be binary that is 0 or 1.

Target column is Useful_tag which will be predicted by our Machine Learning models. This is classification problem which also contains the natural language processing and sentiment analysis as well as Deep learning.

Columns have been added in Microsoft Excel sheet. There were some records which have missing data and they are imputed using sklearn.preprocessing import Imputer library of python.

Firstly, since there were 3 unique IDs i.e. Business_id, Review_id and user_id were present in the dataset and ID columns plays no importance in model training hence they were removed manually. Type column was also containing only one type value i.e. review hence it has been also removed the data. Rest all columns were considered further for processing.

Secondly, some preprocessing of data have been performed analyze the impact of every attribute on the target. A correlation graph has been plotted between all the features and target variable (Useful_tag) to analyze the relation among them. Using Seaborn python library, this heat map plot was coded. For visualizing statistical data, this library is recommended. Below figure illustrates correlation between the all the attributes and the Useful_tag.

Correlation in features The right side scale which is between -0.8 to +0.8 shows the correlation on the graph. -0.8 shows least relation and +0.8 shows high relation. Correlation can be analyzed based on the color shade. The dark color shows high

correlation and light color shows low correlation. This graph says that except star rating, rest all columns are important to predict the target variable.

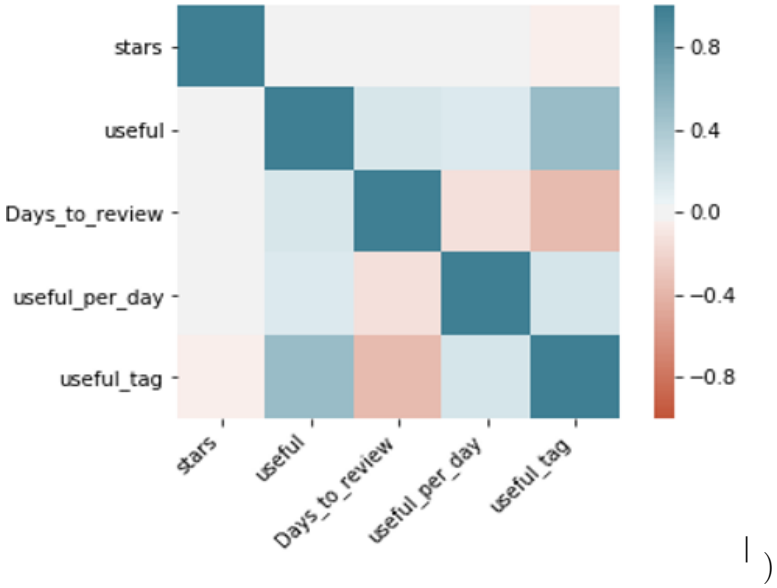


Figure 5.1: Correlation graph

Since text column which contains reviews is being analyzed and prediction is being done, natural language processing methods are also used to transform the data in this piece of work.

For pre-analyses of the text column, a word cloud has been plotted to view the most frequent words of the dataset. There are following steps before making the word cloud and same has been implemented in final data which will be used to train the models:

- Case of the words has been ignored while creating the cloud. For example, a word Amazing can be written as amazing and AMAZING. Cloud will consider both of the ways as same word and plot it as per its occurrence.
- Removed special characters and brackets.

- Expanded the words like dont is changed to do not.
- Applied lemmatization Example: is or are converted to be. Lemmatizations is the process to group together the words which are same in meaning but their verb forms are different so they can be considered and analyzed as single word. For example eat, ate, eaten will be considered as eat.
- Converted all words to lower cases.
- Removed all the stop words, using NLTK library by tokenizing the text, then split the string into a list of substrings and then remove stop words like a, an, the etc.
- Numbers not removed from the text as prices of something can impact the review usefulness [22]

5.2 Assumptions

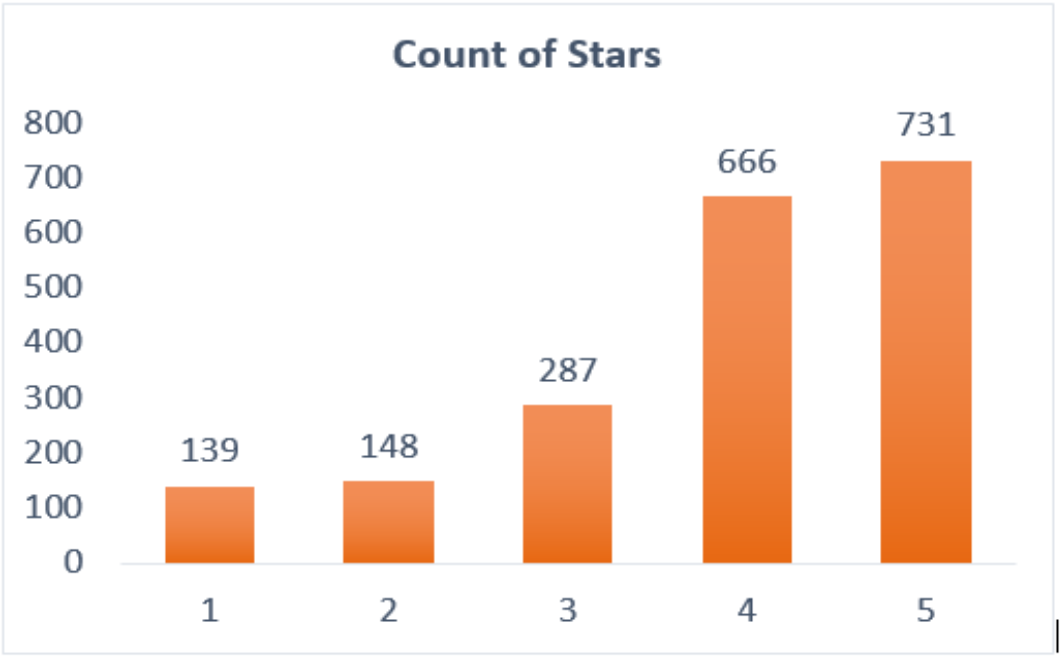
- A long review is assumed as more informative for readers as compared to sort one.
- Also, any review which has more sentences is considered as more informative and useful for the users.
- Count of exclamation signs in a review are considered as excitement and positivity of user. [7]

)

Figure 5.2: Word Cloud

In the above word cloud, it can be seen that some frequent words which are location, like, beef, place etc. are not helpful in providing the information and however NLP will help to get more information out of it. This suggest to give some weighting to the words and not just count and frequencies.

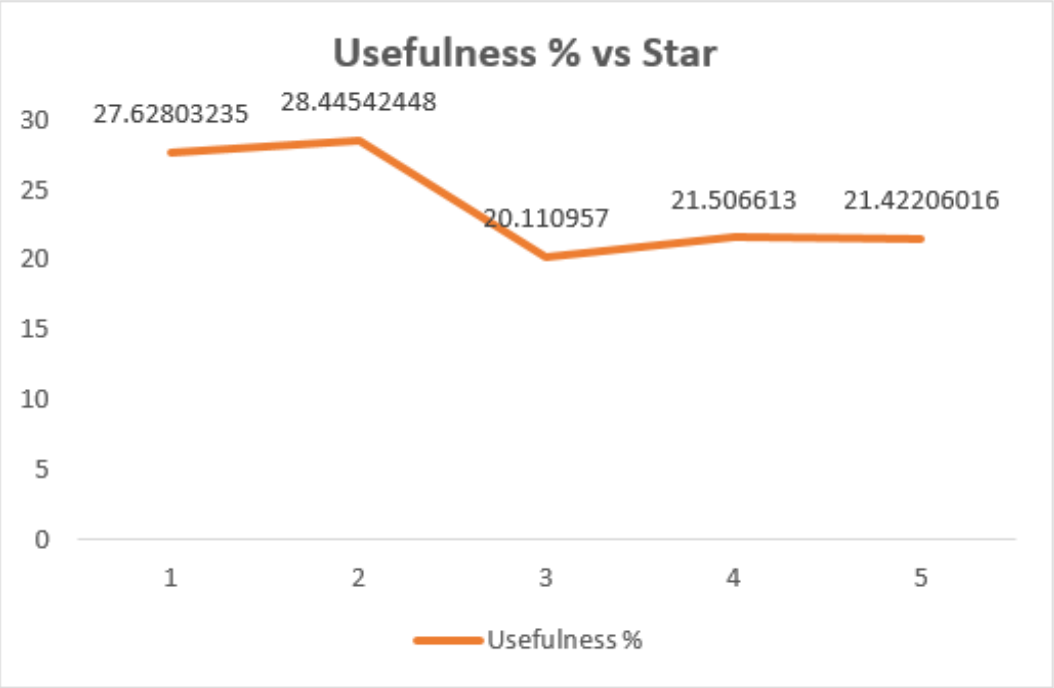
In the graph below, count of star ratings for each category of stars from 1 to 5 has been shown. The records which has star rating as 5 are maximum in number followed by star rating 4 and so on. There are least number of records which have star rating as 1.



)

Figure 5.3: Count of Stars

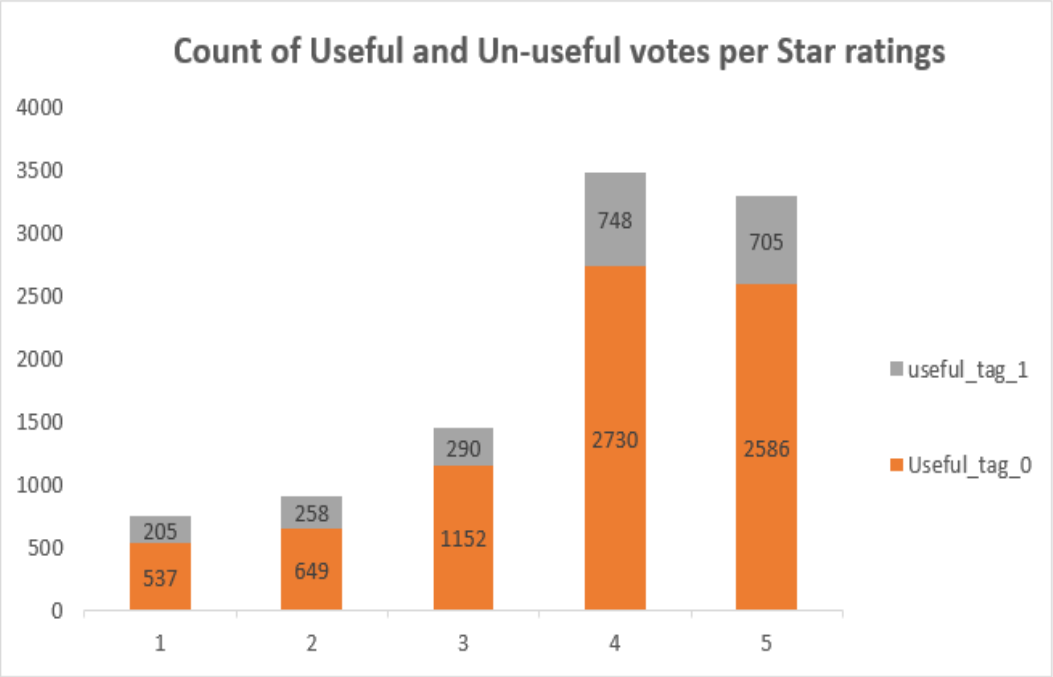
Below graphs illustrates count and percentage of useful and not-useful votes as per star ratings for all the 5 categories of star rating. It is clear from these graphs that data has class imbalance problem which has to be rectified before training the models otherwise model will be biased towards the class which have maximum number of records. It is clear from the graph that approximately 20% to 28% reviews are useful in the data and rest records are not-useful.



)

Figure 5.4: Usefulness vs Star in percentage

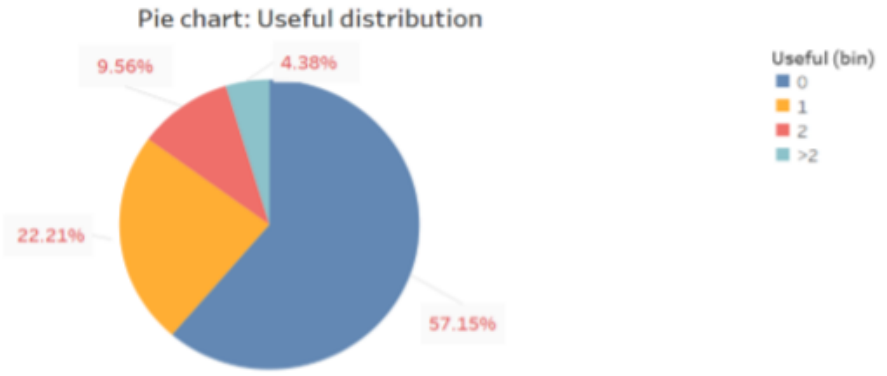
Below graph shows the count of records with useful and not-useful category as per the star ratings. It is discovered from the below graph that all reviews with rating 1 star, useful reviews are just 27% and rest are useless reviews. With increasing rating stars like in reviews with 2 star, useful reviews have increased to approximately 28%. Further, for star rating 3, 4 and 5, useful reviews are sharing just 21% out of total reviews. Therefore it can be concluded that usefulness and star ratings on a review are not (negatively) correlated.



)

Figure 5.5: Count of usefulness and Un-usefulness votes per star ratings

Below Pie chart shows the distribution of useful (frequency of upvotes) reviews. It is clear that data has nearly 60% of reviews who have 0 votes that means they are not-useful. 22.21% reviews received 1 upvote only. Rest, the reviews who have received 2 upvotes or more than 2 upvotes are approximately 14% altogether. Data is highly imbalance and hence this will be handled in this research further.



)

Figure 5.6: Useful distribution

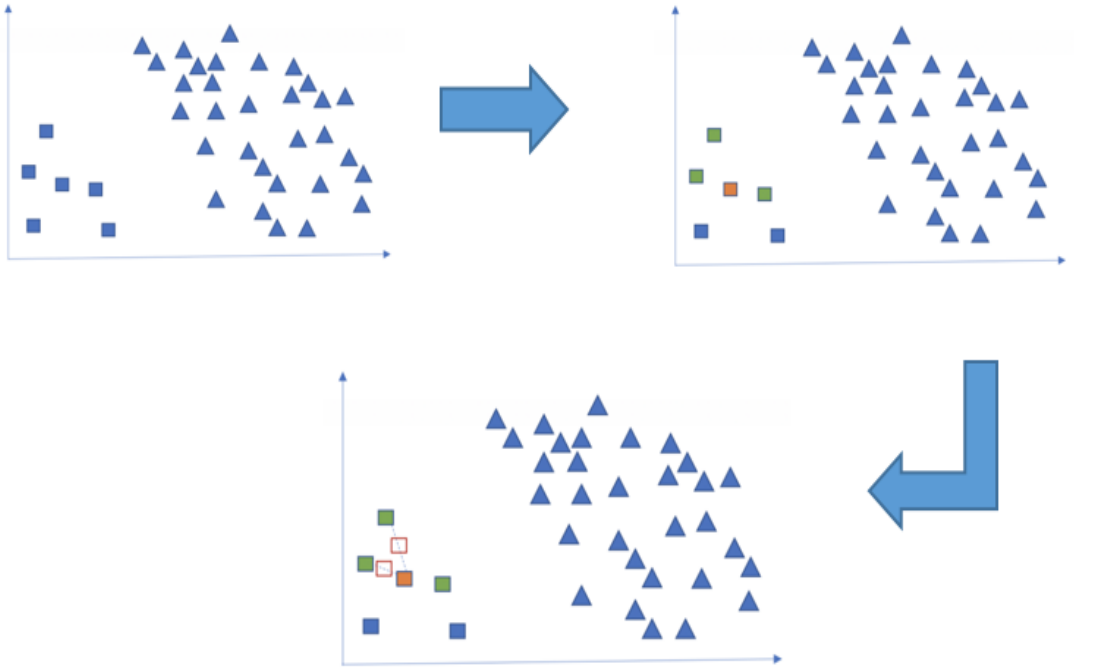
There are some reviews which have received 10 or more than 10 votes and some have received only 1 vote. This unequal distribution has been handled in a way that if review have at least 1 vote, it is useful else 0 that means not useful. This is now a classification problem with 2 classes of 0 and 1.

It can be seen that data is highly imbalanced and models will be misfit. To overcome this problem, Synthetic Minority Over-Sampling Technique (SMOTE) technique has been used.

)

Figure 5.7: SMOTE [5]

Synthetic Minority Over-Sampling Technique: SMOTE (Synthetic Minority Over-sampling Technique) is the technique to synthesize new minority classes. Its the method to create a new class from the data using k-nearest neighbor. It is an advanced technique for class imbalance handling and it is widely used. SMOTE actually oversamples the lesser (minority) class, it does not really depends on reusing existing records. It completely SMOTE creates new (synthetic) records based on the observations in the dataset [6] Below chart is clearly explaining how SMOTE actually works:



)

Figure 5.8: SMOTE Process [6]

In the above figure, it is clear that square class is under sampled. SMOTE process will choose k-nearest neighbor of each sample of the minority class. In this research, K is taken as 3. Let us assume that Orange Square is the sample and green squares are the 3 nearest neighbors of this sample. Final step is to create a line connecting the sample and its neighbors and fill the samples in between with new observations (with transparent squares) as shown in the above figure.

Main advantage of SMOTE over other traditional sampling methods is that by creating new synthetic observations rather than using existing samples, there are less chances of overfitting the model. Simultaneously, it is important to consistently ensure that the new samples made by SMOTE are reasonable and realistic. SMOTE process creates new samples possibly helps if the new samples are practical and can be seen in reality.

Sentiment Polarity Score Sentiment analysis is maybe one of the most well-known utilizations of NLP, with countless instructional exercises, courses, and applications that focus on analyses of the human sentiments of various datasets extending from corporate studies and surveys to movie audits and reviews. The key part of sentiment analysis is to examine a group of content for understanding the feeling communicated by it and for analyzing the expression conveyed. Ordinarily, we measure this feeling with a positive or negative score, called **polarity**. The overall sentiment cab be interpreted as positive, negative or neutral from the sign of the polarity score. Its value exists between -1 to +1. The result of this sentimental analyzer is the probability of negative, positive or neutral sentiment and the nal summation is known as compound. An example of the result of polarity will be like neg: 0.0, neu: 0.255,pos: 0.745, compound: 0.8316, which shows that user is positive with the service and posted a good review. [23]

Mostly, sentiment analysis works best on content that has an emotional (subjective) notion than on content with just an objective context. Objective text more often contains some ordinary facts or statements and they are usually without communicating any feeling, sentiments, or temperament. Subjective content contains language by a human having mind-sets, feelings, and sentiments. Sentiment analysis is broadly utilized, particularly as a piece of social media content analysis, be it a business, an ongoing movie, an event, to comprehend its vibes among the individuals and what they consider it dependent on their conclusions or, you got it, sentiment!

All the code is written in Python programming language.

Following libraries are being used:

- import numpy as np
- import pandas as pd
- import re
- import seaborn as sns

- from seaborn import categorical
- import nltk
- nltk.downloader.download('vader_lexicon')
- from nltk.sentiment.vader import SentimentIntensityAnalyzer as sia
- import sys
- from nltk.stem.porter import PorterStemmer
- from sklearn.metrics import mean_squared_error
- import math
- from math import sqrt
- from sklearn.metrics import mean_absolute_error
- from sklearn.ensemble import GradientBoostingClassifier
- from sklearn.ensemble import GradientBoostingRegressor
- import xgboost as xgb
- from sklearn.preprocessing import Imputer
- import os
- import time
- from tqdm import tqdm
- from sklearn.model_selection import train_test_split
- from sklearn import metrics
- from sklearn.ensemble import RandomForestClassifier
- from sklearn.metrics import accuracy_score
- from sklearn.metrics import confusion_matrix

- from keras.preprocessing.text import Tokenizer
- from keras.preprocessing.sequence import pad_sequences
- from keras.layers import Dense, Input, LSTM, Embedding, Dropout, Activation, CuDNNGRU, Conv1D,GRU
- from keras.layers import Bidirectional, GlobalMaxPool1D
- from keras.models import Model
- from keras import initializers, regularizers, constraints, optimizers, layers

Test and Train: It is important to split the dataset into training and test sets in a specific proportion for training the models and for achieving good accuracy scores. In this work, total number of records are 9860 and training set and test set has been split into 80% and 20% respectively which is equal to 7888 and 1972 in number respectively.

To evaluate our models, accuracy, precision, recall and F1 scores are calculated. Root mean square error, Mean absolute error and mean square error are calculated and compared.

- **Precision :** Precision talks about how precise/accurate a model is out of total predicted positive, how many of them are actual positive.
- **Recall:** Recall actually calculates how many of the Actual Positives a model capture through labeling it as Positive (True Positive).

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$)

- **F1 score:** F1 Score is needed when one want a balance between Precision and Recall and there is an uneven class distribution (large number of Actual Negatives).

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

)

5.3 Model Training

After all the pre-processing of the done is done, now machine learning models will be trained and model evaluation will be done.

Random Forest: While training Random Forest model, some parameters has been used and tried various combinations to get the better accuracy. The parameters taken are: n_estimators= 1000, max_depth= 100, max_features = 5, random_state= 11, class_weight='balanced'. Max_depth and random_state were tried with 50 and 42 combination respectively however there were no considerable impact on result. Since the dataset is imbalanced between the 2 classes, balanced class weight were taken.

Xtreme Gradient Boosting model: This classifier were trained with different combinations as well as to handle the class imbalance. Some important parameters are learning rate, n_estimators and max_depth which were taken as 0.05, 1000 and 5 respectively.

Long short term memory model: For training LSTM model, work tokenizer has been used and activation functions were relu and sigmoid. Batch_size were taken 512 and epochs were taken as 7 because after trying more number of epochs, it was noticeable that accuracy were consistent after 7 epochs. Later result converted into binary and threshold value taken as 0.5.

Gradient Boosting model: This model were trained using python library from sklearn.ensemble import GradientBoostingClassifier. This model is generally used

to boost the gradient and hence accuracy of the model. To train this model, several combinations were tried and some important parameters are learning_rate=0.07, loss='deviance', max_depth=4, min_samples_leaf=1, min_samples_split=2, n_estimators =30, random_state=40. Various random state and max depths as well as min sample leaf parameters were increased and decreased to try for better results.

Light Gradient Boosting model: This model were trained by importing lightgbm library in python. This is last model trained for this piece of research. Learning rate, number of leaves and max depth are the various parameters which were tried with different combinations to get the better accuracy. Learning rate taken as 0.003 while number of leaves and max depth taken as 10 each.

Chapter 6

Results and Evaluation

After the implementation of the all the above mentioned 5 models, their accuracy, precision, recall and F1 score were recorded.

Random Forest: Overall achieved accuracy is 76.93%. Confusion matrix and precision recall scores are as follows:

		Predicted	
		Negative	Positive
Actual	Negative	613	1
	Positive	175	0

	precision	recall	f1-score	support
0	0.78	1.00	0.87	614
1	0.00	0.00	0.00	175
accuracy			0.78	789
macro avg	0.39	0.50	0.44	789
weighted avg	0.61	0.78	0.68	789

)

Figure 6.1: Accuracy of random Forest

Extreme Gradient Boosting Model: Overall achieved accuracy is 76.68%. Confusion matrix and precision recall scores are as follows:

		Predicted	
		Negative	Positive
Actual	Negative	612	2
	Positive	170	5

	precision	recall	f1-score	support
0	0.78	1.00	0.88	614
1	0.71	0.03	0.05	175
accuracy			0.78	789
macro avg	0.75	0.51	0.47	789
weighted avg	0.77	0.78	0.69	789

)

Figure 6.2: Accuracy of XGBM

LSTM: Overall achieved accuracy is 76.93%. Confusion matrix and precision recall scores are as follows:

		Predicted	
		Negative	Positive
Actual	Negative	614	0
	Positive	175	0

	precision	recall	f1-score	support
0	0.78	1.00	0.88	614
1	0.00	0.00	0.00	175
accuracy			0.78	789
macro avg	0.39	0.50	0.44	789
weighted avg	0.61	0.78	0.68	789

)

Figure 6.3: Accuracy of LSTM

GBM: Overall achieved accuracy is 76.93%. Confusion matrix and precision recall scores are as follows:

		Predicted	
		Negative	Positive
Actual	Negative	612	2
	Positive	174	1

	precision	recall	f1-score	support
0	0.78	1.00	0.87	614
1	0.33	0.01	0.01	175
accuracy			0.78	789
macro avg	0.56	0.50	0.44	789
weighted avg	0.68	0.78	0.68	789

)

Figure 6.4: Accuracy of GBM

LGBM: Overall achieved accuracy is 76.93%. Confusion matrix and precision recall scores are as follows:

		Predicted	
		Negative	Positive
Actual	Negative	614	0
	Positive	175	0

	precision	recall	f1-score	support
0	0.78	1.00	0.88	614
1	0.00	0.00	0.00	175
accuracy			0.78	789
macro avg	0.39	0.50	0.44	789
weighted avg	0.61	0.78	0.68	789

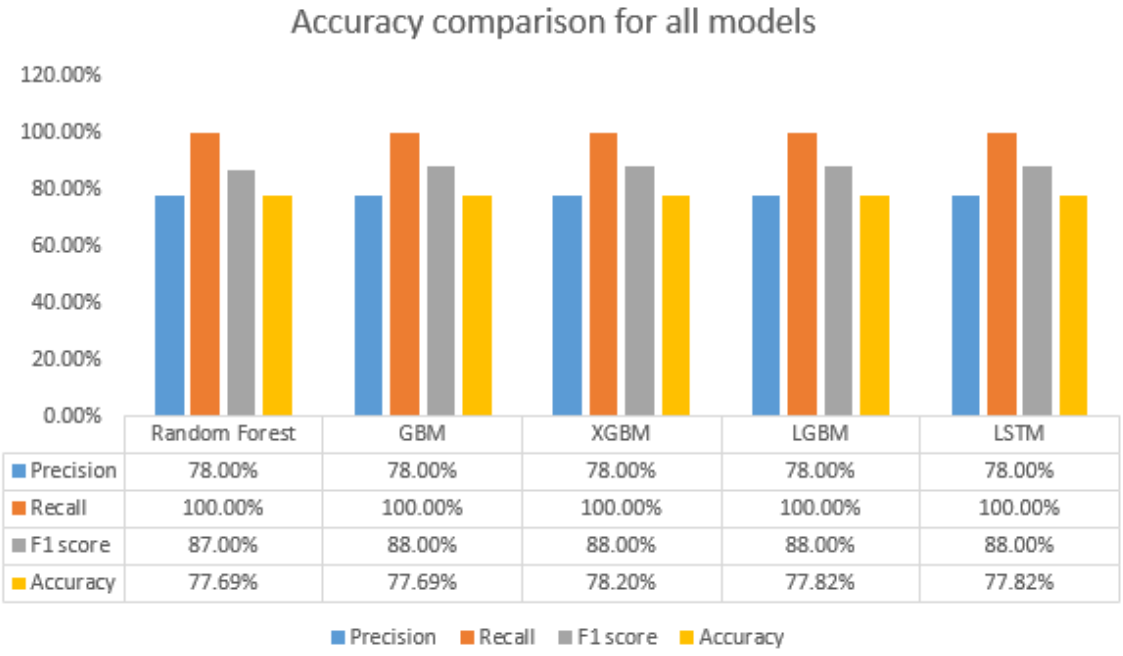
)

Figure 6.5: Accuracy of LGBM

Below are the graphs plotted and accuracy comparison table of all the models. Graph illustrates that all the models are performing good and achieving accuracy of approximately 77%. However, no model is really different from others and accuracy, precision, recall and F1 score values are same.

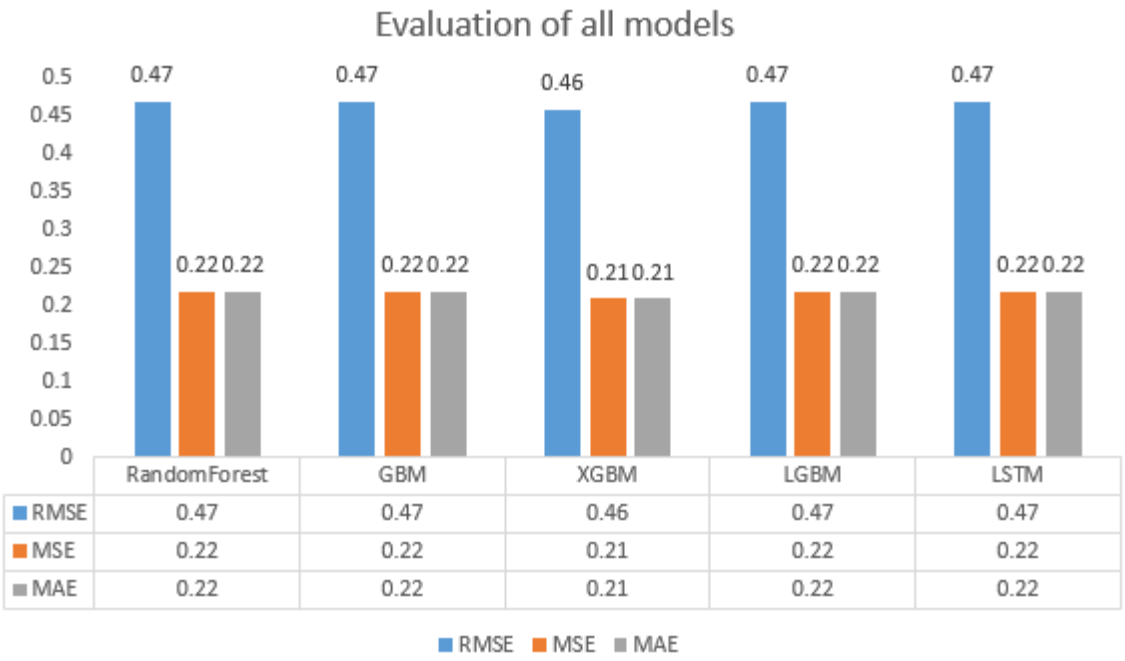
XGBM model performed well and have highest accuracy with 78.20%.

Table 6.1: Accuracy comparison of all models



	Random Forest	GBM	XGBM	LGBM	LSTM
Precision	78.00%	78.00%	78.00%	78.00%	78.00%
Recall	100.00%	100.00%	100.00%	100.00%	100.00%
F1 score	87.00%	88.00%	88.00%	88.00%	88.00%
Accuracy	77.69%	77.69%	78.20%	77.82%	77.82%

Table 6.2: Error comparison of all models



Above table and graph shows the evaluation of all the models. Root mean square error, mean square error and mean absolute error are calculated to understand the model performance. Since all the models performed well and accuracy are almost similar, error also reflects in the same manner.

As XGBM achieved best accuracy hence error is least for this model which is obvious.

Chapter 7

Conclusion and Future work

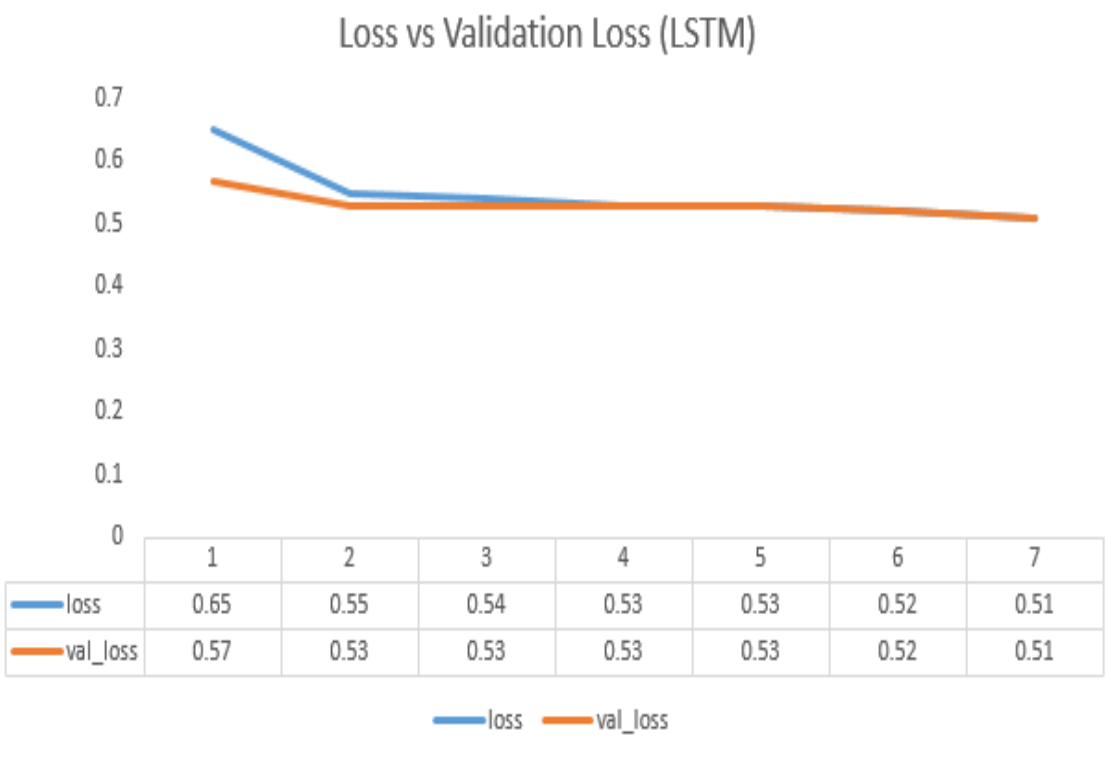
After the model training and analysis, it has been observed that bag-of-words, polarity score, tokenizer and linguistic feature approach did well but could not get better accuracy in any of the model. All the models have almost similar accuracy, precision, recall and F1 score. For evaluation of the models, various evaluation techniques have been used like Root Mean Square Error, Mean Square Error and Mean Absolute Error as well as Confusion Matrix. As the data were imbalanced between the two classes of 0 and 1, balanced class weight has been also tried but somehow it decreased the accuracy by 2%. SMOTE process also could not help much to improve the accuracy of the models.

It can be seen in confusion matrix results that model is predicting true negative values very well however count of false negative is too much which shows the model weakness. It can simply be seen that models are mis-trained due to imbalanced data of 80% and 20% of class 0 and 1 respectively. It is said that if the data has highly imbalanced like in this case, precision, recall and specially F1 score are more important than accuracy. F1 score for all models is consistently similar of around 88% for non-useful reviews which is assumed to be good. If the same models would be trained with large datasets and with more features, it will perform better.

Graphs below shows accuracy and loss for every epoch on the train and test dataset. The graph at the top illustrates the loss, and one at the bottom shows accuracy. The

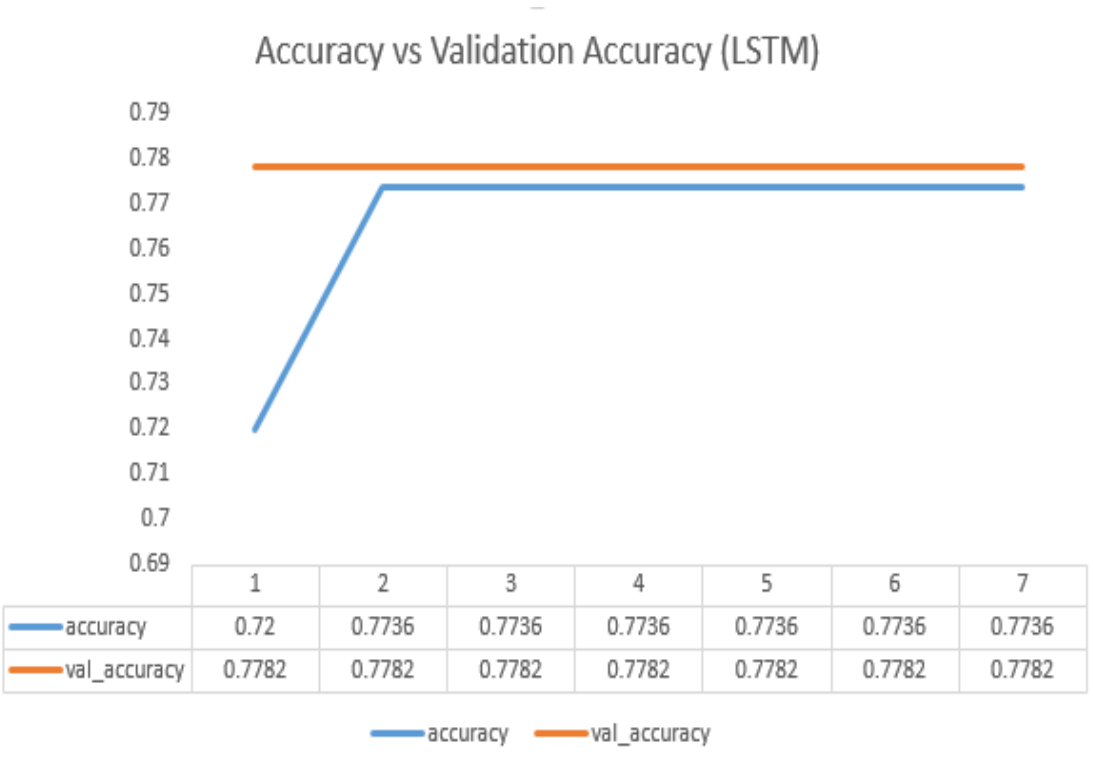
model records the loss and accuracy of train and test dataset in each epoch. Blue lines on both the graphs show the results on the train data, the orange lines on the test set. The loss on the train set getting decreased after each epoch, however the test loss is consistent. The accuracy on the train dataset jumped up in the second epoch, while it is consistent for the test dataset. This might be because the model is overtting. Since model were giving consistent accuracy after 7 epochs, it has been stopped and loss and accuracy recorded as 0.54 and 0.77 respectively. The result of the LSTM model also depends on the activation functions used for getting the output which are relu and sigmoid function in the layers of the model. The predicted label of 0 and 1 are calculated with a threshold value of 0.5.

The parameters of the LSTM model were not tuned in this research. Changing and trying different combinations of the number of neurons might have increased the performance and results.



)

Figure 7.1: Loss vs Validation Loss (LSTM)



)

Figure 7.2: Accuracy vs Validation Accuracy (LSTM)

All the models are working quite well for most of the true negative reviews, however all the models are overfitting and giving more than 150 records as false negatives which means that in actual dataset these reviews were useful but model predicting them as not useful.

K- Nearest neighbors (used in SMOTE) and analysis of the average ratings, were quite helpful in capturing data properties as well as understanding the patterns to predict the user votes and ratings. Furthermore, feature extraction used in this work proved effective at generalizing across models. Expectation from LSTM model were high as it is effective model when it comes about sentiment analysis as said in various literature works however imbalanced data might have proved this fact wrong.

It was challenging to train the model on CPU and tenserflow, Keras libraries of python does not support CPU and hence GPU were used on Google Colab. Linguistic pre-processing of the data was very tedious and challenging task which consumed most

of the resources and time while running the code. All models took around 1 hour all together to train while LSTM alone took 1 hour 30 minutes to give the results. For future work, the performance of all the models can also be evaluated by selecting some different type of words in the reviews, such as removing the top 100 frequently occurring keywords using word cloud analysis. Words such as an, "the", be are frequent words in a text of the review and these are not good for analysis of the sentiments.

There were other features analyzed in the dataset like some businesses and services always gets better reviews as their quality is good and some users always post helpful and useful reviews. This fact can also be taken into account in the future analysis of the sentiments.

It is also been noticed that mostly reviews have the rating between 3 to 5 star and 1 and 2 star ratings were very less. Assuming the fact that people do not take services which have less star ratings and hence there will be less upvotes or down-votes on these type of businesses always. There is another pattern noticed in the dataset that the reviews which has received more than 2 upvotes are more likely to get more upvotes as more users are reading them and finding them helpful. Unlike, the reviews with 0 or 1 upvotes are not catching sufficient eyes of the users and hence these reviews will have lesser votes even if they are being posted for long time. Leveraging the above facts from the data, like average usefulness in terms of a user and average usefulness for a business can help to improve the performance and get better results. Time also plays a very important role while a user is reading the review as user dont consider very old reviews and there are high chances that user will not read and upvote old reviews. It can also be noticed that a user will more likely read reviews which have good star ratings businesses and so the votes on these businesss reviews will be more hence more usefulness count and in the same manner more users hit with more upvotes. Business star rating is very important feature and should be given higher weight when analyzing the reviews in sentiment analysis. Another interesting analysis can be the locality of a business and users crowd availing that business and their review writing pattern, can also play an important role to decide a review to be find useful by other users.

All mentioned analyzed patterns can be used for future work to improve in this area.

The other future work can be to focus on understanding the various kinds of users as per their choices and locality patterns who are providing the reviews and the factors which these users are considering to decide the rating of the business.

Finally, this research can be extended for other types of datasets as well.

Chapter 8

Online Code Repository

The Code of this experiment and results are uploaded on a github repository with the link: https://github.com/adaditi4/Prediction-of-Justdial-reviews-using-Machine-Learning-Techniques

Bibliography

[1] Y. Li, "Prediction of useful reviews on yelp dataset final report," 2019. https://bcourses.berkeley.edu/files/65096735/download?download$_frd = 1$.

[2] N. Z. Xinyue Liu, Michel Schoemaker, "Predicting usefulness of yelp reviews," 2019. http://cs229.stanford.edu/proj2014.

[3] "Unique visitors of justdial in the given duration," 2019. https://images.jdmagicbox.com/investors/Justdial-Company-Presentation-181029083024.pdf.

[4] A. Guerra-Hernndez, R. Mondragn-Becerra, and N. Cruz-Ramrez, "Explorations of the bdi multi-agent support for the knowledge discovery in databases process," *Research in Computing Science*, vol. 39, pp. 221–238, 01 2008.

[5] "Using over-sampling techniques for extremely imbalanced data," 2019. https://towardsdatascience.com/sampling-techniques-for-extremely-imbalanced-data-part-ii-over-sampling-d61b43bc4879.

[6] "A deep dive into imbalanced data: Over-sampling," 2019. https://towardsdatascience.com/a-deep-dive-into-imbalanced-data-over-sampling-f1167ed74b5.

[7] Ruhui Shen, Jialiang Shen, Yuhong Li, and Haohan Wang, "Predicting usefulness of yelp reviews with localized linear regression models," in *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 189–192, Aug 2016.

[8] S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, "Automatically assessing review helpfulness," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, (Stroudsburg, PA, USA), pp. 423–430, Association for Computational Linguistics, 2006.

[9] "Exploring the yelp dataset and extracting useful features with text mining and exploring regression techniques for count data," 2019. https://pdfs.semanticscholar.org/9fb2/489930b3f1204ce0bd7cf1287854b3999798.pdf.

[10] Yujun Yang, Jianping Li, and Yimei Yang, "The research of the fast svm classifier method," in *2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 121–124, Dec 2015.

[11] P. D. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews," 2002. https://www.aclweb.org/anthology/P02-1053.

[12] B. Pang, "Thumbs up sentiment classification using machine learning techniques," 2002. https://www.aclweb.org/anthology/W02-1011.

[13] F. Miedema, "Sentiment analysis with long short-term memory networks," 2018. https://beta.vu.nl/nl/Images/werkstuk-miedema$_t$*cm*235 − 895557.*pdf*.

[14] Datascience.com, "Introduction to random forests," 2019. https://www.datascience.com/resources/notebooks/random-forest-intro.

[15] "Understanding gradient boosting machines," 2019. https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab.

[16] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[17] "What is lightgbm, how to implement it? how to fine tune the parameters?," 2019. https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc.

[18] "Elearning long-term dependencies in narx recurrent neural networks. - pubmed - ncbi," 2019. https://www.ncbi.nlm.nih.gov/pubmed/18263528.

[19] X. Wang, Y. Liu, C. Sun, B. Wang, and X. Wang, "Predicting polarities of tweets by composing word embeddings with long short-term memory," in *ACL*, 2015.

[20] T. L. Duyu Tang, Bing Qin, "Document modeling with gated recurrent neural network for sentiment classification," 2019. https://www.aclweb.org/anthology/D15-1167.

[21] U. Shafique and H. Qaiser, "A comparative study of data mining process models (kdd, crisp-dm and semma)," *International Journal of Innovation and Scientific Research*, vol. 12, pp. 2351–8014, 11 2014.

[22] "Creating word clouds with python," 2019. https://towardsdatascience.com/creating-word-clouds-with-python-f2077c8de5cc.

[23] L. T. et all, "Polarity and intensity: the two aspects of sentiment analysis," 2016. https://www.aclweb.org/anthology/W18-3306.