RATING PREDICTION USING REVIEWS

A Project Report submitted in partial fulfilment of the

requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Janvi Pangoriya(181500292)

Nidhi Gupta(181500422)

Under the Guidance

of

Mr. Saurabh Singhal

Department of Computer Engineering and Applications

Institute of Engineering and Technology

` GLA University

Mathura-281406, INDIA

May, 2021

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P (India)

CERTIFICATE

We hereby declare that the work which is being presented in the B.Tech. Project **"Rating Prediction Using Reviews"**, in partial fulfilment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of our own work carried under the supervision of **Mr Saurabh Singhal, Assistant Professor, Dept. of CEA, GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign: *JanviPangoriya*	**Sign**: *Nidhi Gupta*
Name of Candidate: JanviPangoriya	**Name of Candidate**: Nidhi Gupta
University Roll No.: 181500292	**University Roll No.**:181500422

This is to certify that the above statements made by the candidate are correct to the best of my knowledge and belief.

____

Supervisor

Mr. Saurabh Singhal

Assistant Professor

Dept. of CEA, GLA University

ACKNOWLEDGEMENT

Presenting the ascribed project paper report in this very simple and official form, we would like to place my deep gratitude to GLA University for providing us the instructor Mr Saurabh Singhal, our mentor and supervisor.

He has been helping us since Day 1 in this project. He provided us with the roadmap, the basic guidelines explaining on how to work on the project. He has been conducting regular meeting to check the progress of the project and providing us with the resources related to the project. Without his help, we wouldn’t have been able to complete this project.

And at last but not the least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during this project.

Thanking You.

Sign: JanviPangoriya

Name of Candidate: Janvi Pangoriya

University Roll No.:181500292

Sign: NidhiGupta

Name of Candidate: Nidhi Gupta

University Roll No.:181500422

ABSTRACT

Hotels play a crucial role in travelling and with the increased access to information new pathways of selecting the best ones emerged. For hotels to sustain profitable revenue stream there are many factors they must maintain. One of those is their online presence and reputation – and the clearest indicator of a hotel’s reputation is often found in online reviews. Review websites such as Trip Advisor and Yelp can have a significant impact on how travellers choose their accommodation. Most of the time, the general consensus is what people will accept so it’s vital that your hotel has a reputation for quality service and professional standards. Guest reviews are becoming a prominent factor affecting people’s bookings/purchases. When we look for a place to stay for a vacation on Expedia/Booking/Trip Advisor, we scroll the screen to check on the reviews In other words, guest reviews clearly influence people’s booking decision, which means, the social media sites that are related to travelling and travelling blogs better pay attention to what people are saying about various hotels. Reviews can tell you if hotels and restraunts are keeping up with customers’ expectations and help you learn the most about customers, which is crucial for developing marketing strategies based on the personas of your customers.

Focusing on the same, the aim of this project is to develop a Machine Learning Model which predicts the Rating out of the Reviews that are being collected. In this model, the pattern is observed between the dataset that we have collected and we have tried to study the reviews written and grabbed the keywords from them using a process called Feature Extraction and then we have applied machine learning algorithm over the same dataset and have observed that we make the machine learn on that keywords and then categorize the review on the basis of rating between 1 to 5 where 1 stands as the most depreciated value for the review and 5 stands as one of the best compliment in terms of reviews. Moreover the model we have developed is deployed on cloud as well and hence we have created a web based application where we have got a live link to test a review and make a prediction of the rating. After the model is deployed on cloud all the computations will be taking henceforth on the cloud making it faster and resource free. And this can be used as a real world application by various Travel blogs and sites to increase their effectiveness and utility. Online hotel reviews have become far important than hoteliers might assume. It's high time to wake up to the fact that online reviews are affecting hotel's overall reputation and the industry as well.

CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

81% of travellers frequently or always read reviews before booking a hotel. 91% of 18-34 year-olds trust online reviews as much as personal recommendations. 79% will read between six and 12 reviews before making a purchase decision. 88% of travellers filter out hotels with an average star rating below three. When deciding between two similar properties, 79% of consumers are more likely to choose the object with a higher rating.

To the era we are living in, our choices are made virtually, in an online mode rather than actually visiting and making choices so the only way left with us to verify our choices is the reviews or experience we get to hear about from other individuals we get from the other people who have been there in our shoes. Coming to the impact of online reviews on Hospitality Industry as per a report, Online hotel reviews have become far important than hoteliers might assume. If you have not yet paid attention to managing and responding to online reviews, it's high time to wake up to the fact that online reviews are affecting hotel's overall reputation.

With dozens of review sites like Trip Advisor and Online Travel Agencies (OTAs) such as Expedia, vacationers are placing their trust in hotel reviews while booking accommodation. They are relying on them rather than advertising and marketing efforts from your side. The fact that a lot many of consumers read online reviews from other travellers bring attention to the surge of OTAs and review sites.

1.2 OBJECTIVE

The aim of the project is to create a machine learning model which helps us to analyse the reviews of various customers and to predict the rating using the text classification techniques and applying various algorithms for the same. At the end we would be comparing the results of the algorithms and selecting the one with the maximum accuracy and will be deploying the model. The dataset which we will be using will contain reviews from various travelling websites and hence the implementation of this model will help this website to understand and serve their customer well and this will open the gateways to other text classification model and real time applications.

1.3 CHALLENGES

In the last decades, travelling has changed dramatically due to the evolution and popularisation of information and communication technologies (ICT) as well as mobile devices, namely, smart phones. These devices incorporate on board sensors together with significant computing and communication capabilities, enabling users to generate and share large volumes of data known as crowd-sourced data. Crowd sourcing is an outsourcing process supported by ICT and performed voluntarily by a large number of participants.

Using machine learning algorithms and sentiment analysis of the text, we can try to predict rating of the service. This rating can be useful for both service providers, because it allows getting feedback from reviews, without spending lots of time reading them, and for the users, because it allows them to filter out bad services.

1.4 CONTRIBUTION

The classification and recommendation of goods or services, taking into account the user preferences, is an important task of any on-line Business-to-Consumer (B2C) Web platform. The crowd-sourced feedback, which is volunteered by costumers typically in the form of ratings or reviews, is used by potential costumers to choose new goods or services and by businesses to suggest relevant products. In the tourism domain, the crowd-sourced information is growing dramatically. This Big Data scenario has been addressed by several researchers: (i) Fuchs et al. [8] perform multi-criteria rating analysis from Trip Advisor using Linear Regression; (ii) Fang et al. [7], Han et al. and Wang et al. [22,23] apply regression models to textual reviews; and (iii) Jannach et al. [12] and Chen et al. [3] propose different recommendation approaches based on the user ratings.

1.5 ORGANIZATION OF THE PROJECT REPORT

This paper is organised as follows. Section 1 reviews related work on analysis and prediction of crowd-sourced data. Section 2 introduces Pre-processing, describing current techniques and trends. Section 3 describes the algorithms used, the experiments performed and the results obtained. Section 4 provides the deployment of the model. Finally Section 5 provides the conclusions and discusses the outcomes of this work.

CHAPTER 2
SOFTWARE REQUIREMENT ANALYSIS

2.1 OVERALL DESCRIPTION

2.1.1 PRODUCT PERSPECTIVE

This Machine Learning Application is a self-contained product. On the interior it contains a machine learning model that predicts out the Rating out of the Reviews and on the part of User Interface it is displayed as a Web Based Application where you could enter the Reviews and predict out the Rating.

2.1.2 PRODUCT FUNCTIONS

The major function of this product is it provides the user with a web based app which takes the input from the user in the form of the Reviews and the output is the Rating predicted by the machine learning model.

This has got a really wide scope for the Hotel Industry and Tourism industry as everyone who wishes to visit the place first check up on the reviews given by the other people on a particular destination.

2.1.3 USER CLASSES AND CHARACTERISTICS

Although the product holds a single functionality but there are the various users that will use this real life application for their benefit and will be effective in count of every penny the spend using it. Some of the distinguished users that count in are:

- Bloggers: Travel Bloggers or various types of tech writers whose whole of the study is based on the reviews that are given up by the people can make use of this model to choose the place they focus to highlight on.
- General Public: For anyone who has not visited a particular restraunts or hotel, will definitely look up for reviews on the various sites and hence a good or bad rating might be helpful to make a choice.
- Stake Holders of Hotel Industry: Checking the rating predicted by this model, the stake holders can work upon the feedbacks and requirements of the customer.

2.1.4 OPERATING ENVIRONMENT

Once the model is being developed and deployed on any of the cloud platform like Heroku or

AWS then all of the computations will take place on the cloud only and the user will be provided with Live Link effectively running on any browser like Chrome.

2.1.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

Hardware Requirement:-

- Processor: intel core i5
- RAM: 4GB (minimum)

Software Requirement

- Software used :Anaconda
- Operating System: Any Operating System
- Deployed :Flask

2.1.6 USER DOCUMENTATION

Along with product there are three documents we are providing

1. Synopsis of the project
2. Software Requirement Specification
3. Final Report of the Project
4. Github repository for the Implementation Details

2.2 EXTERNAL INTERFACE REQUIREMENTS

2.2.1 USER INTERFACES

For the User Interface we will be presenting a web based application and that will be containing two WebPages .The first page will be marked for the Input that will be taken from the user. There will be text box where the user will be supposed to enter the review and below it will be action button that will take the user to the result page where the output will be there as predicted by the Machine learning model running in background that will predict the rating from the review entered 1 will be the most depreciated rating value and 5 will be considered as the highest value that can be crawled from the review. For designing the Webpages we have used the various front-end languages like HTML and CSS hence collecting input from the web pages (users) and feeding to the machine learning algorithm and again collecting output from them (the predicted value) and displaying it to the user and hence providing an efficient User Interface.

2.2.2 HARDWARE INTERFACES

Whole of the project is made on softwares,tools, libraries and in such a case no particular hardware is required in particular except for the system we are developing on which requires an intel core i5 processor and a minimum of 4GB RAM.

2.2.3 SOFTWARE INTERFACES

For developing this product, the machine learning model is developed using the Python language on software called Anaconda in particular the Jupyter Notebook which holds the implementation of machine learning algorithms and after that the model is deployed.

2.2.4 COMMUNICATIONS INTERFACES

When the user will be interacting with the machine learning model by means of Web Based Application we will be providing the user with the Live link and while predicting the rating the computations will be taking place on the cloud so we need the connection between the web based application and the model deployed on the cloud that can be provided by Internet. So Internet stands as one of the communication interface required.

2.3 SYSTEM FEATURES

To the era we are living in, our choices are made virtually, in an online mode rather than actually visiting and making choices so the only way left with us to verify our choices is the review we get from the other people who have been in the same place as we are in .So having a numerical scale on the review might be beneficial for both on the part of customer to know about the choice he/she is making and the owner to serve the customer better. This is a text classification project which classifies the text review of the customer on a numerical scale and has scope of various other text classification that can be added to add to the facilities of the customer.

2.3.1 DESCRIPTION AND PRIORITY

A machine learning model which helps us to analyze the reviews of various customers and to predict the rating using the text classification techniques and applying various algorithms for the same. At the end we would be comparing the results of the algorithms and selecting the one with the maximum accuracy and will be deploying the model on cloud using either the AWS or the Heroku so as to ensure all the computations will be taking place on the cloud and hence this model can be practically implemented as a real time application in the field of text classification and an application of cloud computing as well. The dataset which we will be using will contain reviews from various travelling websites and hence the implementation of this model will help these website to understand and serve their customer well There will be text box where the user will be supposed to enter the review and below it will be action button that will take the user to the result page where the output will be there as predicted by the Machine learning model running in background that will predict the rating from the review entered 1 will be the most depreciated rating value and 5 will be considered as the highest value that can be crawled from the review.

2.3.2 STIMULUS/RESPONSE SEQUENCES

The response of the application will be displayed on the web based application that the user is interacting with .It will be displaying the rating for which 5 stands as the highest compliment in rating decreasing the value of the comment to 1 which stands for the most underrated comment.

2.3.3 FUNCTIONAL REQUIREMENTS

The flowchart below explains the various Functional requirements for the model. At each of the step the functional requirements are fulfilled by creating functions

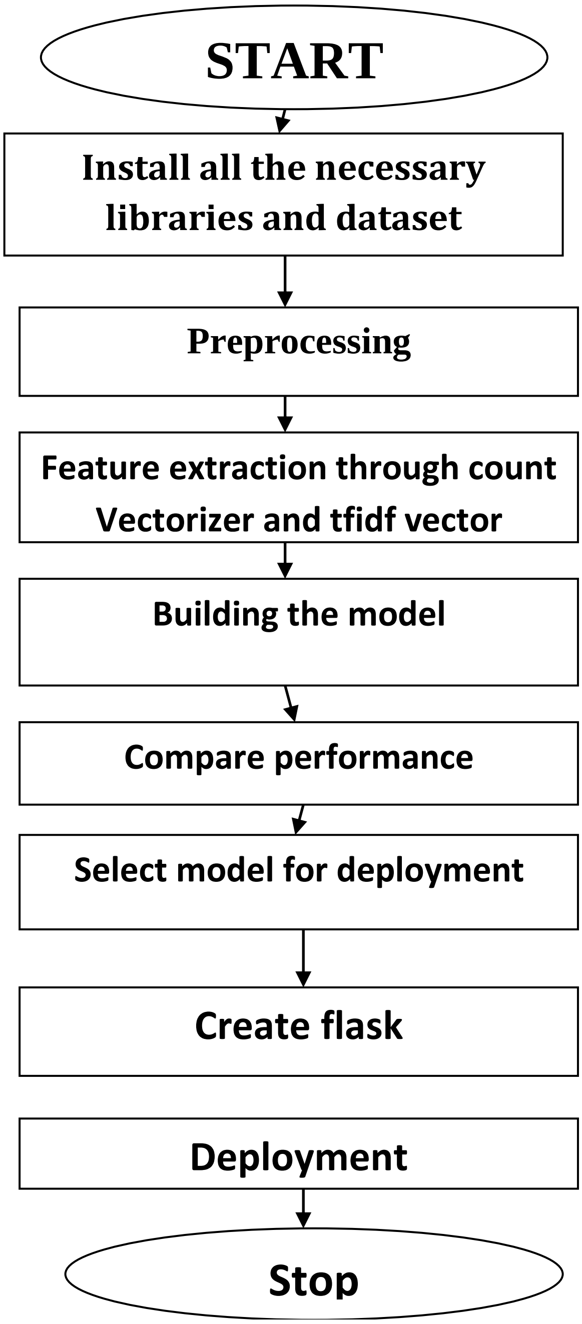

START
Install all the necessary libraries and dataset
Preprocessing
Feature extraction through count Vectorizer and tfidf vector
Building the model
Compare performance
Select model for deployment
Create flask
Deployment
Stop

Figure 2.1: Flowchart

CHAPTER 3
SOFTWARE DESIGN

3.1 USE CASE DIAGRAM

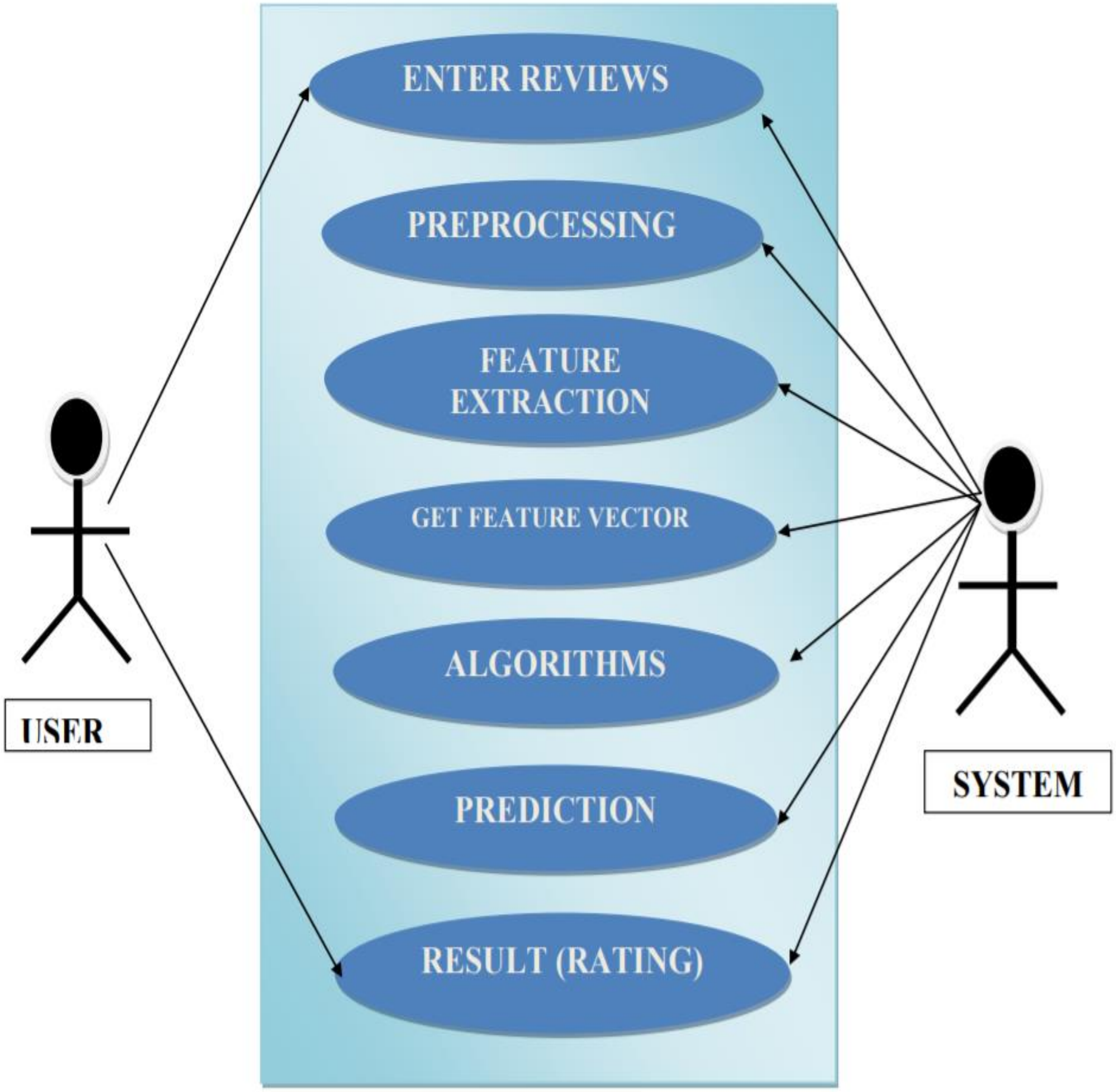


Figure 3.1: Use Case Diagram

Use Case Diagram is a visual representation of how one interacts with the model or software and which modules of the software can be accessed. The above use case diagram shows the visual representation of the “RATING PREDICTION” model. As we observed there will be two views, from which the person can interact one will be the view of the machine and other will be the corresponding rating.

3.2 DATA FLOW DIAGRAM

3.2.1 LEVEL 0 DFD

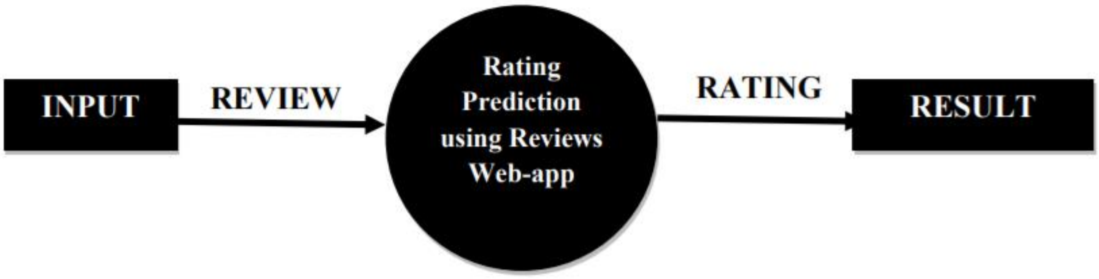


Figure 3.2: Level -0 DFD Diagram

3.2.2 LEVEL 1 DFD

Figure 3 : Level -0 DFD

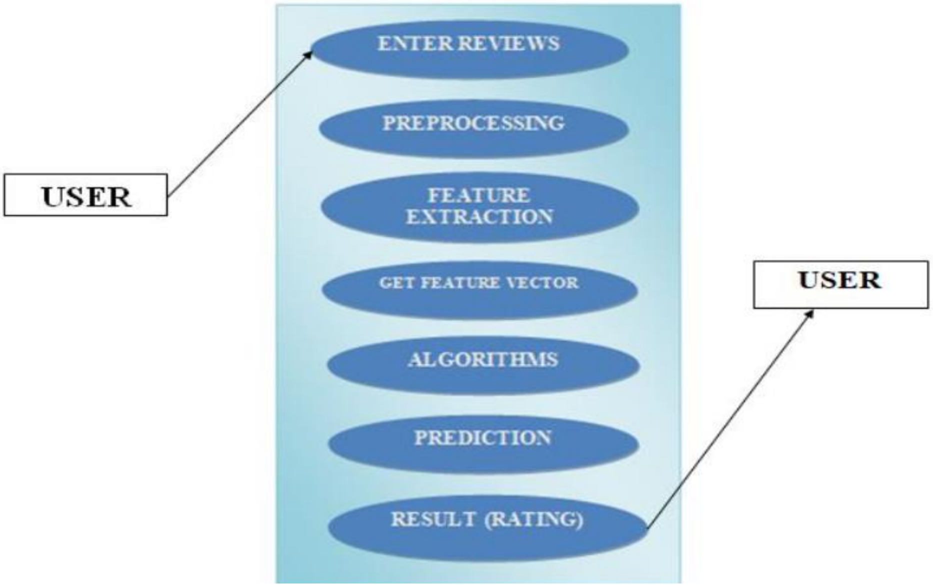


Figure 3.3 : Level -1 DFD Diagram

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. As the user gives the review the data is sent for the pre-processing then the model is Rating Prediction using the algorithm and then the features is taken as input for predicting the rating corresponding review.

CHAPTER 4

METHODOLOGY

The aim of the project is to create a machine learning model which helps us to analyse the reviews of various customers and to predict the rating using the text classification techniques and applying various algorithms for the same. At the end we would be comparing the results of the algorithms and selecting the one with the maximum accuracy and will be deploying the model. The dataset which we will be using will contain reviews from various travelling websites and hence the implementation of this model will help this website to understand and serve their customer well and this will open the gateways to other text classification model and real time applications.

4.1 MACHINE LEARNING WORKFLOW

There are five core tasks in the common ML workflow:

1. Get Data: The first step in the Machine Learning process is getting data. This process depends on your project and data type. For example, are you planning to collect real-time data from an IoT system or static data from an existing database? You can also use data from internet repositories sites such as Kaggle and others.

2. Clean, Prepare & Manipulate Data: Real-world data often has unorganized, missing, or noisy elements. Therefore, for Machine Learning success, after we chose our data, we need to clean, prepare, and manipulate the data. This process is a critical step, and people typically spend up to 80% of their time in this stage. Having a clean data set helps with your model's accuracy down the road. After getting the data to a state you like, you need to convert the data sets into valid formats for your chosen ML platform. For example, you may need to translate the data into a .CSV file. Finally, you split your data into training and test data sets. The training set is used to train the model in the next step, while the test data is used to validate the model in the fourth step. The typical default is a 70/30 split between training and test sets.

3. Train Model: This step is where the magic happens! The data set connects to an algorithm, and the algorithm leverages sophisticated mathematical modelling to learn and develop predictions. These algorithms commonly fall into one of three categories: Binary – Classify into two categories Classification – Classify into many categories Regression – Predict a numeric

4. Test Model: Now, it’s time to validate your trained model. Using the test data from Step 3, we check the model’s accuracy. If the results are not satisfactory, you need to improve and retrain your ML model

5. Improve: Practice makes perfect! Here are a few things you can do to refine your model and improve accuracy: Review your model’s results with your business stakeholders. Are there other data elements worth adding to your model to make it more accurate?

Reconsider your algorithm choice. Within each class of algorithm, there are dozens of algorithm choices. A different algorithm may perform better for you

Adjust the parameters of your chosen algorithm to improve performance. Sometimes small adjustments have a significant impact.

4.2 DATASET EXPLAINATION

The objective of this task is to predict rating out of the reviews. For the sake of simplicity, we say a review can have an impact on the Hospitality Industry So according to the review and the word it contains every review is classified on the scale of 1 to 5.

The dataset used in our project is taken from well know dataset factory i.e. kaggle.com Formally, given a training sample of reviews and ratings, where rating '1' denotes the most depreciated value of the review and rating '5' denotes the most appreciated value for a review, our objective is to predict the rating on the test dataset.

It has a full training dataset with the following attributes:

• Review: the text experience of the user

• Rating: the user experience marked on numerical scale..

For implementation purpose I have choose the training set file , both column from the file for classification that is review and rating.
Originally there were 5 types of ratings:

1. Highly positive is considered as 5 star
2. Positive is considered as 4 star
3. Neutral is considered as 3 star
4. Negative is considered as 2 star
5. Highly negative is considered as 1 star

4.3 PREPROCESSING

After actually getting a hold of your text data, the first step in cleaning up text data is to have a strong idea about what you're trying to achieve.

1. **Load Data:** The text is small and will load quickly and easily fit into memory. This will not always be the case and you may need to write code to memory map the file. Tools like NLTK will make working with large files much easier.

2. **Split by Whitespace**: Clean text often means a list of words or tokens that we can work with in our machine learning models. This means converting the raw text into a list of words and saving it again. We can also use select words to select the words (use the regex model (re) and split the document into words by selecting for strings of alphanumeric characters (a-z, A-Z, 0-9 and '_').

3. **Remove Punctuation**: We may want the words, but without the punctuation like commas and quotes. We also want to keep contractions together. One way would be to split the document into words by white space (as in "2. Split by Whitespace"), then use string translation to replace all punctuation with nothing (e.g. remove it).

4. Stemming**:** Stemming refers to the process of reducing each word to its root or base. For example "fishing," "fished," "fisher" all reduce to the stem "fish." Some applications, like document classification, may benefit from stemming in order to both reduce the vocabulary and to focus on the sense or sentiment of a document rather than deeper meaning. There are many stemming algorithms, although a popular and longstanding method is the Porter Stemming algorithm. This method is available in NLTK via the Porter Stemmer class.

4.4 DATA VISUALIZATION

"A picture is worth a thousand words". We are all familiar with this expression. It especially applies when trying to explain the insight obtained from the analysis of increasingly large datasets. Data visualization plays an essential role in the representation of both small and large-scale data. Data visualization is the discipline of trying to understand data by placing it in a visual context so that patterns, trends and correlations that might not otherwise be detected can be exposed.

Python offers multiple great graphing libraries that come packed with lots of different features. No matter if you want to create interactive, live or highly customized plots python has an excellent library

In our project we have done the data visualization using word cloud. The First is a bar plot for Positive Words and the word cloud is created in the form of squares and the words arranged horizontally and vertically and the size of each word represents its importance.

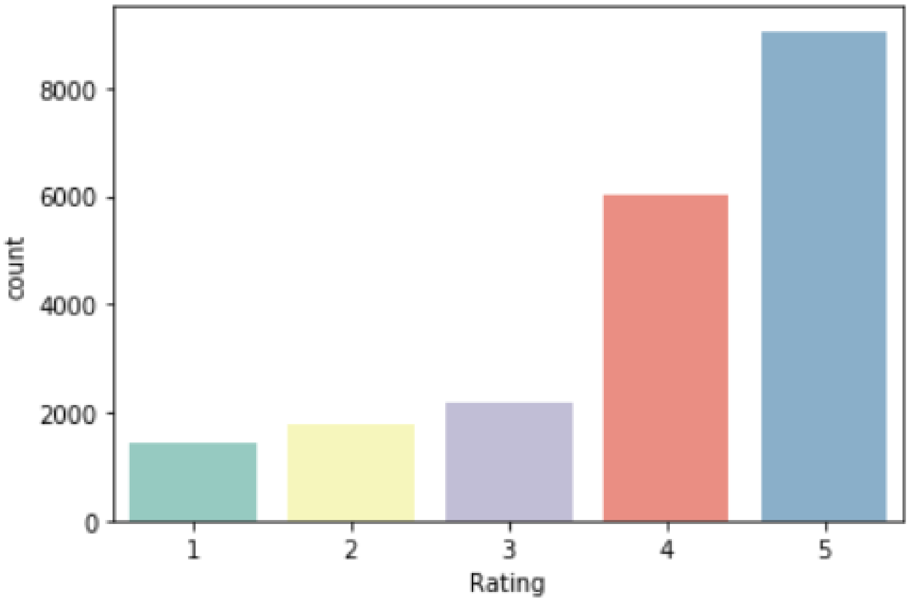


Figure 4.1: Bar Plot of the rating field

Word Cloud of words that occur in the reviews:

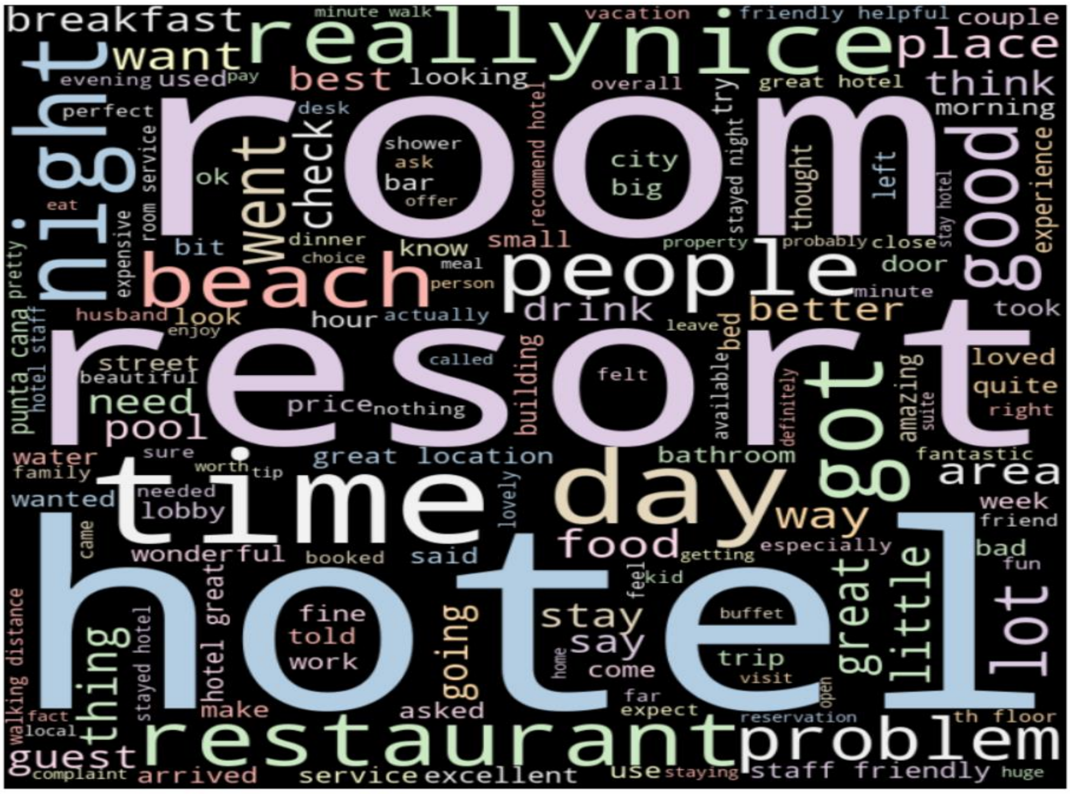


Figure 4.2: Word cloud of review

4.5 FEATURE EXTRACTION

We cannot work with text directly when using machine learning algorithms. Instead, we need to convert the text to numbers. We may want to perform classification of documents, so each document is an “input” and a class label is the “output” for our predictive algorithm. Algorithms take vectors of numbers as input; therefore we need to convert documents to fixed-length vectors of numbers.

To create the vectors of words we use an approach **Bag of Words Algorithms**

- **BAG OF WORDS ALGORITHM**

Bag of words is a way of representing text data when modelling with machine learning algorithm. To convert the text to numbers, in precise vector of numbers, so for extraction of text we use this algorithm. Bag of words describes occurrence of words within the document. And hence creates vocabulary of known words. Here model is concerned with whether the words are known to the documents and their position does not matter. This can be done by assigning each word a unique number. Then any document we see can be encoded as a fixed-length vector with the length of the vocabulary of known words. The value in each position in the vector could be filled with a count or frequency of each word in the encoded document. This is the bag of words model, where we are only concerned with encoding schemes that represent what words are present or the degree to which they are present in encoded documents without any information about order.

- **TFIDF VECTORIZER**

The approach is to rescale the frequency of words that appear often in all documents. Like the short words “the” can be penalised if not removed. The approach is called term frequency or Inverse Document Frequency.
This can be better understood as :-
Term frequency is defined as scoring of the words normally in current documents
Inverse Document Frequency is the scoring of how rare the word is along the document.
Scores have the effect of highlighting words that are different (useful information container) in document. i.e. inverse document frequency of the rare term is higher and inverse document frequency of frequent term is lower.

4.6 ALGORITHM USED

In this following project of classifying the reviews we have used the following classification algorithms after the data pre-processing and feature extraction. The algorithms are:-

- Naïve Bayes Algorithm
- Logistic Regression
- Decision Tree Classification
- Random Forest
- Support Vector Machine

4.6.1 NAÏVE BAYES ALGORITHM

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Bayes' Theorem is stated as:

$$P\left(\frac{h}{d}\right) = \frac{\left(P\left(\frac{d}{h}\right) * P(h)\right)}{P(d)}$$

Where

❖ P (h|d) is the probability of hypothesis h given the data d. This is called the posterior probability.

❖ P (d|h) is the probability of data d given that the hypothesis h was true.

❖ P (h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.

❖ P (d) is the probability of the data (regardless of the hypothesis).

It is called naive Bayes or idiot Bayes because the calculation of the probabilities for each hypothesis is simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value P (d1, d2, d3|h), they are assumed to be conditionally independent given the target value and calculated as P(d1|h) * P(d2|H) and so on.

This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold

4.6.2 LOGISTIC REGRESSION

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

Sigmoid activation

In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

$$S(z) = \frac{1}{1+e^{-z}}$$

Decision boundary

Our current prediction function returns a probability score between 0 and 1. In order to map this to a discrete class (true/false, cat/dog), we select a threshold value or tipping point above which we will classify values into class 1 and below which we classify values into class 2.

$$p \geq 0.5, \text{class}=1$$

$$p < 0.5, \text{class}=0$$

For example, if our threshold was .5 and our prediction function returned .7, we would classify this observation

as positive. If our prediction was .2 we would classify the observation as negative. For logistic regression with multiple classes we could select the class with the highest predicted probability.

Making predictions

Using our knowledge of sigmoid functions and decision boundaries, we can now write a prediction function. A prediction function in logistic regression returns the probability of our observation being positive, True, or “Yes”. We call this class 1 and its notation is

P(class=1)P(class=1). As the probability gets closer to 1, our model is more confident that the observation is in class 1.

COST FUNCTION

Since the hypothesis function for logistic regression is sigmoid in nature hence, The First important step is finding the gradient of the sigmoid function. We can see from the derivation below that gradient of the sigmoid function follows a certain pattern.

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

$$\begin{aligned} J(\theta) &= \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) \\ \mathrm{Cost}(h_\theta(x), y) &= -\log(h_\theta(x)) && \text{if } \mathrm{y} = 1 \\ \mathrm{Cost}(h_\theta(x), y) &= -\log(1 - h_\theta(x)) && \text{if } \mathrm{y} = 0 \end{aligned}$$

This is the required cost function for the logistic regression.

4.6.3 DECISION TREE CLASSIFIER

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

A classification model is typically used to,

- Predict the class label for a new unlabeled data object
- Provide a descriptive model explaining what features characterize objects in each class

A tree can be *“learned”* by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subsetat a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

4.6.4 RANDOM FOREST CLASSIFIER

The Random Forest (RF) classifiers are suitable for dealing with the high dimensional noisy data in text classification. An RF model comprises a set of decision trees each of which is trained using random subsets of features. Given an instance, the prediction by the RF is obtained via majority voting of the predictions of all the trees in the forest. However, different test instances would have different values for the features used in the trees and the trees should contribute differently to the predictions. This diverse contribution of the trees is not considered in traditional RFs. Many approaches have been proposed to model the diverse contributions by selecting a subset of trees for each instance.

Random forest is like bootstrapping algorithm with Decision tree (CART) model. Say, we have 1000 observation in the complete population with 10 variables. Random forest tries to build multiple CART models with different samples and different initial variables. For instance, it will take a random sample of 100 observation and 5 randomly chosen initial variables to build a CART model. It will repeat the process (say) 10 times and then make a final prediction on each observation. Final prediction is a function of each prediction. This final prediction can simply be the mean of each prediction.

- **Step 1** − First, start with the selection of random samples from a given dataset.
- **Step 2** − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- **Step 3** − In this step, voting will be performed for every predicted result.
- **Step 4** − At last, select the most voted prediction result as the final prediction result.

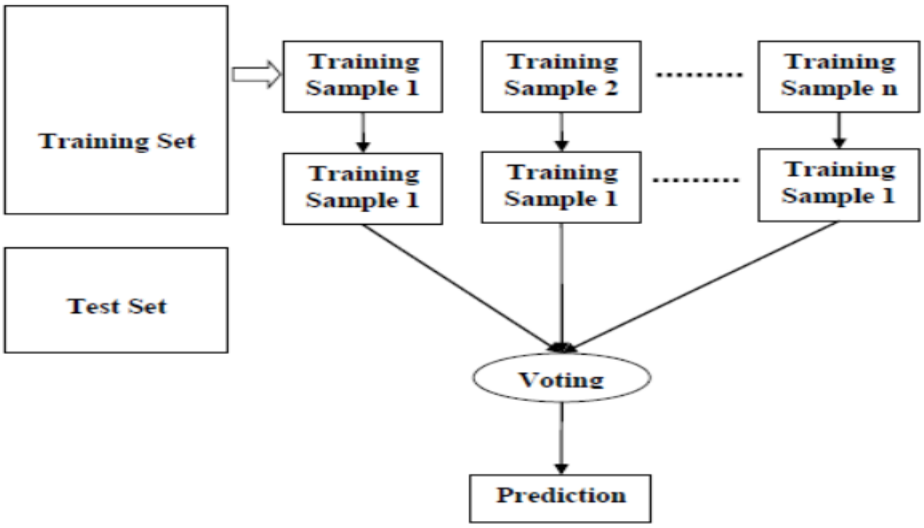


Figure 4.3: Working of Random Forest

4.6.5 SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

Let's consider two independent variables x1, x2 and one dependent variable which is either a blue circle or a red circle.

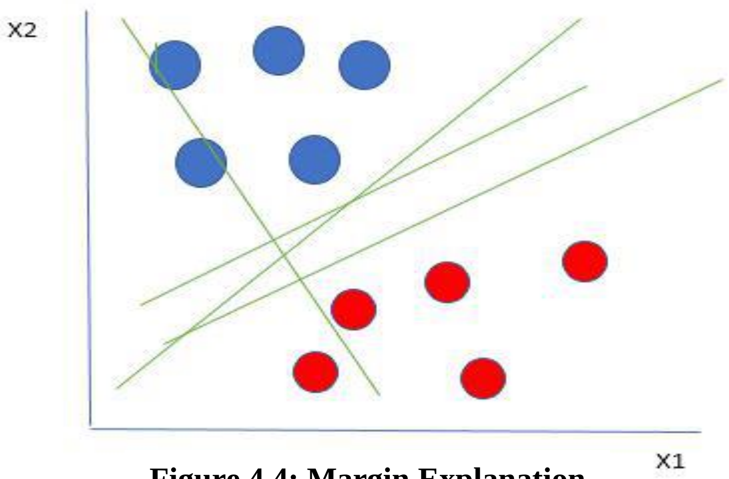


Figure 4.4: Margin Explanation

From the figure above its very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x1, x2) that segregates our data points or does a classification between red and blue circles. So how do we choose the best line or in general the best hyperplane that segregates our data points. One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.

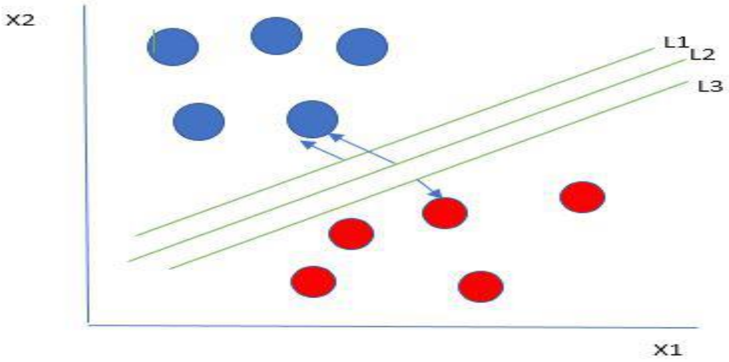


Figure 4.5: Margin Explanation -II

So we choose the hyperplane whose distance from it to the nearest data point on each side is maximized. If such a hyperplane exists it is known as the maximum-margin hyperplane/hard margin. So from the above figure, we choose L2

Let's consider a scenario like shown below

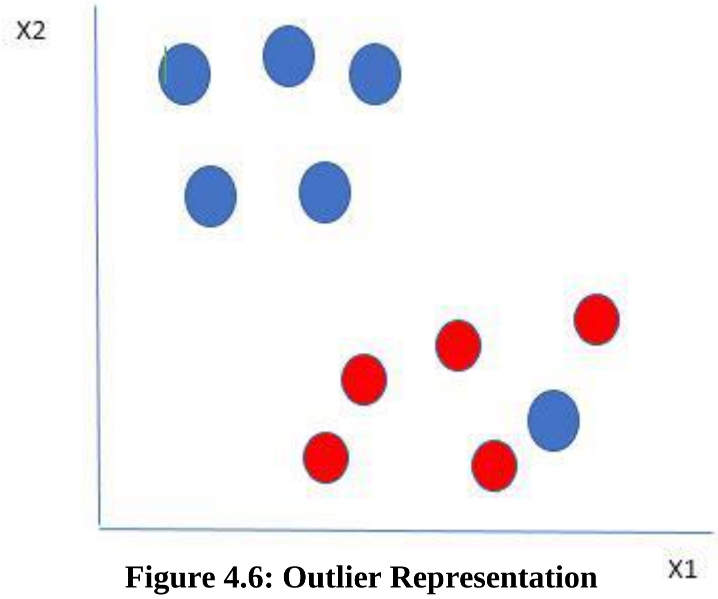


Figure 4.6: Outlier Representation

Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It's simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.

CHAPTER 5
EXPERIMENTAL AND RESULT ANALYSIS

In this section we are going to deal with testing, testing is finding out how well something works. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. There are various techniques for testing the accuracy but we are going to use some of them.

5 .1 ACCURACY

It is the most common evaluation metric for classification problems. It is defined as the number of correct predication as against the number of total predictions. However, this metric alone cannot give enough information to decide whether the model is a good one or not. It is suitable when there are equal numbers of observation in every class.

$$\mathbf{Accuracy} = \frac{\mathbf{TP + TN}}{\mathbf{TP + TN + FP + FN}}$$

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ prediction}$$

5.2 F1_SCORE

The accuracy of a machine learning classification algorithm is one way to measure how often the algorithm classifies a data point correctly. Accuracy is the number of correctly predicted data points out of all the data points. More formally, it is defined as the number of true positives and true negatives divided by the number of true positives, true negatives, false positives, and false negatives. A true positive or true negative is a data point that the algorithm correctly classified as true or false, respectively. A false positive or false negative, on the other hand, is a data point that the algorithm incorrectly classified. For example, if the algorithm classified a false data point as true, it would be a false positive. Often, accuracy is used along with precision and recall, which are other metrics that use various ratios of true/false positives/negatives. Precision and recall are two numbers which together are used to evaluate the performance of classification or information retrieval systems. Precision is defined as the fraction of relevant instances among all retrieved instances. Recall, sometimes Referred to as ‘sensitivity, is the fraction of retrieved instances among all relevant instances. Perfect classifiers have precision and recall both equal to 1. Precision and Recall Formulas:

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP + FP}}$$

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP + FN}}$$

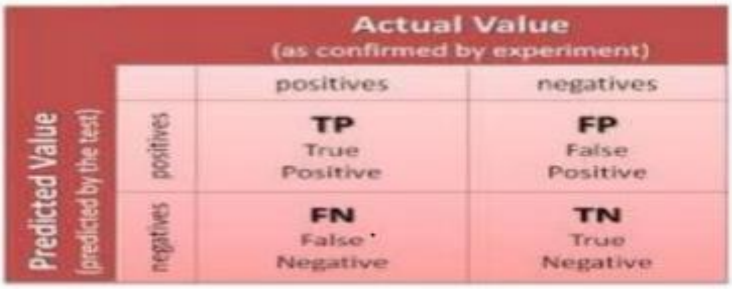
The F-score, also called the F1-score, is a measure of a model’s accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into ‘positive’ or ‘negative’. The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model’s precision and recall. The F-score is commonly used for evaluating information retrieval systems such as search engines, and also for many kinds of machine learning models, in particular in natural language processing. The formula for the standard F1-score is the harmonic mean of the precision and recall. A perfect model has an F-score of 1.

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

$$\mathbf{F_1} = \frac{\mathbf{TP}}{\mathbf{TP} + \frac{1}{2}\ (\mathbf{FP + FN})}$$

5.3 CONFUSION MATRIX

It is also known as Error matrix, which is a table representation that shows the performance of the model. It is special kind of table having two dimensions- “actual”, labelled on xaxis and “predicted” on y-axis. The cells of the table are the number of predictions made by the algorithm. True Positives: It is correctly predicted positive values. True Negatives: It is correctly predicted negative values. False Positives: It is incorrectly predicted negative values as positive values. False Negatives: It is incorrectly predicted negative values as positive values.



		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	TP True Positive	FP False Positive
	negatives	FN False Negative	TN True Negative

Figure 5.1: Confusion Matrix

5.4 RESULT ANALYSIS

We have performed six algorithms on our dataset namely Naïve Bayes algorithm, K Nearest Neighbors, Logistic Regression, Decision tree and Random forest. And here we have tabulated the results.

5.4.1 ACCURACY SCORE

- **Using Count Vectorizer**

NAME OF THE ALGORITHM	ACCURACY (IN %)
NAIVE BAYES	**40.92**
LOGISTIC REGRESSION	**56.63**
DECISION TREE	**44.35**
RANDOM FOREST	**54.47**
SVM	**53.23**

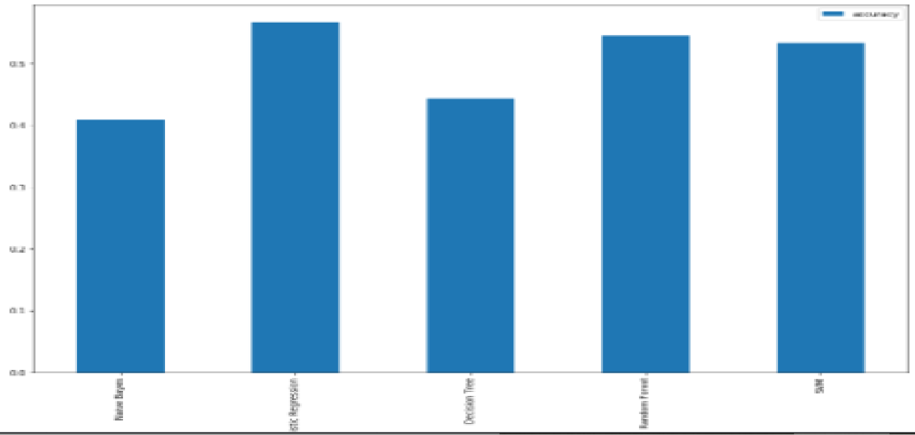
Table 1 : Count Vectorizer Accuracy Table

- **Using TFIDF vector**

NAME OF THE ALGORITHM	**ACCURACY (IN %)**
LOGISTIC REGRESSION	**62.06**
DECISION TREE	**44.95**
RANDOM FOREST	**55.36**
SVM	**61.04**

Table 2: TFIDF Vector Accuracy Table

- **Bar Plot of Count Vectorizer**



- **Figure 5.2: Bar Plot of Count Vectorizer**

- **Bar Plot of TFIDF Vector**

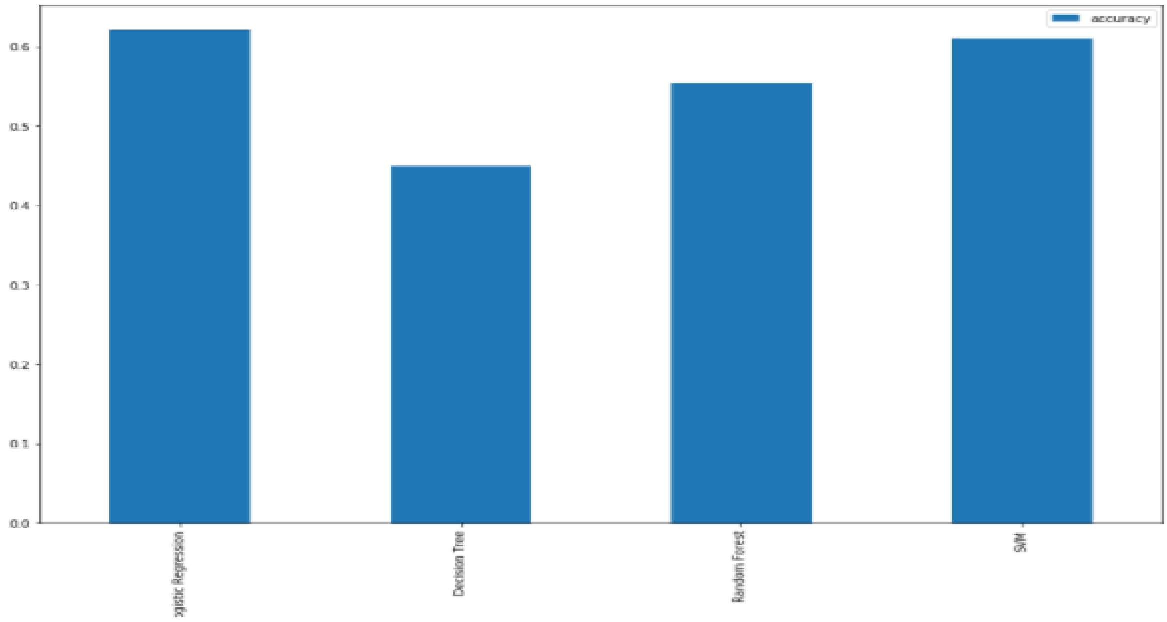
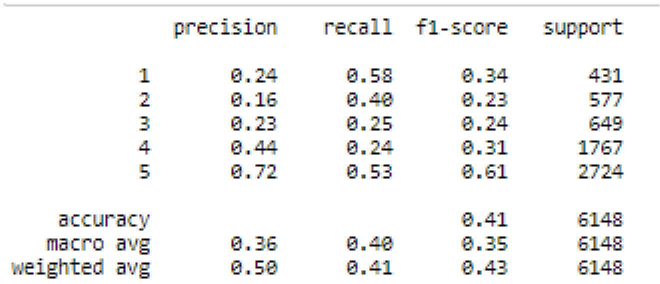


Figure 5.3: Bar Plot of TFIDF Vector

5.4.1 CLASSIFICATION REPORT

CountVectorizer

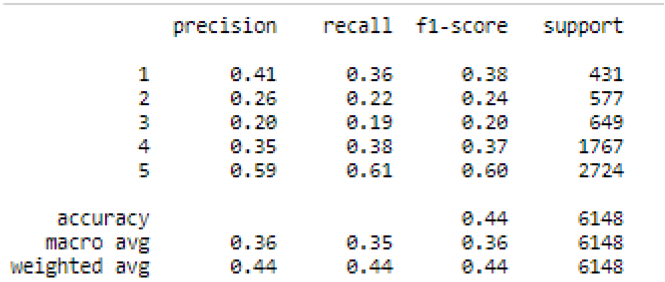
- **For Naïve Bayes**



```
              precision    recall  f1-score   support

           1       0.24      0.58      0.34       431
           2       0.16      0.40      0.23       577
           3       0.23      0.25      0.24       649
           4       0.44      0.24      0.31      1767
           5       0.72      0.53      0.61      2724

    accuracy                           0.41      6148
   macro avg       0.36      0.40      0.35      6148
weighted avg       0.50      0.41      0.43      6148
```

Figure 5.4: Confusion Matrix of Naïve Bayes (CountVectorizer)

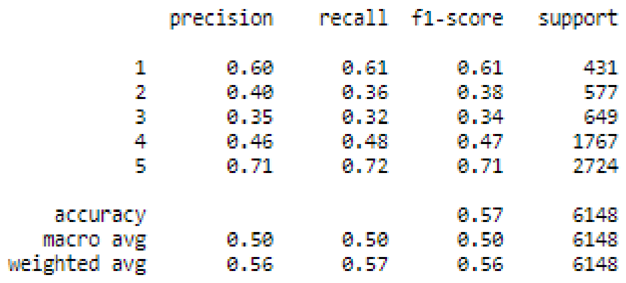
- **For Logistic Regression**



```
              precision    recall  f1-score   support

           1       0.41      0.36      0.38       431
           2       0.26      0.22      0.24       577
           3       0.20      0.19      0.20       649
           4       0.35      0.38      0.37      1767
           5       0.59      0.61      0.60      2724

    accuracy                           0.44      6148
   macro avg       0.36      0.35      0.36      6148
weighted avg       0.44      0.44      0.44      6148
```

Figure 5.5: Confusion Matrix of Logistic Regression (CountVectorizer)

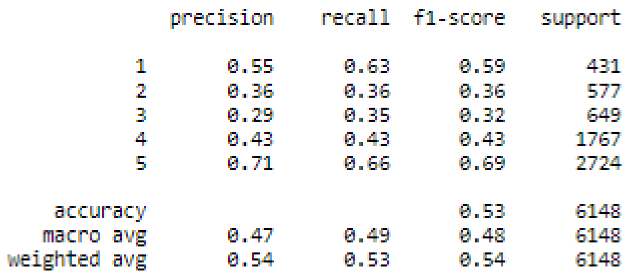
- **For Decision Tree**



```
              precision    recall  f1-score   support

           1       0.60      0.61      0.61       431
           2       0.40      0.36      0.38       577
           3       0.35      0.32      0.34       649
           4       0.46      0.48      0.47      1767
           5       0.71      0.72      0.71      2724

    accuracy                           0.57      6148
   macro avg       0.50      0.50      0.50      6148
weighted avg       0.56      0.57      0.56      6148
```

Figure 5.6: Confusion Matrix of Decision Tree (CountVectorizer)

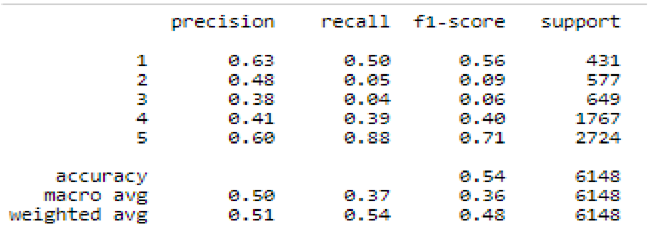
- **For SVM**



```
              precision    recall  f1-score   support

           1       0.55      0.63      0.59       431
           2       0.36      0.36      0.36       577
           3       0.29      0.35      0.32       649
           4       0.43      0.43      0.43      1767
           5       0.71      0.66      0.69      2724

    accuracy                           0.53      6148
   macro avg       0.47      0.49      0.48      6148
weighted avg       0.54      0.53      0.54      6148
```

Figure 5.7: Confusion Matrix of SVM (CountVectorizer)

- **For Random Forest**

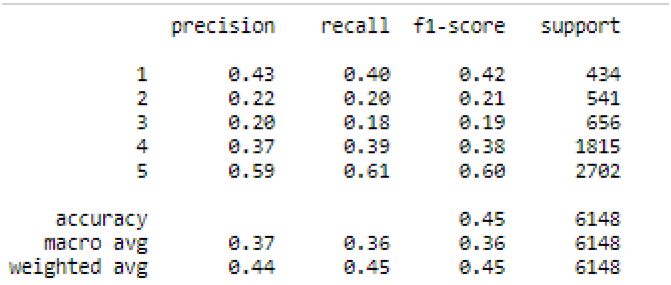


```
              precision    recall  f1-score   support

           1       0.63      0.50      0.56       431
           2       0.48      0.05      0.09       577
           3       0.38      0.04      0.06       649
           4       0.41      0.39      0.40      1767
           5       0.60      0.88      0.71      2724

    accuracy                           0.54      6148
   macro avg       0.50      0.37      0.36      6148
weighted avg       0.51      0.54      0.48      6148
```

Figure 5.8: Confusion Matrix of Random Forest (CountVectorizer)

TFIDF Vector

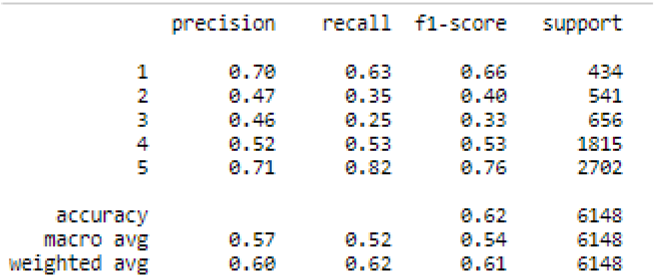
- **For Logistic Regression**



```
              precision    recall  f1-score   support

           1       0.43      0.40      0.42       434
           2       0.22      0.20      0.21       541
           3       0.20      0.18      0.19       656
           4       0.37      0.39      0.38      1815
           5       0.59      0.61      0.60      2702

    accuracy                           0.45      6148
   macro avg       0.37      0.36      0.36      6148
weighted avg       0.44      0.45      0.45      6148
```

Figure 5.9: Confusion Matrix of Logistic Regression ((TFIDF Vector)

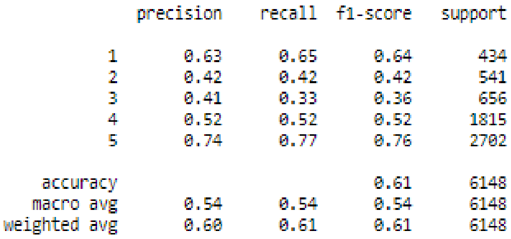
- **For Decision Tree**



```
              precision    recall  f1-score   support

           1       0.70      0.63      0.66       434
           2       0.47      0.35      0.40       541
           3       0.46      0.25      0.33       656
           4       0.52      0.53      0.53      1815
           5       0.71      0.82      0.76      2702

    accuracy                           0.62      6148
   macro avg       0.57      0.52      0.54      6148
weighted avg       0.60      0.62      0.61      6148
```

Figure 5.10: Confusion Matrix of Decision Tree (TFIDF Vector)

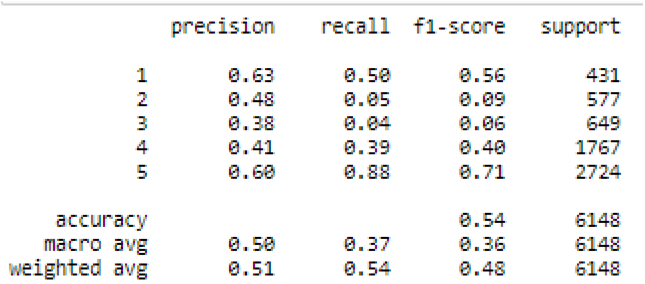
- **For SVM**



```
              precision    recall  f1-score   support

           1       0.63      0.65      0.64       434
           2       0.42      0.42      0.42       541
           3       0.41      0.33      0.36       656
           4       0.52      0.52      0.52      1815
           5       0.74      0.77      0.76      2702

    accuracy                           0.61      6148
   macro avg       0.54      0.54      0.54      6148
weighted avg       0.60      0.61      0.61      6148
```

Figure5.11: Confusion Matrix of SVM (TFIDF Vector)

- **For Random Forest**



```
              precision    recall  f1-score   support

           1       0.63      0.50      0.56       431
           2       0.48      0.05      0.09       577
           3       0.38      0.04      0.06       649
           4       0.41      0.39      0.40      1767
           5       0.60      0.88      0.71      2724

    accuracy                           0.54      6148
   macro avg       0.50      0.37      0.36      6148
weighted avg       0.51      0.54      0.48      6148
```

Figure 5.12: Confusion Matrix of Random Forest (TFIDF Vector)

CHAPTER 6
USER INTERFACE AND SOFTWARE TESTING

After developing our machine learning model, training it with the available data-set and testing it with the test dataset we have received satisfactory accuracy and so to convert the model to be implemented as a application we have used a flask framework to provide the user inte rface to the model. And deployed the model for use and testing.

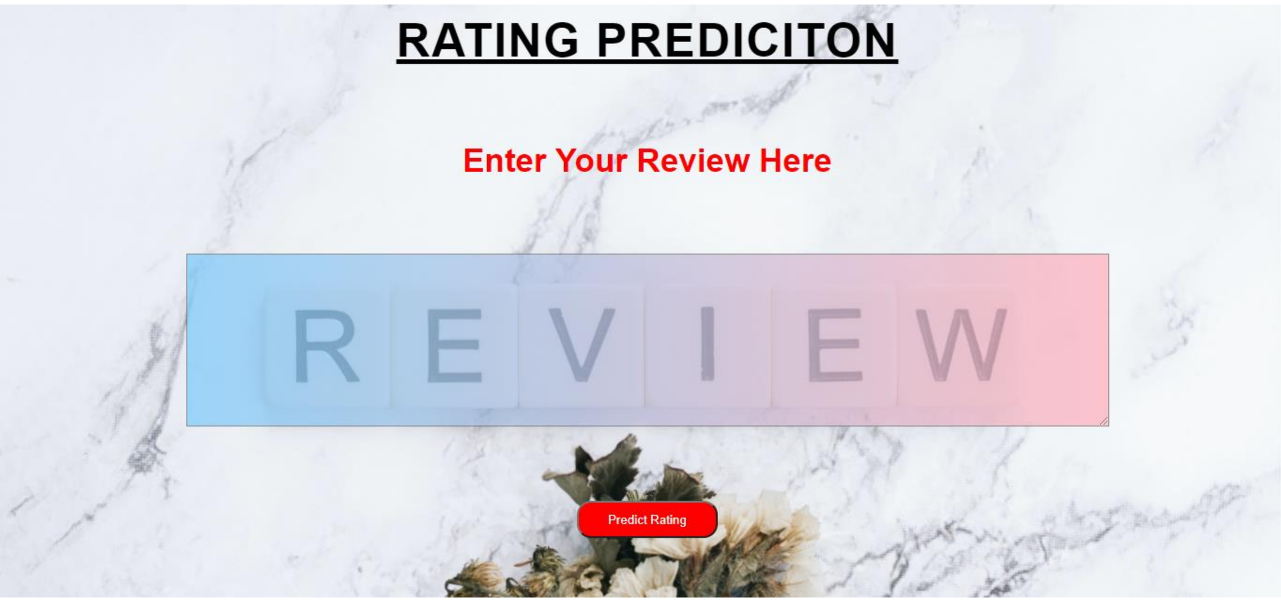
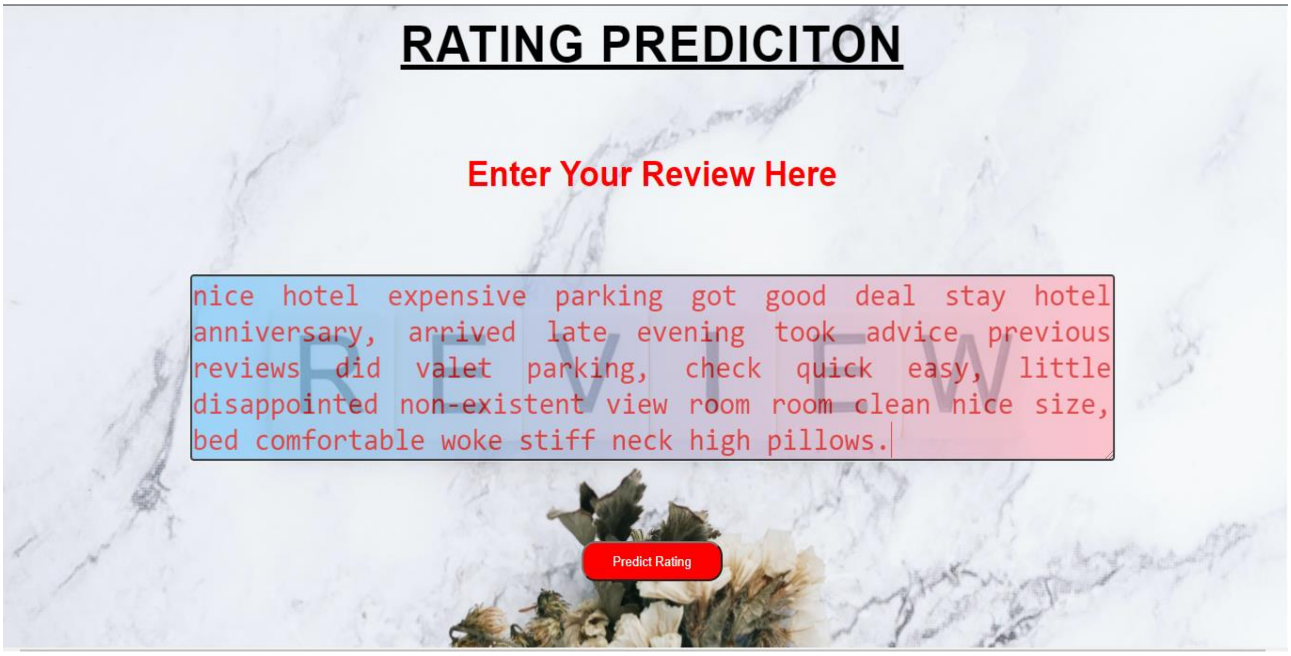


Figure 5.1: Home Screen Of the Deployed Model

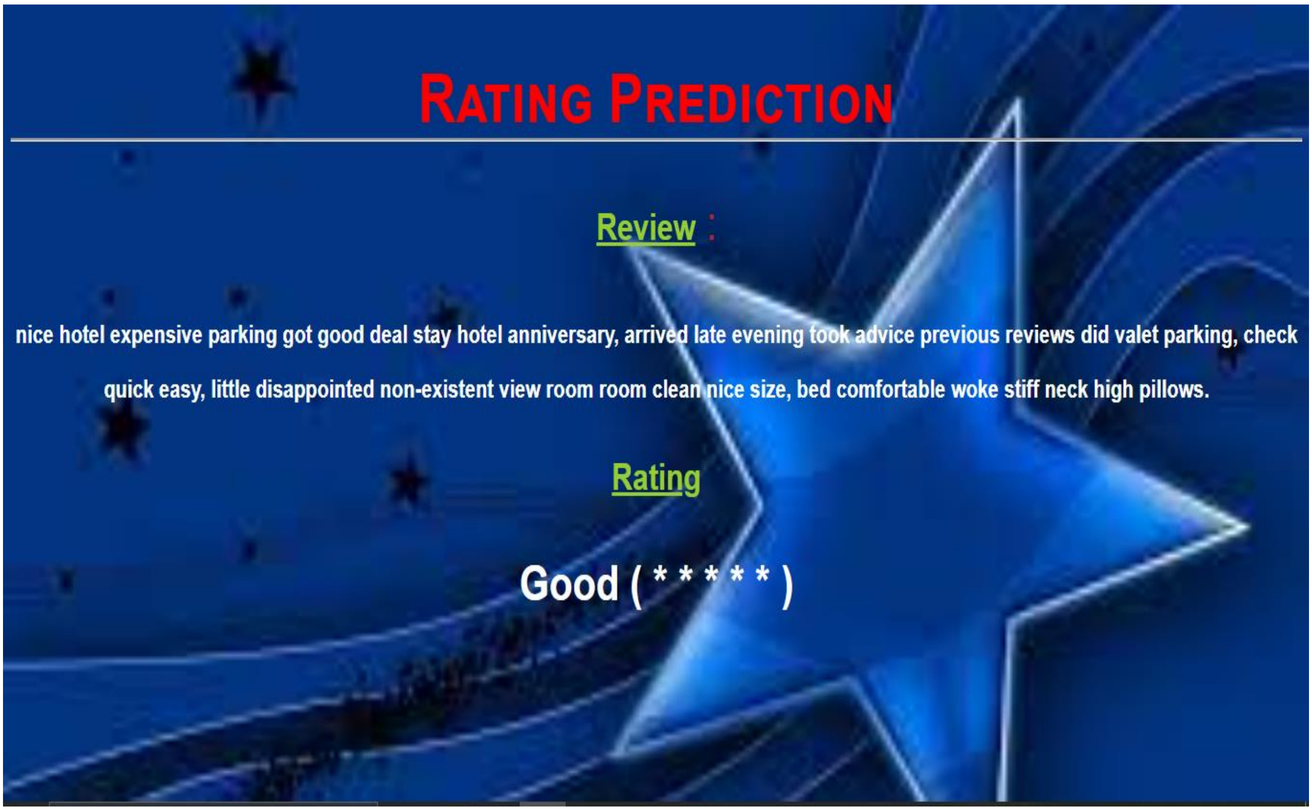
This is the first screen that appears at our deployed model and asks for the input which is to be tested.



nice hotel expensive parking got good deal stay hotel anniversary, arrived late evening took advice previous reviews did valet parking, check quick easy, little disappointed non-existent view room room clean nice size, bed comfortable woke stiff neck high pillows.

Figure 5.2: Inputting the review to predict rating

After we enter the features inside the box and click on the predict the result appears



RATING PREDICTION

Review :

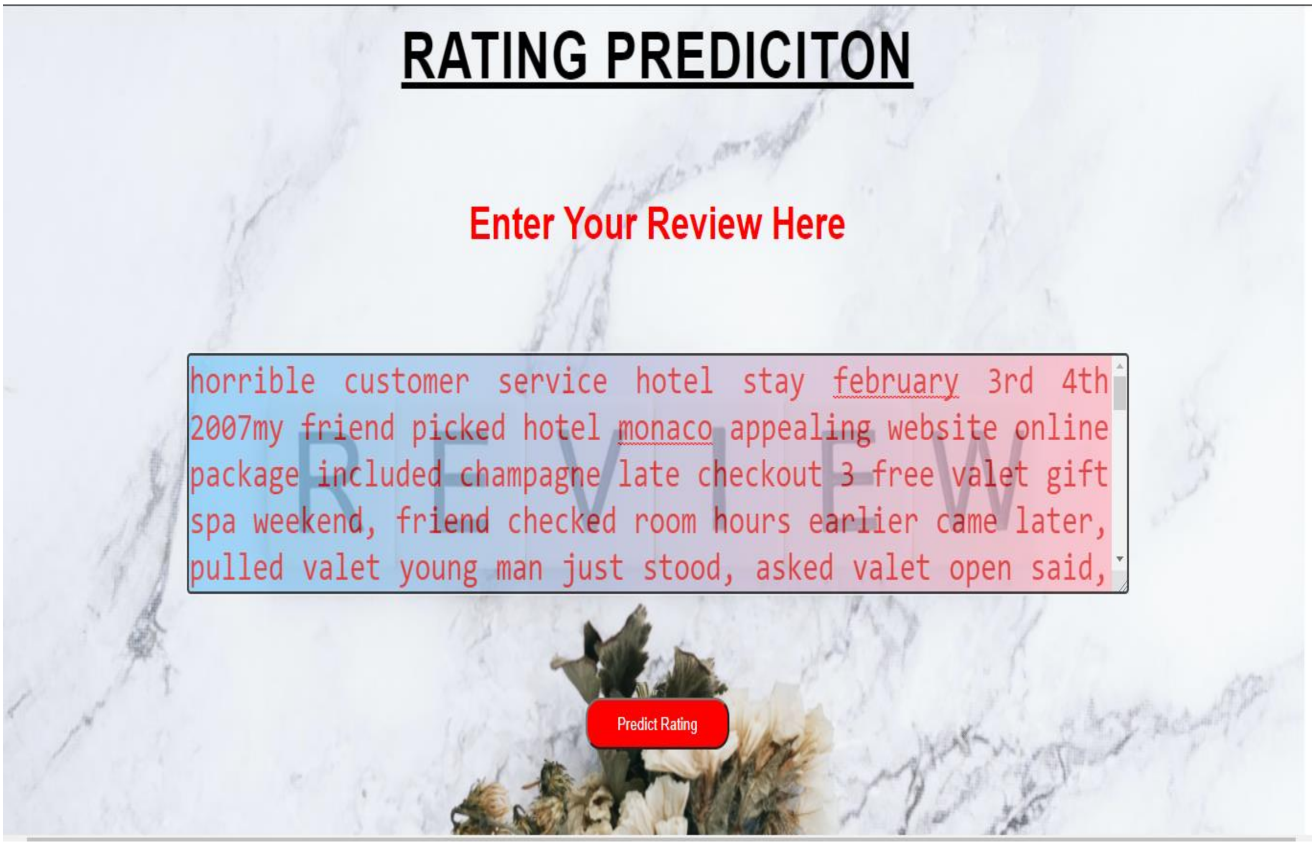
nice hotel expensive parking got good deal stay hotel anniversary, arrived late evening took advice previous reviews did valet parking, check quick easy, little disappointed non-existent view room room clean nice size, bed comfortable woke stiff neck high pillows.

Rating

Good (* * * * *)

Figure 5.3: Result of the given review

Another example: Features Entered



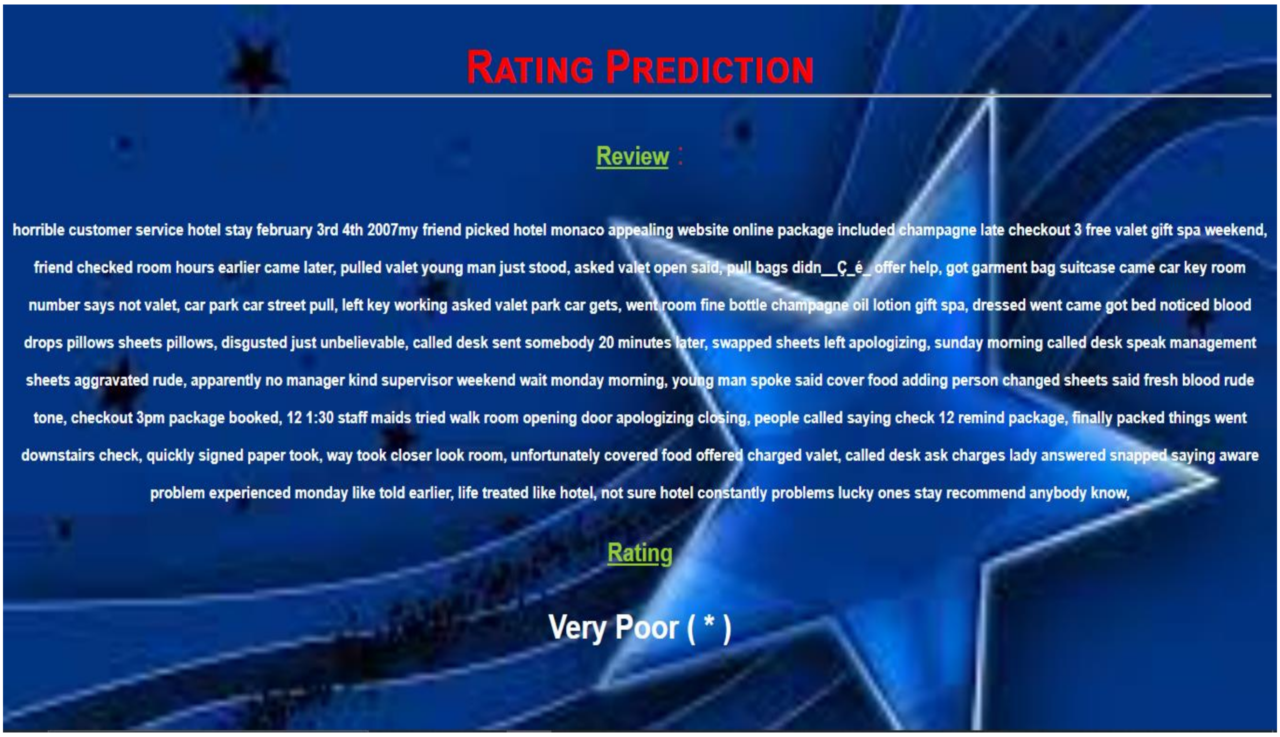
RATING PREDICITON

Enter Your Review Here

horrible customer service hotel stay february 3rd 4th 2007my friend picked hotel monaco appealing website online package included champagne late checkout 3 free valet gift spa weekend, friend checked room hours earlier came later, pulled valet young man just stood, asked valet open said,

Figure 5.4: Inputting the review to predict rating

After we enter the features inside the box and click on the predict the result appears



RATING PREDICTION

Review :

horrible customer service hotel stay february 3rd 4th 2007my friend picked hotel monaco appealing website online package included champagne late checkout 3 free valet gift spa weekend, friend checked room hours earlier came later, pulled valet young man just stood, asked valet open said, pull bags didn__Ç_é_ offer help, got garment bag suitcase came car key room number says not valet, car park car street pull, left key working asked valet park car gets, went room fine bottle champagne oil lotion gift spa, dressed went came got bed noticed blood drops pillows sheets pillows, disgusted just unbelievable, called desk sent somebody 20 minutes later, swapped sheets left apologizing, sunday morning called desk speak management sheets aggravated rude, apparently no manager kind supervisor weekend wait monday morning, young man spoke said cover food adding person changed sheets said fresh blood rude tone, checkout 3pm package booked, 12 1:30 staff maids tried walk room opening door apologizing closing, people called saying check 12 remind package, finally packed things went downstairs check, quickly signed paper took, way took closer look room, unfortunately covered food offered charged valet, called desk ask charges lady answered snapped saying aware problem experienced monday like told earlier, life treated like hotel, not sure hotel constantly problems lucky ones stay recommend anybody know,

Rating

Very Poor (*)

Figure 5.5: Result of the given review

CHAPTER-7
CONCLUSION

Crowd sourcing platforms play a crucial role in travel planning. The tourists try to validate their options using the information shared by the other tourists, that is, crowd sourced information. This digital footprint has been used to model tourists and tourism resources in order to provide personalized recommendations. Specifically in hotel industry, the guests use ratings to classify the different hotel services.

This paper presents a survey regarding analysis and rating prediction of hotel ratings including:
Sentiment analysis for the hotel reviews has been carried out labelling reviews as positive sentiments which include word like- happy, amazing, tasty, nice, pretty as well as negative sentiments which include words bad, disgusting, sad, and disappointed, etc. The whole point of the analysis is to provide suitable recommendations to the customers to select the best available option and to the business owner for successful decision making, using sentiment-based results, and implying sentiments. Moreover, the sentiment analysis in this work has been applied to determine the attitude of customers through online feedbacks given by them on hotel services, food, staff, and ambiance of the respective hotel.

REFERENCES

- Sentiment Analysis of Review Datasets using Naïve Bayes'
- Leal, F., Malheiro, B., & Burguillo, J. C. (2017b). Prediction and analysis of hotel ratings from crowd-sourced data. In World Conference on Information Systems and Technologies, (pp. 493–502). Springer.
- World Committee Tourism Ethics. (2017). Recommendations on the responsible use of ratings and reviews on digital platforms. In 3rd International Congress on Ethics
- Technical Report: http://www.ics.uci.edu/vpsaini/files/technical report.pdf
- Scaria, Aju Thalappillil, Rose Marie Philip, and Sagar V. Mehta. ”Predicting Star Ratings of Movie Review Comments.”
- Qu, Lizhen, Georgiana Ifrim, and Gerhard Weikum. ”The bag-of-opinions method for review rating prediction from sparse text patterns.” Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, 2010.