A Project Report on

Hotel Recommender System and Sentiment Analysis of Customer Reviews

Submitted in partial fulfilment for the award of the degree of

MBA

In **Business Analytics**

Submitted by

Mahapara Gayasuddin
R19DM004

Under the Guidance of

Mr Krishna Kumar Tiwari
RACE | Jio

REVA Academy for Corporate Excellence
REVA University
Rukmini Knowledge Park, Kattigenahalli,
Yelahanka, Bangalore – 560064

February 2021

REVA
UNIVERSITY
Bengaluru, India

Candidate's Declaration

I, Mahapara Gayasuddin hereby declare that I have completed the project work towards the PGDM in Business Analytics at, REVA University on the topic entitled Hotel Recommender System and Sentiment Analysis of Customer Reviews under the supervision of Mr Krishna Tiwari Chief Mentor, RACE | Jio. This report embodies the original work done by me in partial fulfilment of the requirements for the award of the degree for the academic year 2021.

Mpara G.

Place: Bengaluru

Date:11-02-2021

MAHAPARA GAYASUDDIN

REVA
UNIVERSITY
Bengaluru, India

Certificate

This is to Certify that the PROJECT work entitled Hotel Recommender System and Sentiment Analysis of Customer Reviews carried out by Mahapara Gayasuddin with R19DM004, is a bonafide student of REVA University, is submitting the project report in fulfilment for the award of BA05 in Business Analytics during the academic year 2021. The Project report has been tested for plagiarism and has passed the plagiarism test with the similarity score of less than 15%. The project report has been approved as it satisfies the academic requirements in respect of PROJECT work prescribed for the said Degree.

Mr Krishna Kumar Tiwari
Guide

Dr Shinu Abhi
Director

Krishna Kumar tiwari

External Viva
Names of the Examiners

1. Indrajit Kar, Head of AI-Chief Architect & Data Scientists, Siemens
2. Pradeepta Mishra, Associate Principal & Head of AI, LTI -Larsen and Toubro Infotech

Place: Bengaluru
Date: 11.02.21

REVA
UNIVERSITY
Bengaluru, India

Acknowledgement

During this hard time even when it wasn’t possible to connect physically, we all have come together through virtual platform with the support of Dr. Shinu and under the guidance of my mentor Krishna Sir, with the help of my classmates and Reva technical staff, I have done my project successfully. So, I would like to thank each and every one for their support.

I would like to thank Hon’ble Chancellor, Dr. P Shayma Raju, Vice Chancellor, Dr. K. Mallikharjuna Babu, and Pro Vice Chancellor, Dr. M. Dhanamjaya for providing us great Infrastructure and quality education at Reva University.

Place: Bengaluru
Date: 11-02-2021

REVA UNIVERSITY
Bengaluru, India

Similarity Index Report

This is to certify that this project report titled **Hotel Recommender System and Sentiment Analysis of Customer Reviews** was scanned for similarity detection. Process and outcome are given below.

Software Used: **Turnitin**
Date of Report Generation: **01-Mar-2021**
Similarity Index in %: **8%**
Total word count: **6780**
Name of the Guide: **Mr Krishna Kumar Tiwari**

Place: Bengaluru

Name of the Student: Mahapara Gayasuddin

Mpara G.

Date: 11-02-2021

Signature of Student

Verified by: Andrea Brian C

Signature
Dr. Shinu Abhi,
Director, Corporate Training

List of Abbreviations

Sl. No	Abbreviation	Long Form
1	RS	Recommender System
2	CF	Collaborative Filtering
3	RBM	Restricted Boltzmann Machines
4	NLP	Natural Language Processing
5	SA	Sentiment Analysis
6	RNN	Recurrent Neural Network
7	CNN	Convolutional Neural Network
8	FM	Factorization Machines

List of Figures

No.	Name	Page No.
1.1	Recommendation Techniques	10
5.1	CRISP-DM Data Mining Methodology	15
7.1	Raw Hotel Dataset	19
7.2	Distribution of Rating	19
7.3	Top 20 Most Reviewed Hotels	20
7.4	Bottom 20 Most Reviewed Hotels	21
7.5	Sentiment Distribution	21
7.6	Frequent User-Name	22
7.7	Reviews Distribution by City	22
7.8	Hotel Category Distribution by Hotel	23
8.1	Data pre-processing	24
8.2	Pre-processed Data	25
8.3	Rating Data	25
9.1	Comparison Matrix of Models using Count Vectorizer	28

9.2	Average F1 Score using Count Vectorizer	28
9.3	Comparison Matrix of Models using TF-IDF Vectorizer	29
9.4	Average F1 Score using TF-IDF Vectorizer	29
9.5	RNN Loss Curve	31
9.6	RNN Accuracy Curve	31
9.7	CNN Loss Curve	33
9.8	CNN Accuracy Curve	33
9.9	Loss Curve per Iterations	34
9.10	Top Popular Hotels to Recommend	35
9.11	Working of Restricted Boltzmann machine	40
9.12	Error per Epoch	41

List of Tables

No.	Name	Page No.
Table 9.1	RMSE for the Test data	37
Table 9.2	Accuracy of the Models	43
Table 11.1	The Predicted rating of some samples with their actual values	45

Abstract

The tourism industry supports different enterprises, programs, and other sectors. In tourism, people prefer to travel to different places and stay in various places beyond their normal climate. The tourism industry has played a critical role in providing competitive and comparable rates for hotel bookings to Internet users. In the current scenario, many users have been shown to express their views by giving their feedback in various ways. (Mishra et al., 2019)

So, the first goal of this project is classifying hotel review as positive or negative based on the ratings provided by the customers thereby analysing the sentiment of a customer. Using hotel reviews from various sources we applied various machine learning techniques. (Sarkar, 2016)

To transform the textual documents into numerical feature vectors, we employed “Term Frequency (TF) “and “Inverse Document Frequency (IDF)” that was applied to the text data. (Mishra et al., 2019) The edge for word occurrence was added to feature selection using min df/max df, “PCA” and “Singular Value Decomposition (SVD)”. (Sarkar, 2016)(Mishra et al., 2019)

One of the primary things to try is to book an honest place to stay when planning a visit. Booking a hotel online is also an awesome activity for each destination, with thousands of hotels to decide on. We agreed to work out the task of recommending hotels to consumers, inspired by the value of such circumstances.

The second goal is building recommender engines to provide recommendations to different users and build different machine learning models to predict the rating of each hotel.

So, the issue we're trying to research here is how to construct efficient recommendation systems that can predict hotels that customers like the most and have the most potential to purchase, and the feelings associated with the customer's stay reviews. We introduce these algorithms and then evaluate them to do comparisons and produce results on some existing datasets.

Keywords: Collaborative Filtering, Recommender System, Natural Language Processing, Sentiment Analysis, Term Frequency, Inverse Document Frequency.

Contents

Candidate’s Declaration 2

Certificate 3

List of Abbreviations 6

List of Figures 6

List of Tables 7

Abstract 8

Chapter 1: Introduction .... **Error! Bookmark not defined.**

Chapter 2: Literature Review .... **Error! Bookmark not defined.**

Chapter 3: Problem Statement 14

Chapter 4: Objectives of the Study .... **Error! Bookmark not defined.**

Chapter 5: Project Methodology 16

Chapter 6: Business Understanding .... **Error! Bookmark not defined.**

Chapter 7: Data Understanding 20

Chapter 8: Data Preparation .... **Error! Bookmark not defined.**

Chapter 9: Data Modeling .... **Error! Bookmark not defined.**

Chapter 9: Data Evaluation 44

Chapter 10: Deployment 46

Chapter 11: Analysis and Results 47

Chapter 12: Conclusions and Recommendations for future work .... **Error! Bookmark not defined.**

Bibliography 49

Appendix 52

- Plagiarism Report 52
- Publications in a Journal/Conference Presented/White Paper .... **Error! Bookmark not defined.**
- Any Additional Details .... **Error! Bookmark not defined.**

Chapter 1: Introduction

What is analytical sentiment?

The method of using natural language processing, text analysis, and statistics to analyse consumer sentiment is sentiment analysis. The renowned companies recognize what their consumers feel, what they say, how they say it, and what they mean from that. Consumer sentiment mentioning a brand in a form of reviews can be found on various social media platforms. Deep learning approaches have pushed sentiment analysis, as well as many other fields, to the forefront of algorithms on the cutting edge." Today, to derive and classify the emotions of words statistics, natural language processing (NLP) and text analysis are used". (Algorithmia, 2018)

Why and what was used to measure sentiment?

Branch performance monitoring

The most well-known applications of sentiment analysis are to provide a complete 360 view of the consumers and stakeholders viewing the brand, product, or company. Product and social reviews, for example, can reveal crucial insights into whether the organisation is doing correct or incorrect. Companies can use sentiment analysis to ascertain the effect of a new product, marketing campaign, or consumer response to current business news on social media. This is delivered as a service by private firms including Unamo. (Algorithmia, 2018)

Client service

Sentiment or intent analysis is also used by customer service agents to sort the incoming user email automatically into "urgent"/"not urgent" buckets depending on the sentiment of the email, recognizing disgruntled customers proactively. The officer then focuses their efforts to concentrate on the users who have the most urgent needs first. As customer service becomes more automated thanks to machine learning, it becomes more important to consider the sentiment and purpose of a situation. (Algorithmia, 2018)

Market research and analysis

In market intelligence, analysis of sentiment is used to explain why customers react to something or do not react to something else. Quite a lot of these applications are there up and running. In its Multi-Perspective Answers product, Bing recently introduced sentiment analysis. The technology is almost definitely used by hedge funds to forecast market volatility based on public opinion. (Algorithmia, 2018)

Recently, finding an acceptable hotel venue and booking accommodation has become a crucial problem for travellers. Because of the availability of information's online (Tavana et al., 2020), online hotel searches have risen at a much faster rate and have become time-consuming. Recommendation systems (RSs) are becoming relevant nowadays because of their ability to provide information about the product or service needed and help to make decisions. It has become difficult to receive a recommendation about a hotel "when dealing with textual hotel reviews, numerical ranks, votes, scores, and several video views". (Hong-Xia, 2019). The collaborative filtering (CF) method is the most commonly used recommendation techniques. (Ramzan et al., 2019)

"Recommender systems are information filtering systems that address the issue of information overload by filtering a large amount of dynamically generated information from vital information fragments according to the expectations, interest, or observed actions of the user about the object". Based on the user's profile, the proposed system would predict if a particular customer would want an object. Recommendation systems have been proven to enhance the efficiency of the decision-making process. It helps to enhance sales in an e-commerce platform as they are well-organized ways of selling more merchandise. Recommended frameworks in scientific libraries benefit users by encouraging them to step beyond looking for catalogues. (Isinkaye et al., 2015)

Working of Recommendation system framework:

The three algorithms used in the recommended systems will be distinguished between:

- ***Content-based*** systems, using detailed knowledge that uses features.

- ***Collaborative filtering*** systems, User-item interactions are facilitated by.
- ***Hybrid systems***, combining both types of data to avoid problems caused when operating with just one type of information. (Tyrolabs, 2019)

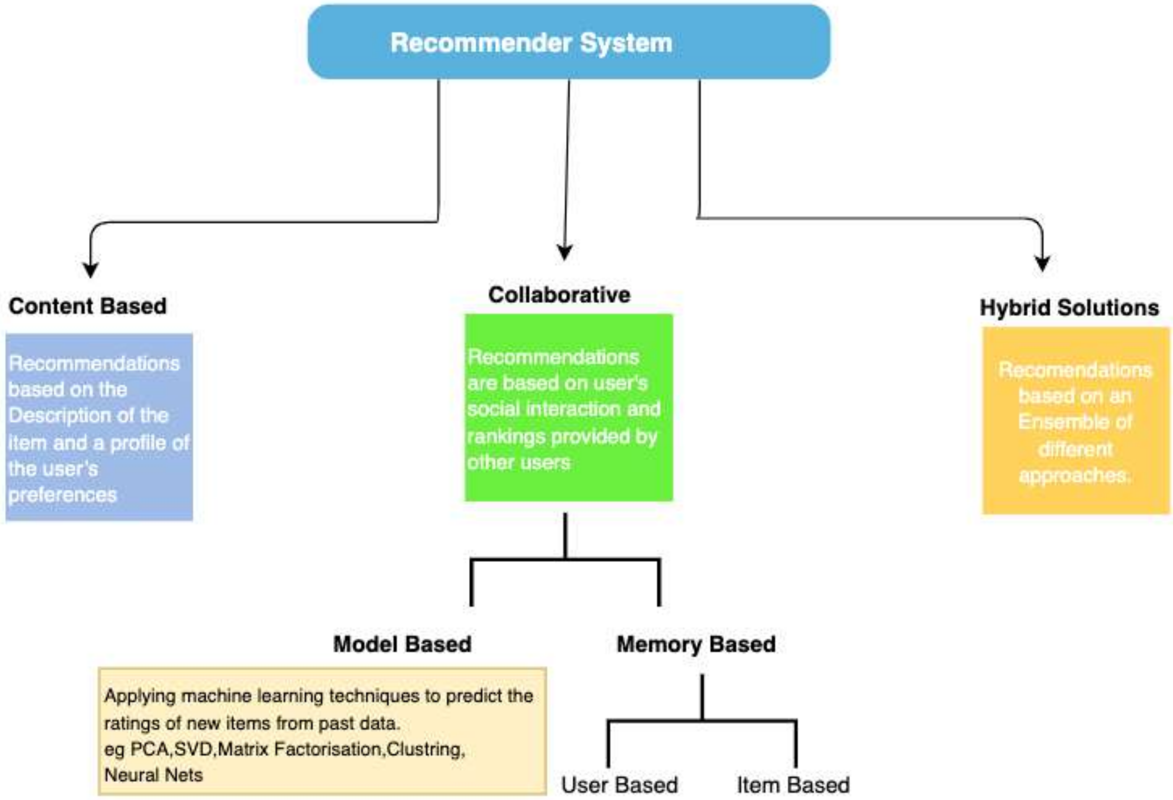

Recommender System
Content Based
Recommendations based on the Description of the item and a profile of the user's preferences
Collaborative
Recommendations are based on user's social interaction and rankings provided by other users
Hybrid Solutions
Recomendations based on an Ensemble of different approaches.
Model Based
Applying machine learning techniques to predict the ratings of new items from past data.
eg PCA,SVD,Matrix Factorisation,Clustring, Neural Nets
Memory Based
User Based
Item Based

Figure 1.1. Recommendation Techniques

Chapter 2: Literature Review

Gomathi et al. propose a machine-learning algorithm that relies on tripadvisor.com search data to overcome the complexity of personalized restaurant discovery. The main purpose of the research paper is to include a more precise and accessible list of recommended restaurants. The conclusion and results indicate that high precision is reached by the proposed method. (Gomathi et al., 2019)

This paper proposes a hotel recommendation based on sentiment analysis using the "TF-IDF" method and Cosine Similarity to suggest additional hotel choices supported by their feedback. "The purpose of "TF-IDF" is to determine the weight meaning of the frequency of words or documents" and thus the similarity of the cosine allows the dataset of sentiment to extract identical kinds of values. (Mishra et al., 2019)

A shared method of filtering for a system of music recommenders. To quantify the similarity of users and things such as Euclidean distance, cosine metric, Pearson correlation et al., they explored different metrics for the experimental purpose. Finally, various assessment measures that indicate the efficacy of the recommendation framework were compared. (Shakirova, 2017)

A recommendation framework that offers information about how to develop powerful recommendation systems that can anticipate goods that clients want the most and have the greatest potential to shop for. The paper explores the architecture, Item Similarity, Bipartite Projection and Spanning Tree, of three new recommendation systems. They can be used, based on the data of all other users and their reviews inside the system, to estimate the rating for a product that a consumer has never tested. (Cheng, 2019)

Different NLP methods are used in this paper on a Corpus of "Hotel Review" to work out the ranking of a hotel based on earlier users to different criteria and generates a user-item-feature database. It also discusses the Cold Start issue when extracting user feedback. (Y. Sharma, 2015)

First, the "Restricted Boltzmann collaborative filtering" algorithm machines are used to predict the item's user rating is r1; then the user interest preference information is used to evaluate the rating preference model of a user, and predict the item's user rating is r2; Finally, the linear regression algorithm is used to validate the weights of r1 and r2 to predict the ultimate regression algorithm. The experimental results show that the proposed algorithm will boost the predicted accuracy of the recommendation method in this paper. (Ge et al., 2019)

Chapter 3: Problem Statement

During the decision-making process, online reviews play an important role. When people have tons of various hotels to remain at, good reviews can help them decide the hotel to stay at. Good reviews can help them make their final judgement and might successively improve the consumer experience. Also, this is often good because the hotel industry will do more business if more people can find them. and successively the sales will increase, and improve the merchandise by understanding customer’s needs and can help attract many people to their location.

It can also be a major part of the approach to marketing analysis and customer care. Not only do you see what people think about their own company or services, but you can also see what your rivals believe. The consumers' overall consumer experience is also easily exposed to sentiment analysis.

The hotel dataset for different hotels is going to be considered in this project. To perform the feeling analysis, the feedback and ratings provided by the user to different hotels as well as reviews on the user's experience during the stay will be considered. The main goal for this project is to build a “model for predicting the user rating, the usefulness of the review and the most similar recommendation”.

To conduct the sentiment analysis of customer feedback using different machine learning techniques and to work with various collaborative filtering (CF) techniques to evaluate the relationship between users and hotel interdependencies to find new connections between user products. The majority of “CF” models are based on the user-item rating matrix in which a user is represented by each row, each column represents an item. “Entries in this matrix are ratings that users give to products”. So, for hotel recommendation three models have been developed i.e. Item-Item Collaborative Filtering, MF CF, ALS and RBM to recommend users the appropriate hotels.

Chapter 4: Objectives of the Study

The agenda of the “recommender system” is to “help the users to find relevant items". The objective was to specialize in the ability to understand the user’s taste for items never seen before or experienced numerically or to provide users with lists of items that are ranked following the user’s taste. The prevailing, though limited, consideration of the “recommendation” issue has been very beneficial in several respects to advance study. (Nnach & Ja, 2016)

A variety of purposes can be fulfilled by recommendation systems from both customers and suppliers. However, most of the needs are greatly underexplored, although many of them are potentially more consistent with recommenders' real-world aspirations. It, therefore, becomes important to review and objectify the potential aims of the recommenders and their activities. (Nnach & Ja, 2016)

The Recommender framework will predict whether or not an item is required by a specific user to support the user profile. To find out the products that fit the taste & interests of the users, a recommendation system uses data analysis techniques. The final objective of every recommendation engine is to improve demand and connect with users. Usually, suggestions speed up searches and make it easier for consumers to access the material they are interested in and impress them with deals they would not have searched for. (Tyrolabs, 2019)

The consumer is likely to shop/consume more content and continues to sense familiar and appreciate additional items. The business gains a competitive advantage by understanding what a consumer wants, and thereby reduces the chance of falling them to a rival. To supply users with the additional value of proposals for systems and goods that are attractive. Moreover, it encourages organizations to put themselves before their competitors and ultimately boosting their profits. (Tyrolabs, 2019) Giving good recommendations can help users spend less time looking for hotels of their kind. This, in turn, will allow the service to continue and offer a better experience.

Chapter 5: Project Methodology

This project aims to create a collaborative hotel recommender system by integrating multiple machine learning algorithms. The model tries to provide accurate prediction and suggestions to which hotel the user might like and the sentiment analysis of the user reviews. The project follows the procedures of "Cross-industry Standard Process for Data Mining" (CRISP-DM), which is a “data mining process” that is commonly used in industries to tackle problems. CRISP-DM breaks problems into six phases:

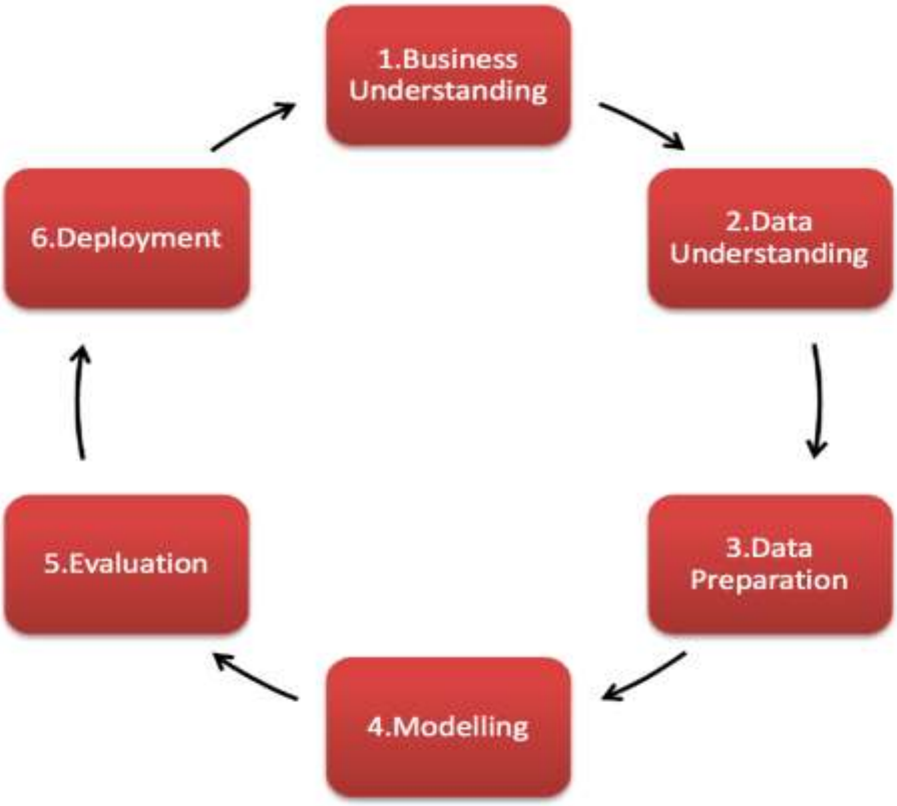


Figure 5.1: CRISP-DM Data Mining Methodology

Workflow

- Collecting data and applying data wrangling methods.
- Starting exploratory data analysis to find trends and storytelling.
- Conduct further data analysis to identify relationships between different variables.

- Perform in-depth Sentiment analysis of the Reviews provided by the Customers for their stay in the hotel using different machine learning technique.

- Perform in-depth analysis using collaborative filtering and machine learning techniques to recommend and predict.

- Conclusion and Future works.

Chapter 6: Business Understanding

All online travel agencies aim to follow the AI-driven level of personalization set by Amazon and Netflix. Additionally, the world of online travel has become a highly competitive space where brands attempt to capture our attention (and wallet) by recommending, comparing, matching and sharing.

If a good organization is managed by anyone, one will possibly survive without a system of recommendation. However, if you want to use the data facility to create a far greater customer experience and improve profits, you should think about seriously adding a recommender system. (Tyrolabs, 2019)

Booking an honest place to stay is one of the key things to do when planning a holiday. Booking a hotel online is also an awesome activity for each destination, with thousands of hotels to decide on. We chose to concentrate on the task of recommending hotels to consumers, inspired by the relevance of these circumstances. For a “hotel recommendation”, we used Dataset collected from multiple sources and that they were combined and converted into one dataset, which has a kind of features that helped us gain a deep understanding of the process that makes a user prefer some hotels over others. This role of hotel recommendation is aimed at predicting and recommending hotels to a consumer who is more likely to book. A “user-item rating matrix” is supported in most “CF” algorithms, where an individual row constitutes a user, individual column an item. Users give objects scores, which are defined by entries in this matrix. Top hotels are recommended supported a particular Location (City) for the convenience of the users.

Best hotels are recommended out of the available lists of hotels for the users to go for if they are looking for the top hotels in the country.

Finally, a Top-10 hotel is recommended supported user preference.

Sentiment analysis of the Reviews is performed to know the emotions of the users during their stay within the hotel.

Chapter 7: Data Understanding

DATA COLLECTION:

- To obtain information on the hotels, data are collected from data. World website that was provided by Datafiniti's Business Database, other data are web scraped using a web scraping software tool Parse hub from the TripAdvisor's website. Finally, all the collected data were merged to make a dataset that contains customer reviews and hotel information which was used for the sentiment analysis and therefore the hotel recommendation to the users.

- The dataset includes hotel id, hotel name, address, location, title, user name, reviews, rating, hotel type, price, hotel category, user id. The data were found to be a mixture of user features and hotel features, with user data reflecting user names and the hotel at which they stayed. The hotel features mainly constitute its geological location.

- **Attributes information:**
  user_id- Each user is labelled with a unique id
  Hotel_id- Each hotel is labelled with a unique id
  Rating- The corresponding hotel ranking by the corresponding user

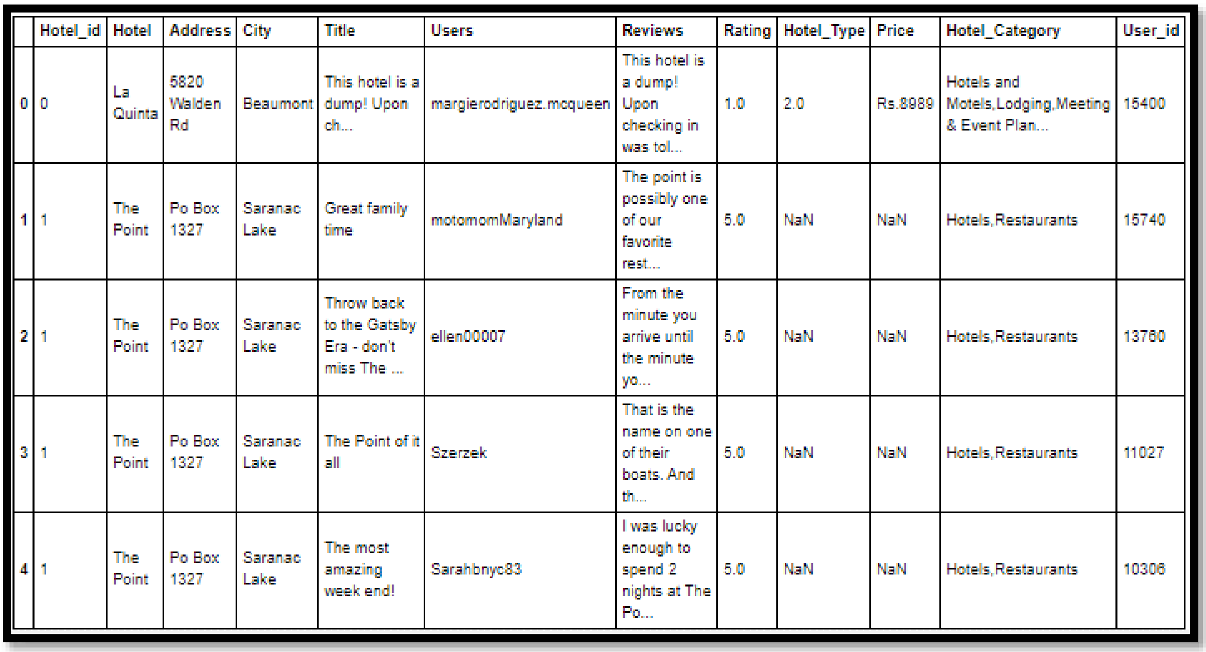
EXPLORATORY DATA ANALYSIS (EDA):

Exploratory studies were carried out after data collection. Via exploratory data review, the following perspectives were explored.

The following are the columns in the dataset:
(['Hotel_id', 'Hotel', 'Address', 'City', 'Title', 'Users', 'Reviews', 'Rating', 'Hotel_Type', 'Price', 'Hotel_Category', 'User_id']

1.Hotel Dataset:

The following is the dataset:



	Hotel_id	Hotel	Address	City	Title	Users	Reviews	Rating	Hotel_Type	Price	Hotel_Category	User_id
0	0	La Quinta	5820 Walden Rd	Beaumont	This hotel is a dump! Upon ch...	margierodriguez.mcqueen	This hotel is a dump! Upon checking in was tol...	1.0	2.0	Rs.8989	Hotels and Motels,Lodging,Meeting & Event Plan...	15400
1	1	The Point	Po Box 1327	Saranac Lake	Great family time	motomomMaryland	The point is possibly one of our favorite rest...	5.0	NaN	NaN	Hotels,Restaurants	15740
2	1	The Point	Po Box 1327	Saranac Lake	Throw back to the Gatsby Era - don't miss The ...	ellen00007	From the minute you arrive until the minute yo...	5.0	NaN	NaN	Hotels,Restaurants	13760
3	1	The Point	Po Box 1327	Saranac Lake	The Point of it all	Szerzek	That is the name on one of their boats. And th...	5.0	NaN	NaN	Hotels,Restaurants	11027
4	1	The Point	Po Box 1327	Saranac Lake	The most amazing week end!	Sarahbnyc83	I was lucky enough to spend 2 nights at The Po...	5.0	NaN	NaN	Hotels,Restaurants	10306

Figure 7.1: Raw Hotel Dataset

The dataset has some Column which was not required for the Sentiment analysis hence they were dropped to get the required dataset.

2.Distribution of Ratings:

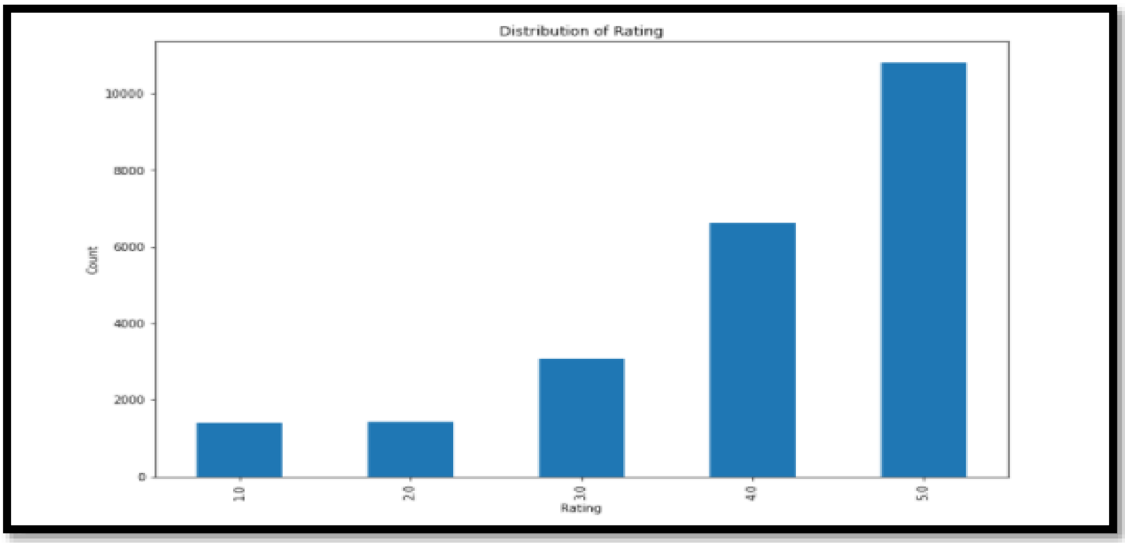


Figure 7.2: Distribution of Rating

Customers have provided the ratings that supported their experience of stay in the hotel. The Rating is distributed from 1 to 5. The maximum and minimum rating provided in the dataset are 5 and 1 respectively. Compared to the other ratings, the number of reviews for grade 5 is higher. Around 74.7% of customers gave 4 & 5 stars for the hotels in which they stayed.

3. Top 20 Most Reviewed Hotels:

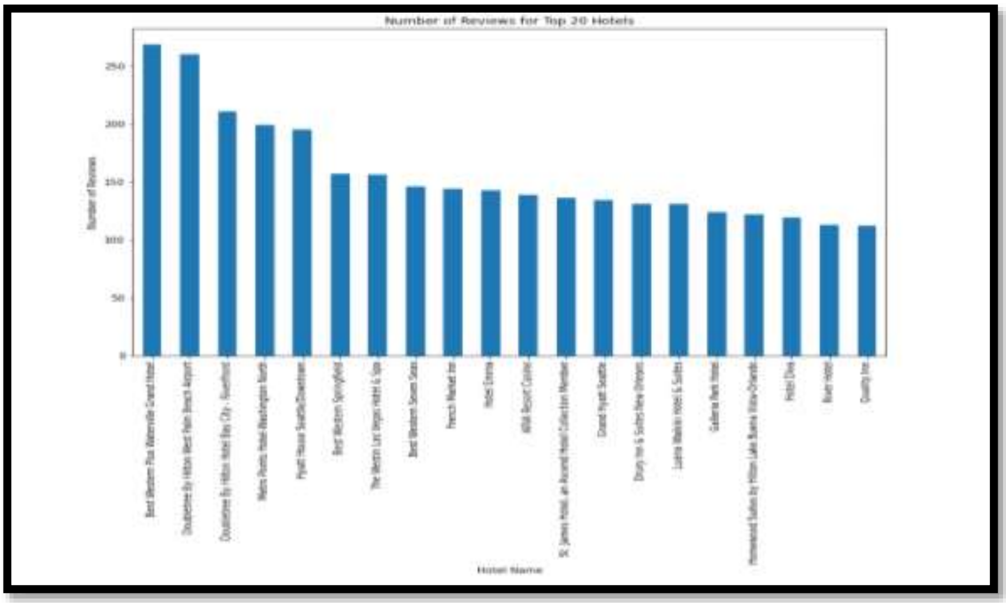


Figure 7.3: Top 20 Most Reviewed Hotels

4. Bottom 20 Most Reviewed Hotels:

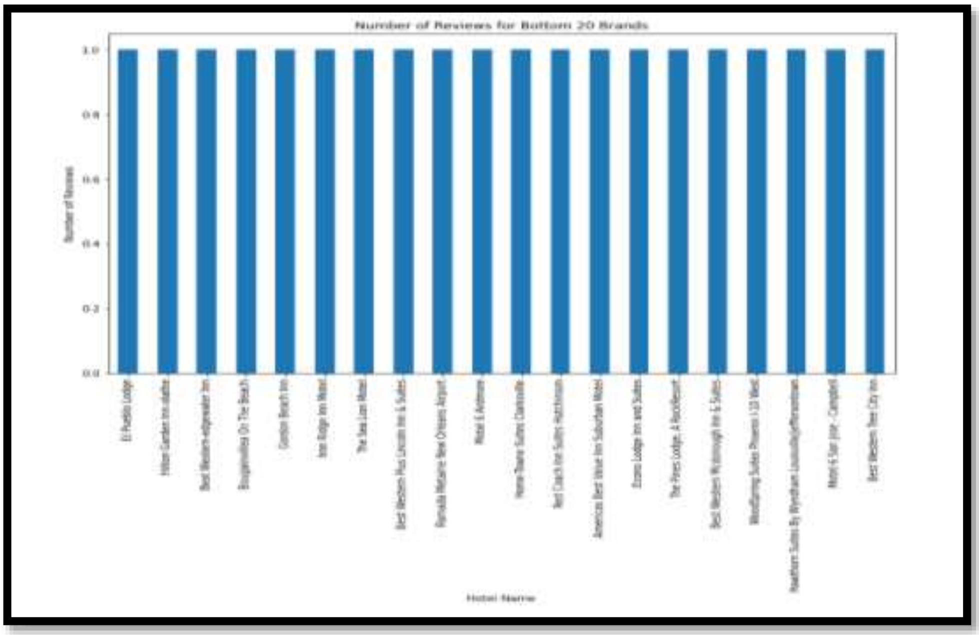


Figure 7.4: Bottom 20 Most Reviewed Hotels

5.Sentiments Distribution:

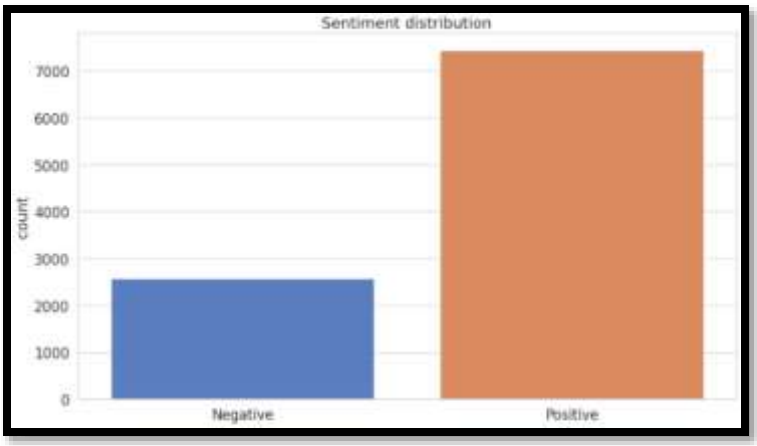


Figure 7.5: Sentiment Distribution

For Sentiment Analysis Rating 4 and 5 are considered as the ‘Positive’ Sentiment and 1,2,3 as ‘Negative’. The sentiments were divided into two categories. They were classified as “Positive” and “Negative” sentiments. The Positive sentiment count is ‘7434’ and the Negative be ‘2568’. This indicates ‘7434’ reviews as the Positive one.

6.Frequent User Name in the Dataset

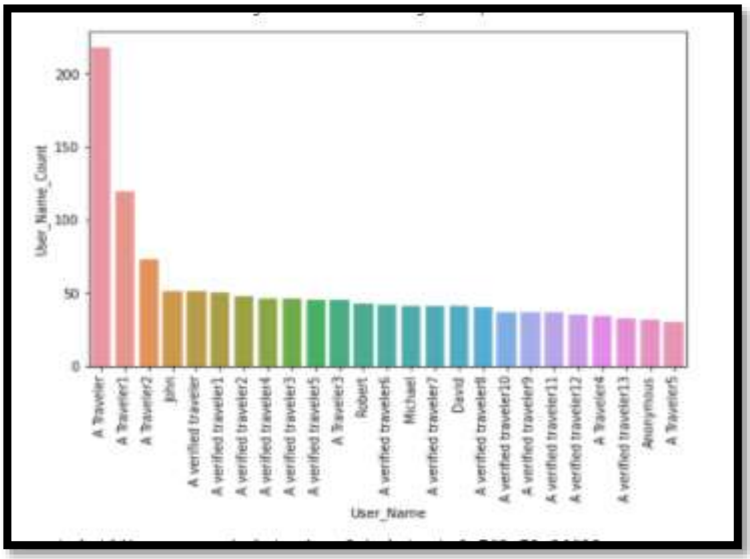


Figure 7.6: Frequent User-Name

As it is evident from the above graph that a traveller is the Frequent User-Name which has appeared in the dataset. We can conclude that A Traveller has visited most of the hotels and is a frequent traveller.

7.Number of hotels by the City

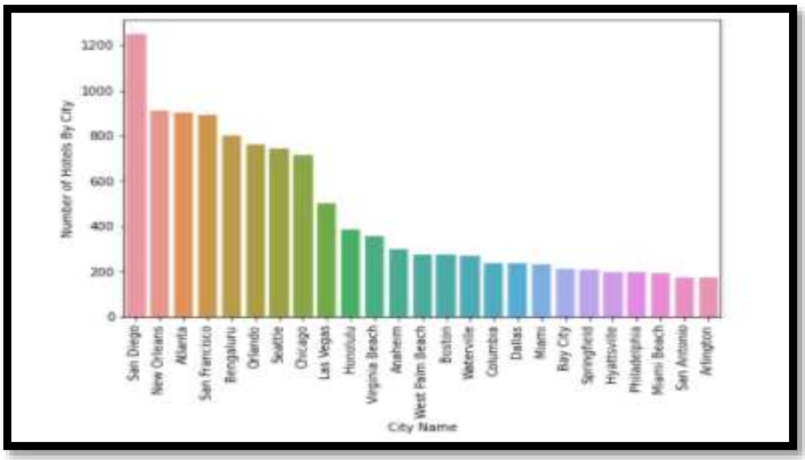


Figure 7.7: Reviews Distribution by City

The above graph shows that San-Diego has the highest number of hotels. The dataset has 3131 unique hotels in various cities.

8.Hotel Category by Hotel

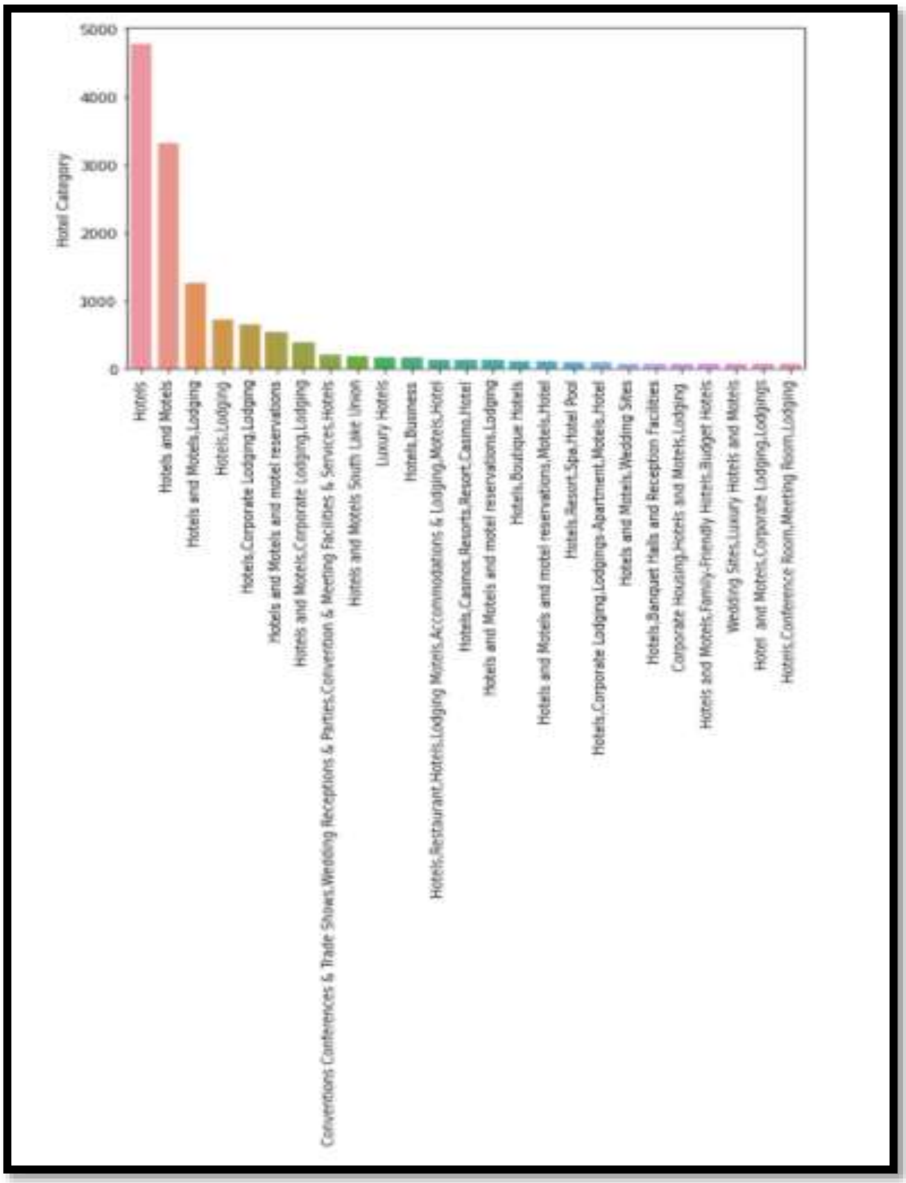


Figure 7.8: Hotel Category Distribution by Hotel

Chapter 8: Data Preparation

- Cleaned the data for missing values that might be within the Hotel data. For numerical data, such as the ratings, averages based on user data were used.
- Some of the hotel's details were missing when web scraped from the TripAdvisor website, so their information had to be entered manually
- Then performed feature selection to remove unnecessary features such as “reviews. source URLs”,” source URLs “,” Websites”,” Postal code”,” reviews.date” and low variance features such as latitude, longitude
- Dropped Duplicate Rows.
- Merged the data collected from different sources into a single format.

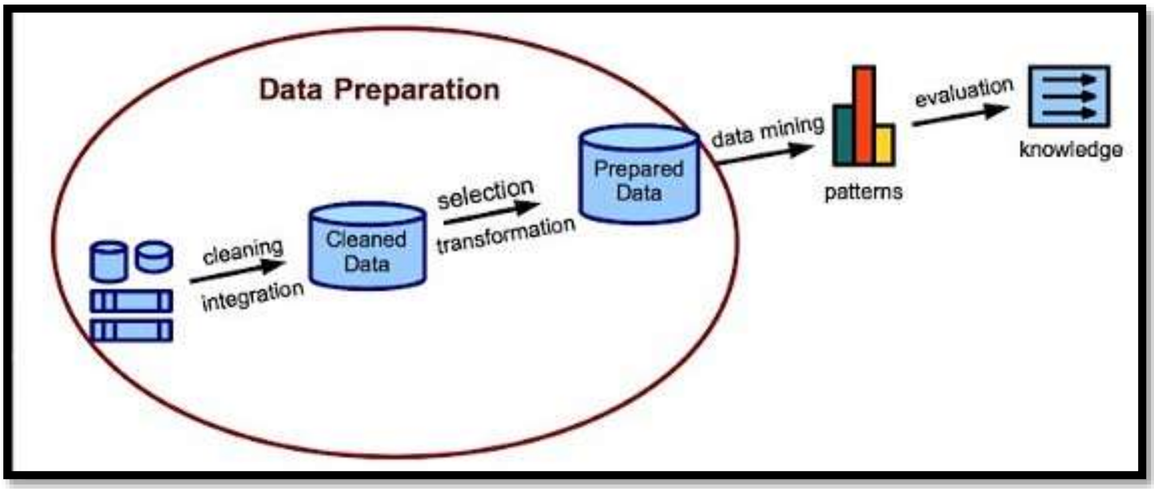
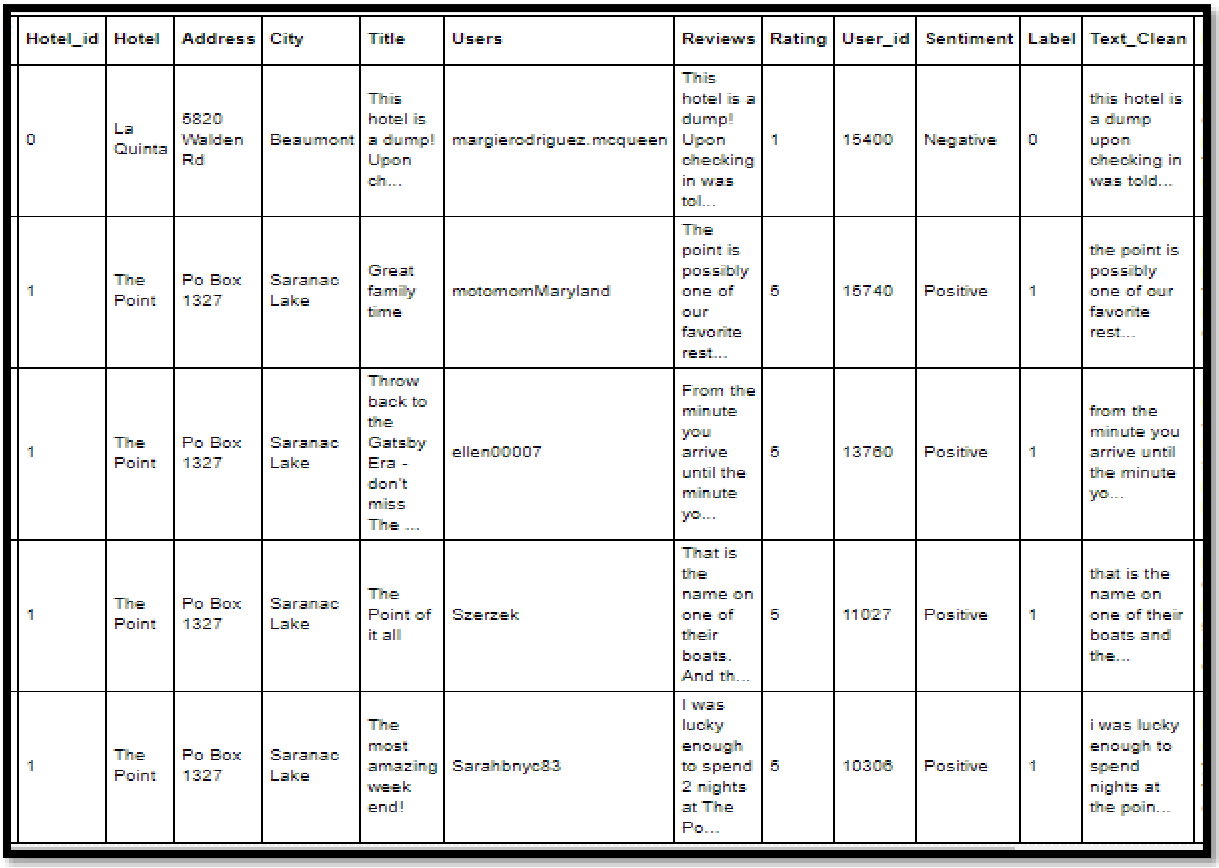


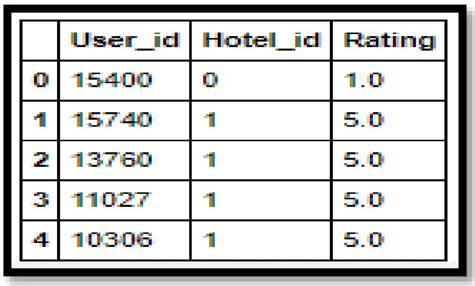
Figure 8.1: Data Pre-Processing



Hotel_id	Hotel	Address	City	Title	Users	Reviews	Rating	User_id	Sentiment	Label	Text_Clean
0	La Quinta	5820 Walden Rd	Beaumont	This hotel is a dump! Upon ch...	margierodriguez.mcqueen	This hotel is a dump! Upon checking in was tol...	1	15400	Negative	0	this hotel is a dump upon checking in was told...
1	The Point	Po Box 1327	Saranac Lake	Great family time	motomomMaryland	The point is possibly one of our favorite rest...	5	15740	Positive	1	the point is possibly one of our favorite rest...
1	The Point	Po Box 1327	Saranac Lake	Throw back to the Gatsby Era - don't miss The ...	ellen00007	From the minute you arrive until the minute yo...	5	13760	Positive	1	from the minute you arrive until the minute yo...
1	The Point	Po Box 1327	Saranac Lake	The Point of it all	Szerzek	That is the name on one of their boats. And th...	5	11027	Positive	1	that is the name on one of their boats and the...
1	The Point	Po Box 1327	Saranac Lake	The most amazing week end!	Sarahbnyc83	I was lucky enough to spend 2 nights at The Po...	5	10306	Positive	1	i was lucky enough to spend nights at the poin...

Figure 8.2: Pre-processed Data

Rating dataset: Every row in the rating data frame has a User_id associated with at least one Hotel_id and a Rating.



	User_id	Hotel_id	Rating
0	15400	0	1.0
1	15740	1	5.0
2	13760	1	5.0
3	11027	1	5.0
4	10306	1	5.0

Figure 8.3: Rating Data

Chapter 9: Data Modeling

Text Processing

Text data contains a lot of noise in the form of ‘symbols’, ‘punctuations’ and ‘stopwords’. Cleaning the text becomes important, not only to make it more understandable but also to recover insights. To convert raw feedback to cleaned review, the subsequent “text pre-processing” is introduced so that it will be easier for us to extract features within the next step and get an insight into the process of choosing a hotel. (SHARMA, 2020)

- Extend contractions
- Lowering the reviews
- Delete digits and digit-containing words
- Deleting punctuations
- Stopwords Removal
- Lemmatization

The “**TextBlob** library” is used to check the polarity of feedback, i.e. how positive or negative a text is. (SHARMA, 2020)

The Hotel dataset consists of Reviews provided by the customers for their stay in the hotel. This binary sentiment classification data set includes a set of 23394 hotel reviews, but we are left with 21770 rows containing reviews after dropping the missing values. The dataset is saved in the “**hotel data final.csv**” file after initial pre-processing. For the study, the first '10006' data points were taken into account in which the missing values were dropped and finally' 10002' data points were taken into account. First, we load the hotel dataset, for 'positive' and 'negative' sentiment respectively, the text reviews are labelled as 1 or 0. "From the dataset, respectively, “lemmatized” and “Mark” were treated as “X (feature)” and “Y (variable)”. The dataset was split into 75% as training and 25% as research.

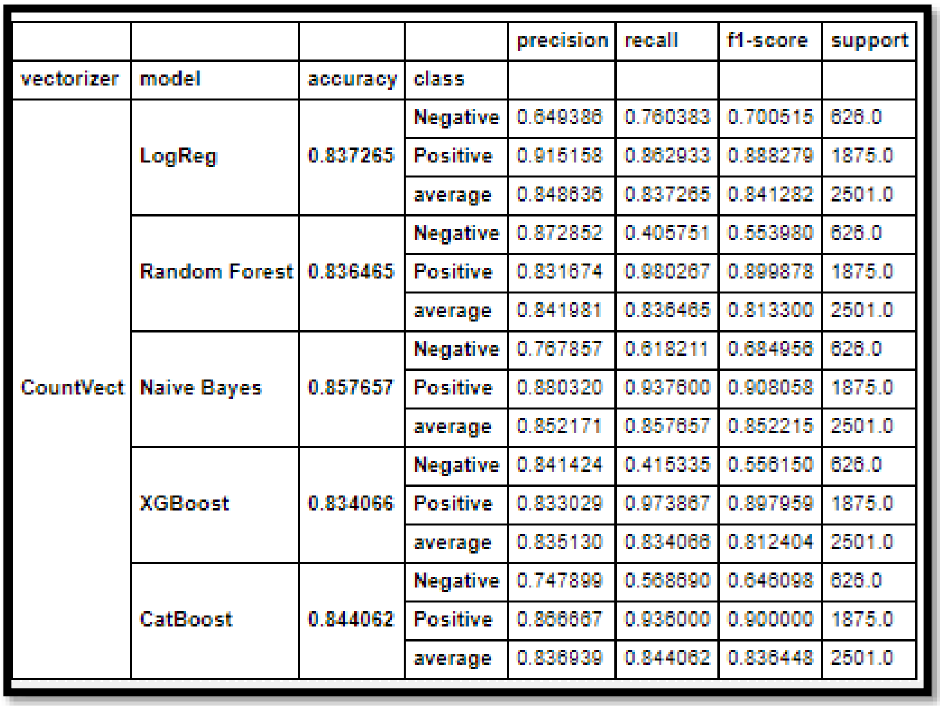
Machine Learning Models: Based on the reviews written by clients who stayed in hotels in different cities, the model needs to forecast sentiment. This is a supervised problem of binary classification. To resolve this problem, Python's “Scikit libraries” were used. The following machine learning algorithms have been introduced.

1. **Logistic Regression**
2. **Naïve Bayes**
3. **Random Forest Classifier**
4. **XGBoost Classifier**
5. **CatBoost Classifier**

Evaluation Metrics: Using a “Confusion Matrix”, which compares the predictions our model makes with the true mark, is a good way to visualize the details. In addition to our evaluation matrix, a confusion matrix was used for this purpose (f1 score).

Modelling: “1-2-3” ratings were classified as 'Negative' and “4-5” ratings were classified as 'Positive, then ' Positive' feelings were listed as ' 1 ' and ' Negative ' as ' 0 '. The threshold for word occurrence using “min df/max df”, “PCA” and “Singular Value Decomposition” was added to “feature selection”. “CountVectorizer”, “TF-IDF”, has been applied to text data for feature engineering to display a collection of “numerical feature vectors”.

Bag of Words Model: Transforming “text documents into vectors” is the core of this model, in such a way that any document is translated into a vector representing the “frequency” of all the distinct words for the particular document within the vector space of the document. The figure below shows that “**Naïve Bayes has the highest precision with 0.8576577**”. (Sarkar, 2016)



				precision	recall	f1-score	support
vectorizer	model	accuracy	class				
CountVect	LogReg	0.837265	Negative	0.649386	0.760383	0.700515	626.0
			Positive	0.915158	0.862933	0.888279	1875.0
			average	0.848636	0.837265	0.841282	2501.0
	Random Forest	0.836465	Negative	0.872852	0.405751	0.553980	626.0
			Positive	0.831674	0.980267	0.899878	1875.0
			average	0.841981	0.836465	0.813300	2501.0
	Naive Bayes	0.857657	Negative	0.767857	0.618211	0.684956	626.0
			Positive	0.880320	0.937600	0.908058	1875.0
			average	0.852171	0.857657	0.852215	2501.0
	XGBoost	0.834066	Negative	0.841424	0.415335	0.556150	626.0
			Positive	0.833029	0.973867	0.897959	1875.0
			average	0.835130	0.834066	0.812404	2501.0
	CatBoost	0.844062	Negative	0.747899	0.568690	0.646098	626.0
			Positive	0.866667	0.936000	0.900000	1875.0
			average	0.836939	0.844062	0.836448	2501.0

Figure 9.1: Comparison Matrix of Models using Count Vectorizer

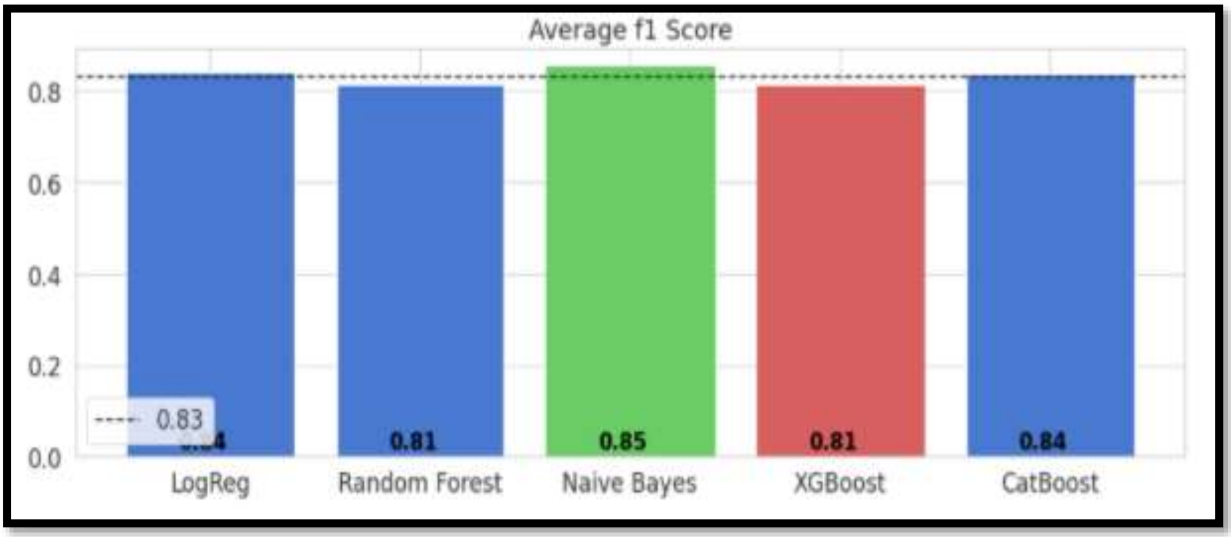
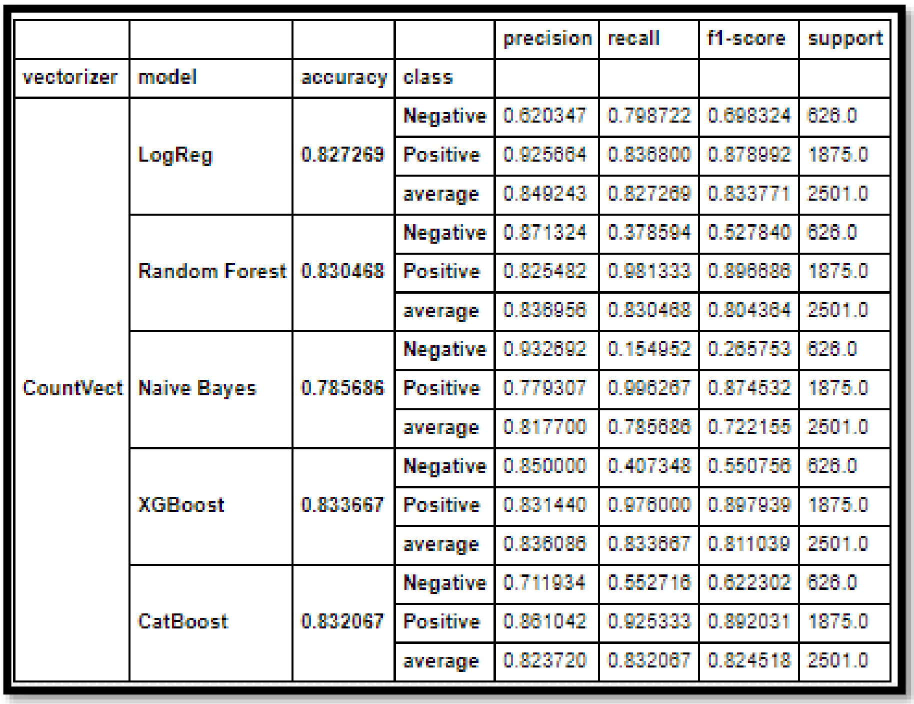


Figure 9.2: Average F1 Score using Count Vectorizer

TF-IDF Model: "TF-IDF" weights terms in our dataset by how rare they are, discounting words that are too common and only boosting the noise. By assigning them lower weights, "TF-IDF" works by penalizing these common words while giving priority to words that occur during a subset of a particular text. The figure below shows that "**CatBoost with 0.832067 has the highest accuracy"**.



				precision	recall	f1-score	support
vectorizer	model	accuracy	class				
CountVect	LogReg	0.827269	Negative	0.620347	0.798722	0.698324	626.0
			Positive	0.925664	0.836800	0.878992	1875.0
			average	0.849243	0.827269	0.833771	2501.0
	Random Forest	0.830468	Negative	0.871324	0.378594	0.527840	626.0
			Positive	0.825482	0.981333	0.896686	1875.0
			average	0.836956	0.830468	0.804364	2501.0
	Naive Bayes	0.785686	Negative	0.932692	0.154952	0.265753	626.0
			Positive	0.779307	0.996267	0.874532	1875.0
			average	0.817700	0.785686	0.722155	2501.0
	XGBoost	0.833667	Negative	0.850000	0.407348	0.550756	626.0
			Positive	0.831440	0.976000	0.897939	1875.0
			average	0.836086	0.833667	0.811039	2501.0
	CatBoost	0.832067	Negative	0.711934	0.552716	0.622302	626.0
			Positive	0.861042	0.925333	0.892031	1875.0
			average	0.823720	0.832067	0.824518	2501.0

Figure 9.3: Comparison Matrix of Models using TF-IDF Vectorizer

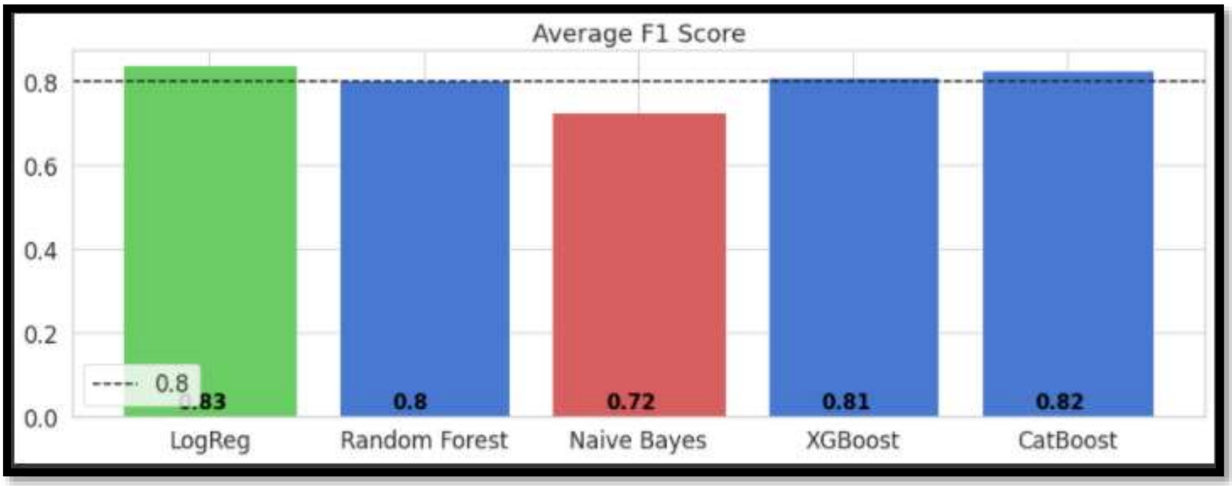
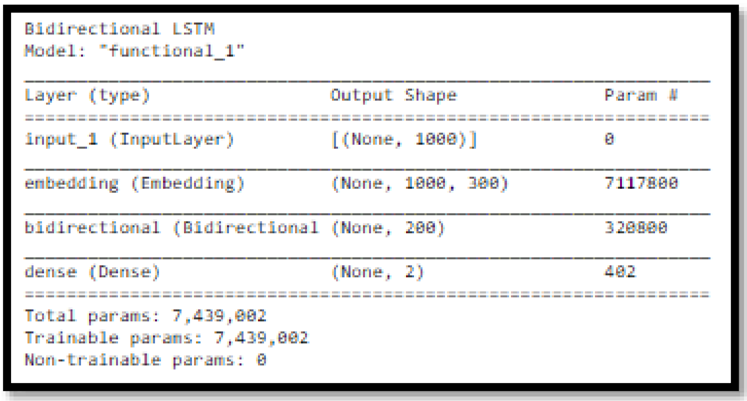


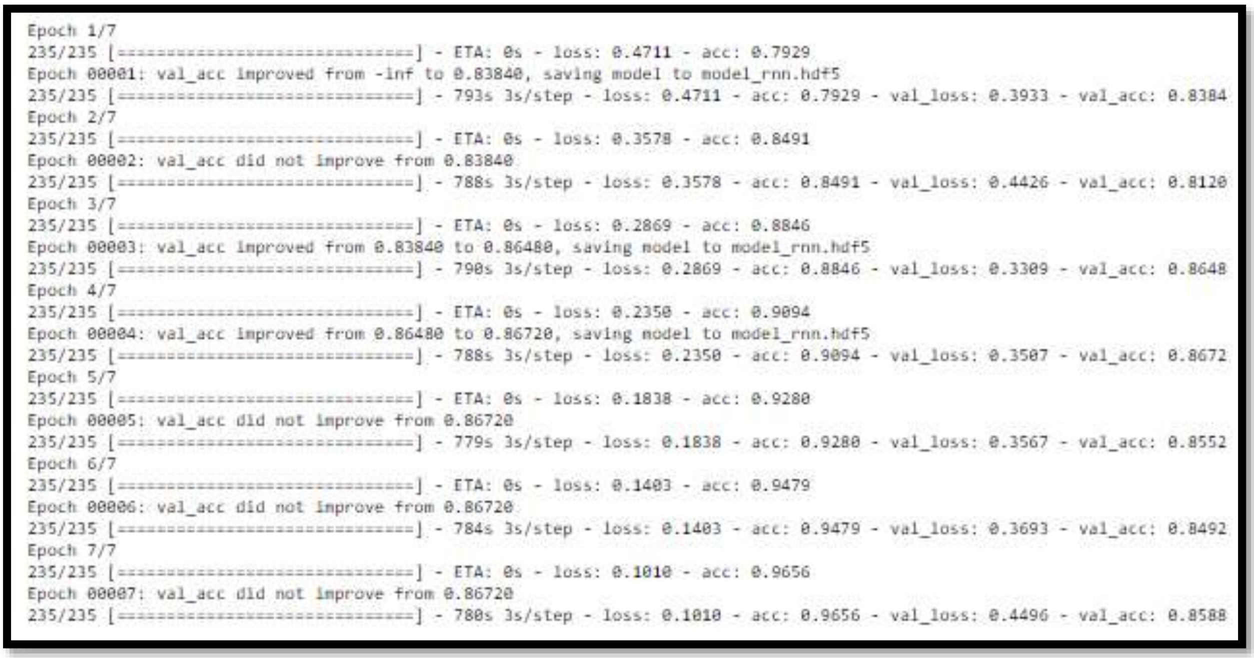
Figure 9.4: Average F1 Score using TF-IDF Vectorizer

LSTM Model for Classification of Sentiment: For evaluating sequences, “RNN’s” are especially useful so that “the hidden layers will learn about earlier parts of the sequence from previous runs of the neural network”. As the text is a series of words, an automatic option to solve text-related problems is a recurrent neural network. Here “LSTM (Long Short-Term Memory Network)” is chosen to solve the sentiment classification, a variant of “RNN”. (Malik, 2019)(Chatterjee, n.d.)

A summary of the model:



```
Bidirectional LSTM
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 1000)]            0
_________________________________________________________________
embedding (Embedding)        (None, 1000, 300)         7117800
_________________________________________________________________
bidirectional (Bidirectional (None, 200)               320800
_________________________________________________________________
dense (Dense)                (None, 2)                 402
=================================================================
Total params: 7,439,002
Trainable params: 7,439,002
Non-trainable params: 0
```



```
Epoch 1/7
235/235 [==============================] - ETA: 0s - loss: 0.4711 - acc: 0.7929
Epoch 00001: val_acc improved from -inf to 0.83840, saving model to model_rnn.hdf5
235/235 [==============================] - 793s 3s/step - loss: 0.4711 - acc: 0.7929 - val_loss: 0.3933 - val_acc: 0.8384
Epoch 2/7
235/235 [==============================] - ETA: 0s - loss: 0.3578 - acc: 0.8491
Epoch 00002: val_acc did not improve from 0.83840
235/235 [==============================] - 788s 3s/step - loss: 0.3578 - acc: 0.8491 - val_loss: 0.4426 - val_acc: 0.8120
Epoch 3/7
235/235 [==============================] - ETA: 0s - loss: 0.2869 - acc: 0.8846
Epoch 00003: val_acc improved from 0.83840 to 0.86480, saving model to model_rnn.hdf5
235/235 [==============================] - 790s 3s/step - loss: 0.2869 - acc: 0.8846 - val_loss: 0.3309 - val_acc: 0.8648
Epoch 4/7
235/235 [==============================] - ETA: 0s - loss: 0.2350 - acc: 0.9094
Epoch 00004: val_acc improved from 0.86480 to 0.86720, saving model to model_rnn.hdf5
235/235 [==============================] - 788s 3s/step - loss: 0.2350 - acc: 0.9094 - val_loss: 0.3507 - val_acc: 0.8672
Epoch 5/7
235/235 [==============================] - ETA: 0s - loss: 0.1838 - acc: 0.9280
Epoch 00005: val_acc did not improve from 0.86720
235/235 [==============================] - 779s 3s/step - loss: 0.1838 - acc: 0.9280 - val_loss: 0.3567 - val_acc: 0.8552
Epoch 6/7
235/235 [==============================] - ETA: 0s - loss: 0.1403 - acc: 0.9479
Epoch 00006: val_acc did not improve from 0.86720
235/235 [==============================] - 784s 3s/step - loss: 0.1403 - acc: 0.9479 - val_loss: 0.3693 - val_acc: 0.8492
Epoch 7/7
235/235 [==============================] - ETA: 0s - loss: 0.1010 - acc: 0.9656
Epoch 00007: val_acc did not improve from 0.86720
235/235 [==============================] - 780s 3s/step - loss: 0.1010 - acc: 0.9656 - val_loss: 0.4496 - val_acc: 0.8588
```

We will see that we get a test accuracy of 85.88 % once we execute the above script. Our precision in training was 96.56 %. For training and test sets, the plot for the loss and accuracy differences is shown below:

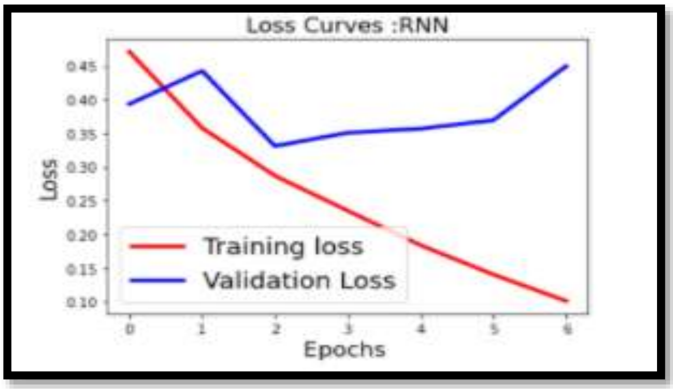


Figure 9.5: RNN Loss Curve

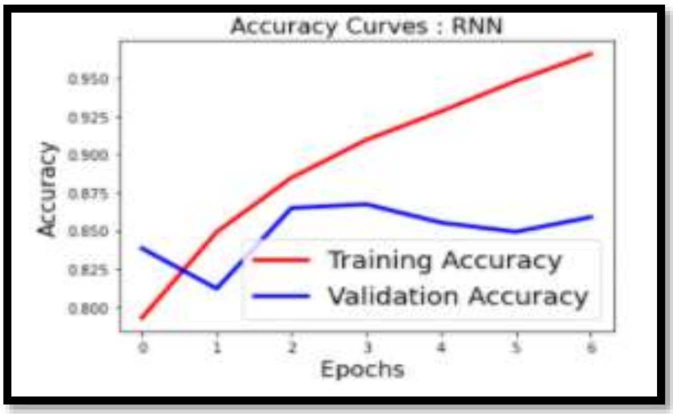
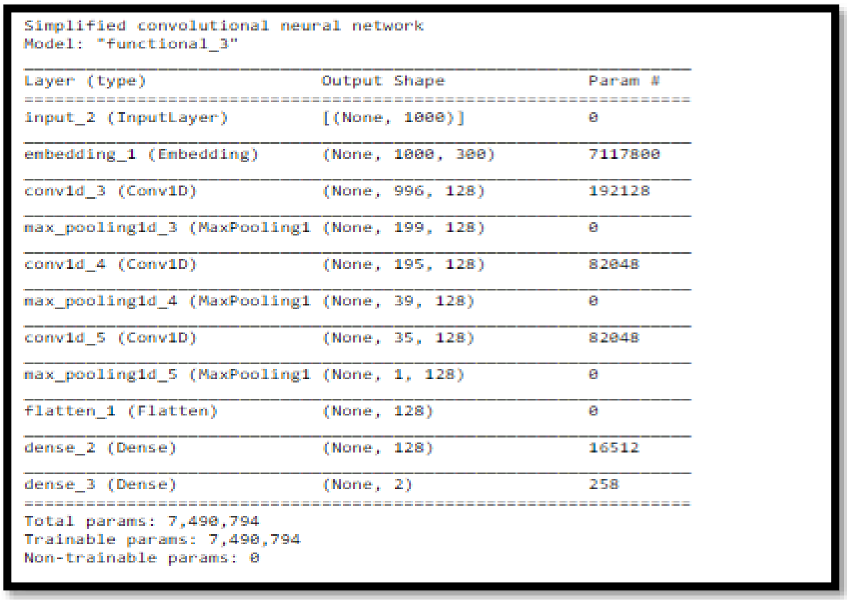


Figure 9.6: RNN Accuracy Curve

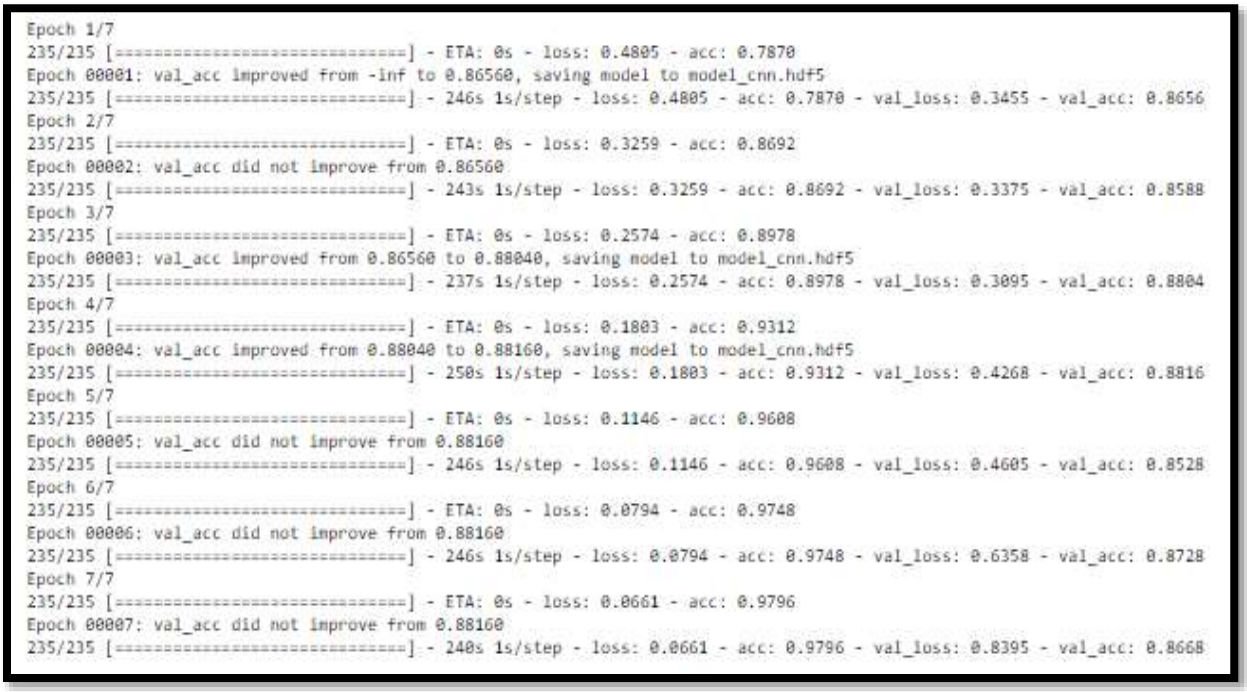
CNN Model for Classification of Sentiment: They are also called “**convnets**” and are the promising advances in the current years of machine-learning. Multiple filters shape the convolutionary layer that can detect unique characteristics that slide across the image. (Malik, 2019)

With 128 features or kernels, we built a one-dimensional convolutional layer, the activation function used is “Sigmoid”. To minimize “feature size”, a global max-pooling layer is added. Finally, we add a dense layer, with the activation “sigmoid”. (Janakiev, 2018)(Malik, 2019)

A summary of the model:



```
Simplified convolutional neural network
Model: "functional_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 1000)]            0
_________________________________________________________________
embedding_1 (Embedding)      (None, 1000, 300)         7117800
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 996, 128)          192128
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 199, 128)          0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 195, 128)          82048
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 39, 128)           0
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 35, 128)           82048
_________________________________________________________________
max_pooling1d_5 (MaxPooling1 (None, 1, 128)            0
_________________________________________________________________
flatten_1 (Flatten)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               16512
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 258
=================================================================
Total params: 7,490,794
Trainable params: 7,490,794
Non-trainable params: 0
```



```
Epoch 1/7
235/235 [==============================] - ETA: 0s - loss: 0.4805 - acc: 0.7870
Epoch 00001: val_acc improved from -inf to 0.86560, saving model to model_cnn.hdf5
235/235 [==============================] - 246s 1s/step - loss: 0.4805 - acc: 0.7870 - val_loss: 0.3455 - val_acc: 0.8656
Epoch 2/7
235/235 [==============================] - ETA: 0s - loss: 0.3259 - acc: 0.8692
Epoch 00002: val_acc did not improve from 0.86560
235/235 [==============================] - 243s 1s/step - loss: 0.3259 - acc: 0.8692 - val_loss: 0.3375 - val_acc: 0.8588
Epoch 3/7
235/235 [==============================] - ETA: 0s - loss: 0.2574 - acc: 0.8978
Epoch 00003: val_acc improved from 0.86560 to 0.88040, saving model to model_cnn.hdf5
235/235 [==============================] - 237s 1s/step - loss: 0.2574 - acc: 0.8978 - val_loss: 0.3095 - val_acc: 0.8804
Epoch 4/7
235/235 [==============================] - ETA: 0s - loss: 0.1803 - acc: 0.9312
Epoch 00004: val_acc improved from 0.88040 to 0.88160, saving model to model_cnn.hdf5
235/235 [==============================] - 250s 1s/step - loss: 0.1803 - acc: 0.9312 - val_loss: 0.4268 - val_acc: 0.8816
Epoch 5/7
235/235 [==============================] - ETA: 0s - loss: 0.1146 - acc: 0.9608
Epoch 00005: val_acc did not improve from 0.88160
235/235 [==============================] - 246s 1s/step - loss: 0.1146 - acc: 0.9608 - val_loss: 0.4605 - val_acc: 0.8528
Epoch 6/7
235/235 [==============================] - ETA: 0s - loss: 0.0794 - acc: 0.9748
Epoch 00006: val_acc did not improve from 0.88160
235/235 [==============================] - 246s 1s/step - loss: 0.0794 - acc: 0.9748 - val_loss: 0.6358 - val_acc: 0.8728
Epoch 7/7
235/235 [==============================] - ETA: 0s - loss: 0.0661 - acc: 0.9796
Epoch 00007: val_acc did not improve from 0.88160
235/235 [==============================] - 240s 1s/step - loss: 0.0661 - acc: 0.9796 - val_loss: 0.8395 - val_acc: 0.8668
```

We will see that we get a test accuracy of 86.68 % once we execute the above script. Our precision in training was 97.96 %.

For training and test sets, the plot for the loss and accuracy differences is shown:

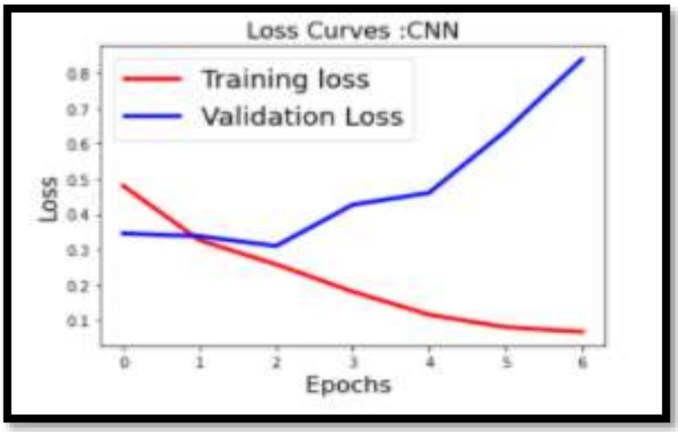


Figure 9.7: CNN Loss Curve

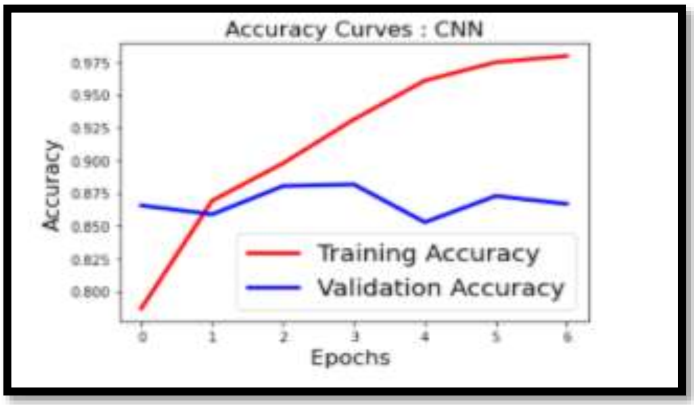


Figure 9.8: CNN Accuracy Curve

Factorization Machine Model for Sentiment Classification: For the implementation of the factorization machine, we'll use a loop-based code as I find it easier to comprehend for the gradient update section. There are different ways to speed up for loop-based code in Python, such as using “**Cython or Numba”**, here we'll be using Numba.

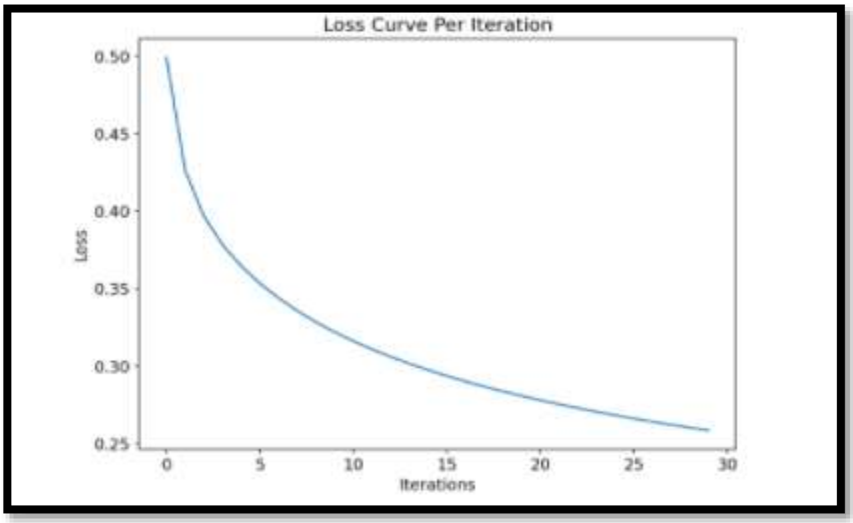


Figure 9.9: Loss Curve per Iterations

The accuracy obtained from the model is 89.37%, which is pretty good as compared to the other models used.

Hotel Recommender System

Recommendation Popularity Dependent: The recommendation method based on popularity works with the pattern. It uses pieces that are in fashion right now. For instance, if any item is normally purchased by a new user, then there are chances that it could mean that item to the user who has just signed up.

The issues with the "popularity-based recommendation system" are that with this approach, personalization is not possible, i.e. even though you know the user's actions, you will not recommend things accordingly.

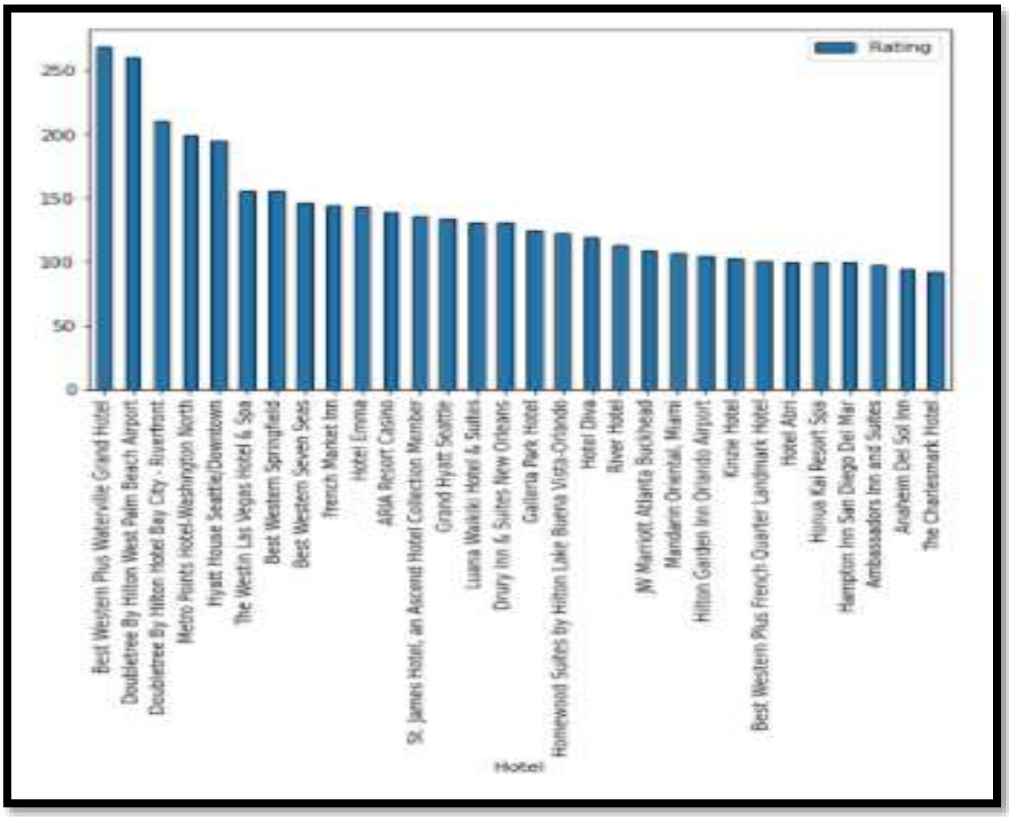
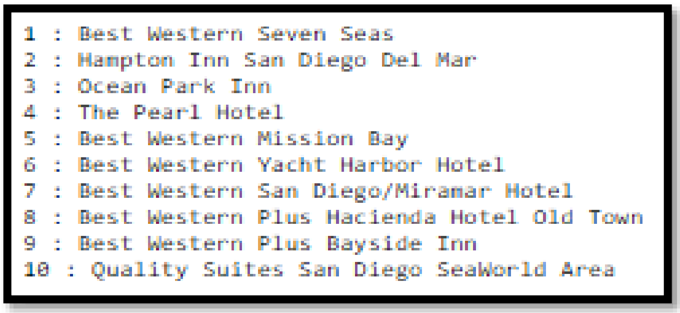


Figure 9.10: Top Popular Hotels to Recommend to the Users

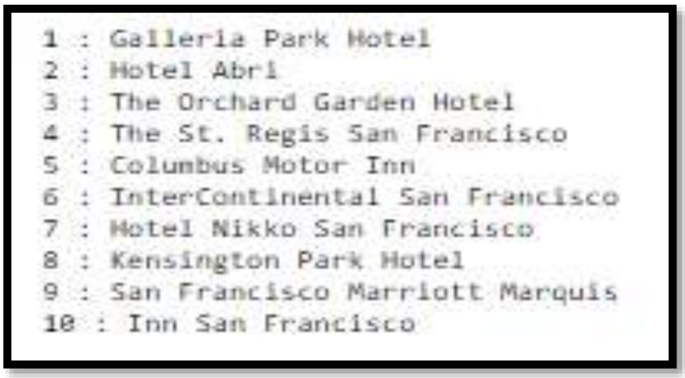
Simple Recommendation: The "Simple Recommender system" provides every user with generalized suggestions based on hotel popularity, ratings, and location. The basic concept behind this suggestion is that more successful and more critically acclaimed hotels are more likely to be liked by users. This "model" gives recommendation to the user based on the location, which would help the user more if they are looking at the best hotels in a particular location.

'San Diego' has the best 10 hotels to recommend:



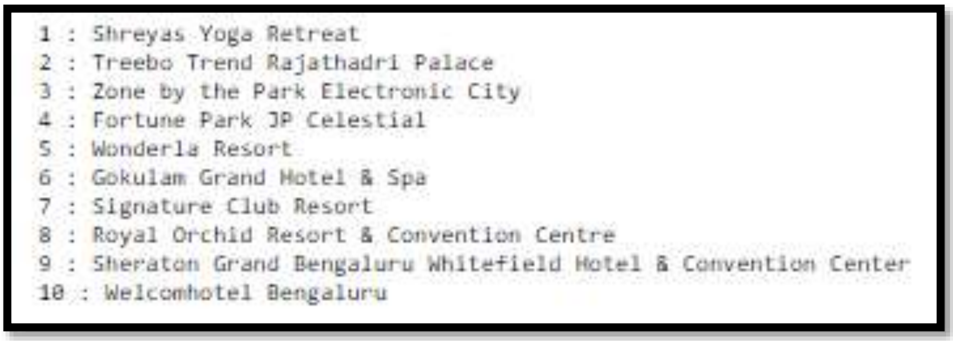
```
1 : Best Western Seven Seas
2 : Hampton Inn San Diego Del Mar
3 : Ocean Park Inn
4 : The Pearl Hotel
5 : Best Western Mission Bay
6 : Best Western Yacht Harbor Hotel
7 : Best Western San Diego/Miramar Hotel
8 : Best Western Plus Hacienda Hotel Old Town
9 : Best Western Plus Bayside Inn
10 : Quality Suites San Diego SeaWorld Area
```

'San Francisco' has the best 10 hotels to recommend:



```
1 : Galleria Park Hotel
2 : Hotel Abri
3 : The Orchard Garden Hotel
4 : The St. Regis San Francisco
5 : Columbus Motor Inn
6 : InterContinental San Francisco
7 : Hotel Nikko San Francisco
8 : Kensington Park Hotel
9 : San Francisco Marriott Marquis
10 : Inn San Francisco
```

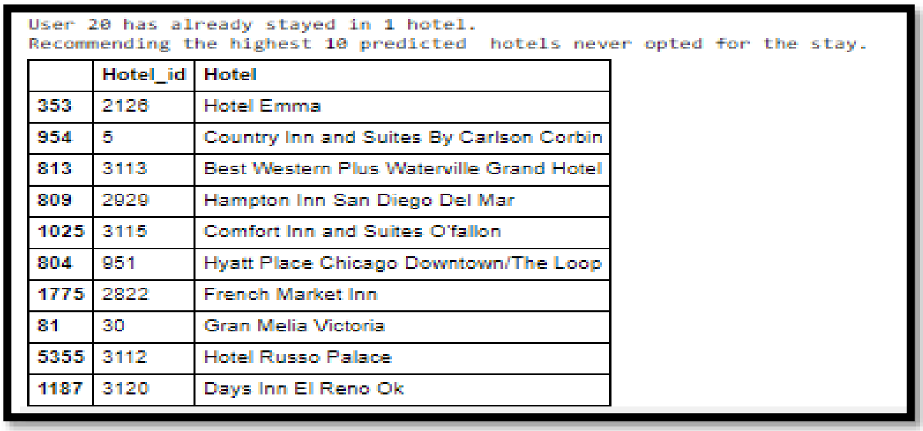
'Bengaluru' has the best 10 hotels to recommend:



```
1 : Shreyas Yoga Retreat
2 : Treebo Trend Rajathadri Palace
3 : Zone by the Park Electronic City
4 : Fortune Park JP Celestial
5 : Wonderla Resort
6 : Gokulam Grand Hotel & Spa
7 : Signature Club Resort
8 : Royal Orchid Resort & Convention Centre
9 : Sheraton Grand Bengaluru Whitefield Hotel & Convention Center
10 : Welcomhotel Bengaluru
```

“Item-Item” based Collaborative Filtering: The resemblance between individual pairs of items was determined by this filtering and a related item that was liked by users in the past were suggested based on that. The weighted number of the 'item-users' ratings was taken. When the number of users is greater than the things being suggested, this “collective filtering” is useful.

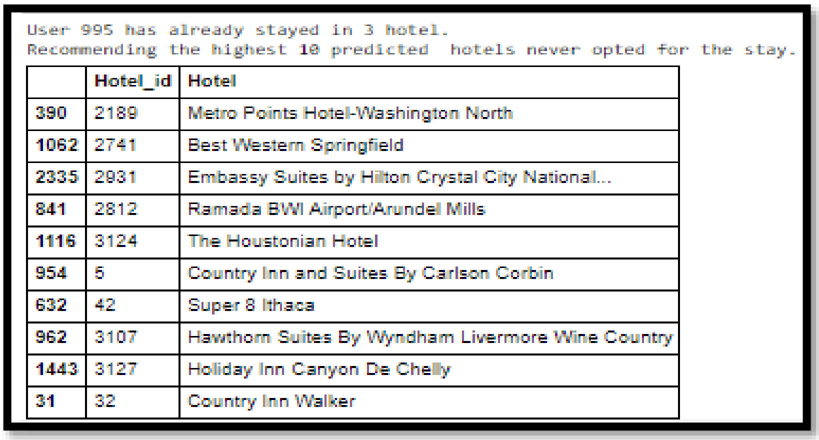
User 20 has already stayed in one hotel, so recommending the other ten hotels to him to stay at.



```
User 20 has already stayed in 1 hotel.
Recommending the highest 10 predicted  hotels never opted for the stay.
```

	Hotel_id	Hotel
353	2126	Hotel Emma
954	5	Country Inn and Suites By Carlson Corbin
813	3113	Best Western Plus Waterville Grand Hotel
809	2929	Hampton Inn San Diego Del Mar
1025	3115	Comfort Inn and Suites O'fallon
804	951	Hyatt Place Chicago Downtown/The Loop
1775	2822	French Market Inn
81	30	Gran Melia Victoria
5355	3112	Hotel Russo Palace
1187	3120	Days Inn El Reno Ok

User 995 has already stayed in one hotel, so recommending the other ten hotels to him to stay at.



```
User 995 has already stayed in 3 hotel.
Recommending the highest 10 predicted  hotels never opted for the stay.
```

	Hotel_id	Hotel
390	2189	Metro Points Hotel-Washington North
1062	2741	Best Western Springfield
2335	2931	Embassy Suites by Hilton Crystal City National...
841	2812	Ramada BWI Airport/Arundel Mills
1116	3124	The Houstonian Hotel
954	5	Country Inn and Suites By Carlson Corbin
632	42	Super 8 Ithaca
962	3107	Hawthorn Suites By Wyndham Livermore Wine Country
1443	3127	Holiday Inn Canyon De Chelly
31	32	Country Inn Walker

Matrix Factorization-based algorithms:

To represent a broad rating matrix, "Matrix Factorization" finds two rectangular matrices with smaller dimensions (RM). These variables preserve the "rating matrix (RM)" dependencies' and properties. A "user matrix (UM)" is a matrix in which users are represented by rows and latent factors are represented by k columns. The alternative matrix is the "items matrix (IM)" where latent k factors are represented by rows, and the objects are represented by columns. (THANDAPANI, 2019). If each customer can express their taste values as a vector and express each item as a vector of what tastes they represent at the same time. We can also make a suggestion very readily. This also allows us to find links between users who have similar interests but have no unique things in common.

SVD: To achieve the results, I have used the Surprise library which has several powerful algorithms like "Singular Value Decomposition (SVD)", "Non-negative Matrix Factorization (NMF)", "K Nearest Neighbour (KNN)", and CoClusternig to minimise RMSE (Root Mean Square Error) and give recommendations.

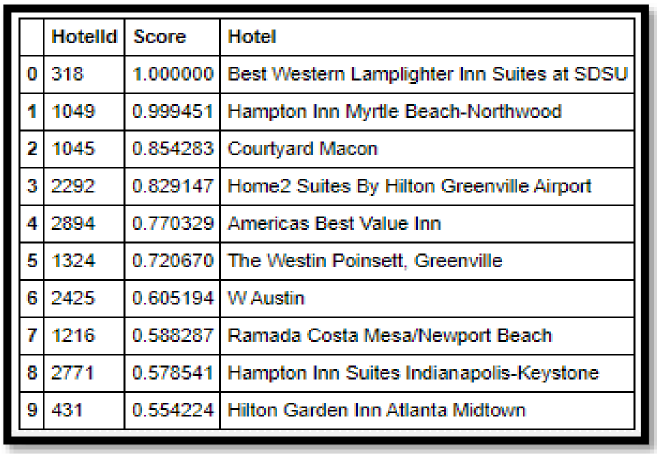
	SVD	**NMF**	**SlopeOne**	**KNN**	**CoClustering**
RMSE (Test Data)	**1.0761**	**1.2097**	**1.2129**	**1.0872**	**1.1943**

Table 9.1: RMSE for the Test data

From the table, we can see that SVD and KNN provide the lowest RMSE for the Test Data.

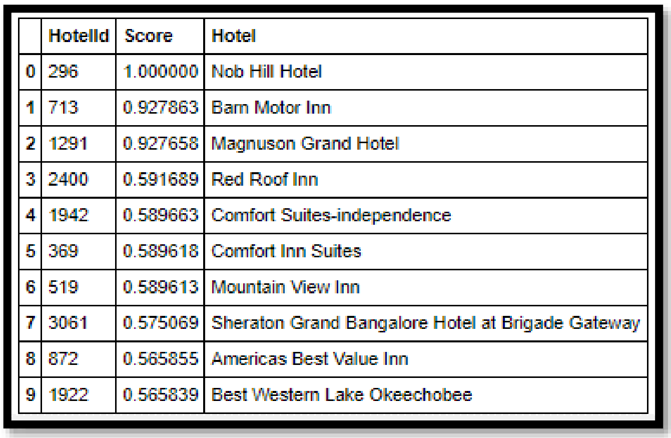
Alternating Least Squares (ALS) Collaborative Filtering: “ALS is an iterative optimization process” in which we try to get closer and closer with every iteration to a factorized representation of our original results. We possess “matrix R of size u x I with our users, products and some sort of feedback data”. Then we should find a way to transform this to a single user matrix and features of hidden size u x f and one with items and features of hidden size f x I. We have weights in U and V for how each user/item relates to every element. We then measure U and V such that their product approximates R as closely as possible: ***R ≈ U x V***. By assigning randomly the U and V values and iteratively using the least-squares, the best approximation of R at what weights can be obtained**.** (Victor, 2017)

Hotel Similar to the Best Western Lamplighter Inn Suites at SDSU (Hotel_id = 318)



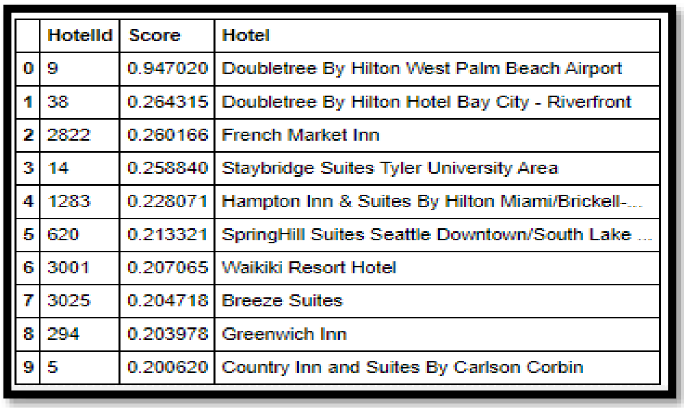
	HotelId	Score	Hotel
0	318	1.000000	Best Western Lamplighter Inn Suites at SDSU
1	1049	0.999451	Hampton Inn Myrtle Beach-Northwood
2	1045	0.854283	Courtyard Macon
3	2292	0.829147	Home2 Suites By Hilton Greenville Airport
4	2894	0.770329	Americas Best Value Inn
5	1324	0.720670	The Westin Poinsett, Greenville
6	2425	0.605194	W Austin
7	1216	0.588287	Ramada Costa Mesa/Newport Beach
8	2771	0.578541	Hampton Inn Suites Indianapolis-Keystone
9	431	0.554224	Hilton Garden Inn Atlanta Midtown

Hotel Similar to the Nob Hill Hotel (Hotel_id = 296)



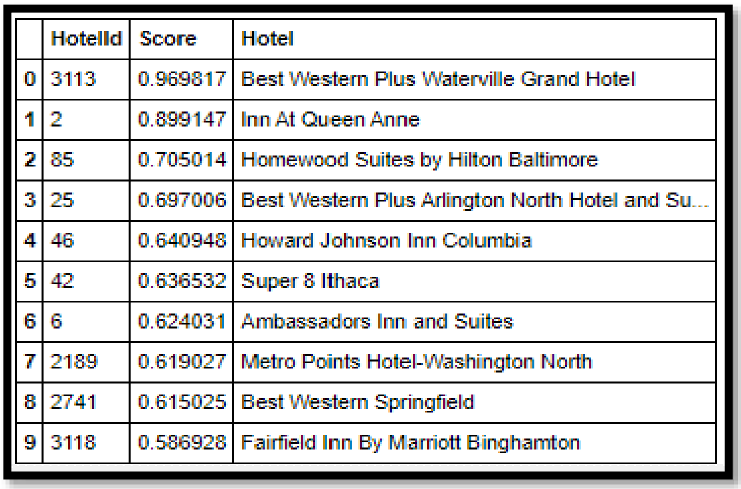
	HotelId	Score	Hotel
0	296	1.000000	Nob Hill Hotel
1	713	0.927863	Barn Motor Inn
2	1291	0.927658	Magnuson Grand Hotel
3	2400	0.591689	Red Roof Inn
4	1942	0.589663	Comfort Suites-independence
5	369	0.589618	Comfort Inn Suites
6	519	0.589613	Mountain View Inn
7	3061	0.575069	Sheraton Grand Bangalore Hotel at Brigade Gateway
8	872	0.565855	Americas Best Value Inn
9	1922	0.565839	Best Western Lake Okeechobee

Hotel recommendations for the user with id= 20



	HotelId	Score	Hotel
0	9	0.947020	Doubletree By Hilton West Palm Beach Airport
1	38	0.264315	Doubletree By Hilton Hotel Bay City - Riverfront
2	2822	0.260166	French Market Inn
3	14	0.258840	Staybridge Suites Tyler University Area
4	1283	0.228071	Hampton Inn & Suites By Hilton Miami/Brickell-...
5	620	0.213321	SpringHill Suites Seattle Downtown/South Lake ...
6	3001	0.207065	Waikiki Resort Hotel
7	3025	0.204718	Breeze Suites
8	294	0.203978	Greenwich Inn
9	5	0.200620	Country Inn and Suites By Carlson Corbin

Hotel recommendations for the user with id= 995



	HotelId	Score	Hotel
0	3113	0.969817	Best Western Plus Waterville Grand Hotel
1	2	0.899147	Inn At Queen Anne
2	85	0.705014	Homewood Suites by Hilton Baltimore
3	25	0.697006	Best Western Plus Arlington North Hotel and Su...
4	46	0.640948	Howard Johnson Inn Columbia
5	42	0.636532	Super 8 Ithaca
6	6	0.624031	Ambassadors Inn and Suites
7	2189	0.619027	Metro Points Hotel-Washington North
8	2741	0.615025	Best Western Springfield
9	3118	0.586928	Fairfield Inn By Marriott Binghamton

The Restricted Boltzmann Machine model:

A restricted Boltzmann machine is a two-layered artificial neural network (input layer and hidden layer) that learns a distribution of probability based on a set of inputs. It is stochastic (non-deterministic), which helps solve various problems based on combinations. (Mohammad-Fawaz-Siddiqi, 2020)

There are two layers of neurons in the Restricted Boltzmann Machine model, one of which is what we call a transparent input layer and the other is called a concealed layer. The hidden layer is used to learn features from the data fed through the “input layer”. The input will contain X neurons for our model, where X is our dataset of the number of hotels.

We train the RBM on it after passing the input and make the concealed layer learn its features. These features are then used to recreate the input, that, in our case, would forecast the ratings for hotels that have not seen the input, which is exactly what we will use to recommend hotels. (Saraswat et al., 2020)

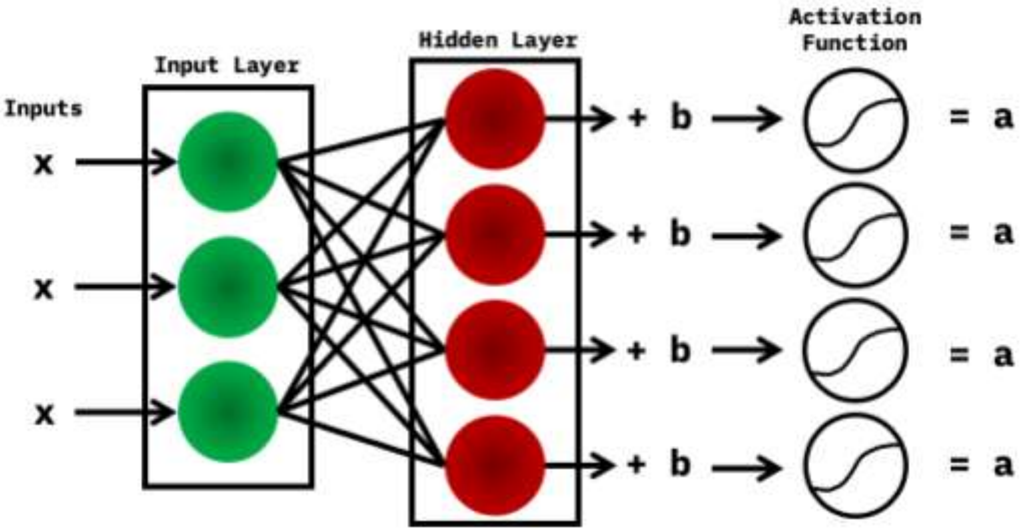


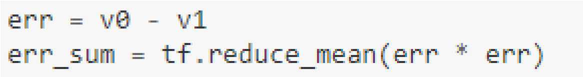
Figure 9.11: Working of Restricted Boltzmann machine

Setting the Model's Parameters:

Here, the RBM Model is developed using TensorFlow. We then move on to developing and setting their activation functions for the visible and hidden layer units. In this scenario, we are going to use "**tf. Sigmoid**" and "**tf. Relu**" as a nonlinear activation.

The selected learning rate =1.0

Now, setting the "Error function", which will be the "Mean Absolute Error (MAE)" in this case.



```
err = v0 - v1
err_sum = tf.reduce_mean(err * err)
```

"Now we're training the RBM with 15 epochs with 10 batches of size 100 for each epoch".

We print a graph with the epoch error after preparation.

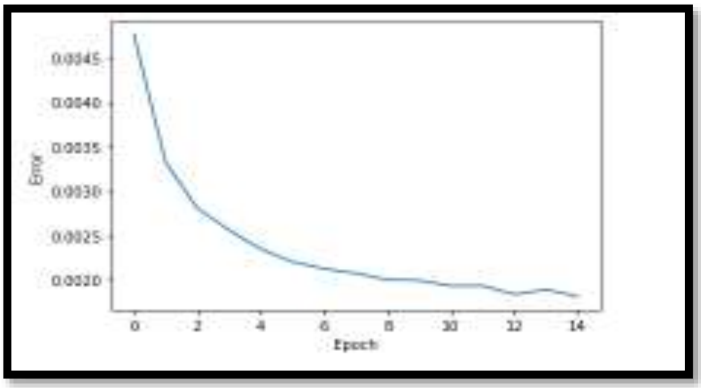
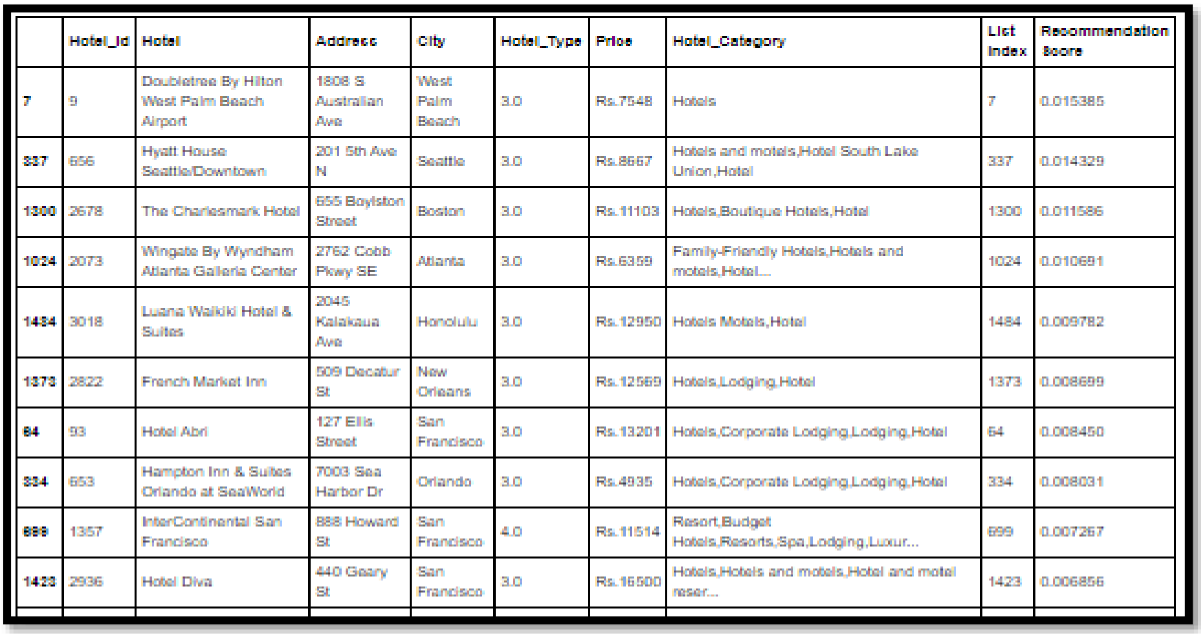


Figure 9.12: Error per Epoch

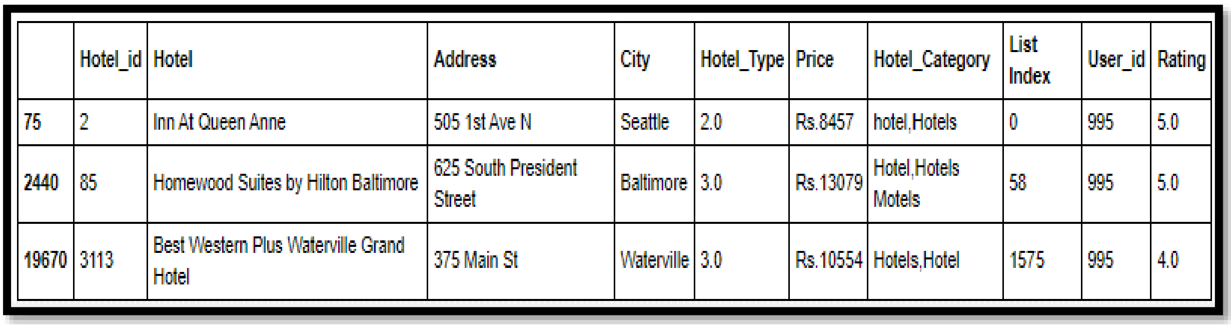
We can now predict hotels that would be liked by an arbitrarily chosen person. This can be achieved by feeding preferences into the "RBM" in the user's visited hotels and then reconstructing the input. The values given to us by the "RBM" will try to estimate the user's preferences for hotels that he has not visited based on the users' preferences that the "RBM" has been trained on.

We can then list our mock user's "**10 most recommended hotels**" by sorting them by their scores provided by our model.



	Hotel_Id	Hotel	Address	City	Hotel_Type	Price	Hotel_Category	List Index	Recommendation Score
7	9	Doubletree By Hilton West Palm Beach Airport	1808 S Australian Ave	West Palm Beach	3.0	Rs.7548	Hotels	7	0.015385
337	656	Hyatt House Seattle/Downtown	201 5th Ave N	Seattle	3.0	Rs.8667	Hotels and motels,Hotel South Lake Union,Hotel	337	0.014329
1300	2678	The Charlesmark Hotel	655 Boylston Street	Boston	3.0	Rs.11103	Hotels,Boutique Hotels,Hotel	1300	0.011586
1024	2073	Wingate By Wyndham Atlanta Galleria Center	2762 Cobb Pkwy SE	Atlanta	3.0	Rs.6359	Family-Friendly Hotels,Hotels and motels,Hotel...	1024	0.010691
1484	3018	Luana Waikiki Hotel & Suites	2045 Kalakaua Ave	Honolulu	3.0	Rs.12950	Hotels Motels,Hotel	1484	0.009782
1373	2822	French Market Inn	509 Decatur St	New Orleans	3.0	Rs.12569	Hotels,Lodging,Hotel	1373	0.008699
64	93	Hotel Abri	127 Ellis Street	San Francisco	3.0	Rs.13201	Hotels,Corporate Lodging,Lodging,Hotel	64	0.008450
334	653	Hampton Inn & Suites Orlando at SeaWorld	7003 Sea Harbor Dr	Orlando	3.0	Rs.4935	Hotels,Corporate Lodging,Lodging,Hotel	334	0.008031
699	1357	InterContinental San Francisco	888 Howard St	San Francisco	4.0	Rs.11514	Resort,Budget Hotels,Resorts,Spa,Lodging,Luxur...	699	0.007267
1423	2936	Hotel Diva	440 Geary St	San Francisco	3.0	Rs.16500	Hotels,Hotels and motels,Hotel and motel reser...	1423	0.006856

Now, we will find all the hotels that our "**mock user**" ('User_id' ==995) has visited/stayed before.



	Hotel_id	Hotel	Address	City	Hotel_Type	Price	Hotel_Category	List Index	User_id	Rating
75	2	Inn At Queen Anne	505 1st Ave N	Seattle	2.0	Rs.8457	hotel,Hotels	0	995	5.0
2440	85	Homewood Suites by Hilton Baltimore	625 South President Street	Baltimore	3.0	Rs.13079	Hotel,Hotels Motels	58	995	5.0
19670	3113	Best Western Plus Waterville Grand Hotel	375 Main St	Waterville	3.0	Rs.10554	Hotels,Hotel	1575	995	4.0

From RBM Model we can see that our mock User_id=995 has mostly stayed in 3-star hotels in the US cities and the model has recommended the user the same category of hotels in different cities of the US.

Chapter 9: Data Evaluation

The sentiment analysis has been performed on the Reviews provided by the customers for their stay in the hotel.

It is seen that the factorization model has performed very well in classifying the sentiments with an accuracy of 89.37%.

Finally, a simple hotel recommendation has been made for the user based on the popularity of the hotel, location, which can help a user find the best hotels in a particular location.

Collaborative filtering is a simple format to recommend based on other user's rating histories. It could be based on matrix Factorization techniques like "Alternating least square (ALS) matrix factorization".

By sorting it by their scores given by our model, the "RBM" model has provided the 10 most recommended hotels for our consumer.

Model Used	Accuracy Achieved
Logistic Regression using CV	0.837625
Naïve Bayes using CV	0.857657
Random Forest using CV	0.836465
XGBoost using CV	0.834066
CATBoost using CV	0.844062
Logistic Regression using "TF-IDF"	0.827269
Naïve Bayes using "TF-IDF"	0.785686
Random Forest using "TF-IDF"	0.830468
XGBoost using "TF-IDF"	0.833667
CATBoost using "TF-IDF"	0.832067
CNN	0.8668
RNN	0.8588
Factorization Machines (FM)	0.893729

Table 9.2: Accuracy of the Models for Sentiment Analysis

Chapter 10: Deployment

1. In the future, the deployment should be done by Creating Web Pages and linking it to Flask Rendering.

2. Using Python, building a full gui with exception handling and form validation.

3. Finally, deploying it to Herokuapp.com and creating an app to get the top 10 hotels.(Kumar, 2020)

Chapter 11: Analysis and Results

The sentiment analysis of the customer reviews has been done and it is seen that the factorization machine model has performed very well and various recommender system models have been implemented to provide recommendations to the users. So, a user can get various new options while searching for a hotel which they were not aware of earlier.

The collaborative Filtering Recommendation framework is focused on the premise that it is possible to use users who are close to me to speculate how much I have would appreciate a new service or product I never encountered before.

User_Id Hotel_Id	SVD	NMF	Slope One	KNN	CoClustering	Actual Rating
User_id=30 **Hotel_id=96**	4.28	4.89	5	4.99	5	5
User_id=3259 **Hotel_id=25**	4.10	3.68	3.60	4.17	2.09	4
User_id=13532 **Hotel_id=9**	4.05	3.93	4	4.01	4.17	4

Table 11.1: The predicted rating of some samples with their actual values

In this work, several recommended systems are implemented as:

1. Simple Recommender system uses the overall User Rating and Rating Averages to build Top Hotel Charts.

2. Collaborative Filtering in a very simple RS that could recommend based on other user's rating histories. It could also be based on matrix factorization techniques like SVD or ALS.

3. From the input data, “RBMs can learn latent factors/variables” (variables that are not accessible directly, but can be deduced from available variables). (A. Sharma, 2018)

Chapter 12: Conclusions and Recommendations for future work

The report highlighted the processes of data wrangling, EDA, data visualization, sentiment analysis using various machine learning techniques and finally the hotel recommendation has been made for the User.

In this project various sentiment analysis technique has been implemented i.e. Logistic Regression, Naïve Bayes, Random Forest, XGBoost, CatBoost etc. and some deep learning techniques too i.e. CNN, RNN and Factorization machines. We have also explored different ways of building a Recommender System. The Factorization machines come out with the highest accuracy of the score. This project illustrates how, in a real-life situation, data science techniques can be used.

Future work other recommendation technique like Hybrid techniques could be implemented. Also, the A/B testing technique could be combined with previous techniques to improve the result. It is necessary to integrate time into a recommendation method because there are sometimes seasonal effects of choice.

Bibliography

Algorithmia. (2018). *Introduction to sentiment analysis: What is sentiment analysis?* Algorithmia. https://algorithmia.com/blog/introduction-sentiment-analysis

Chatterjee, S. (n.d.). *recurrent neural network*. https://developers.google.com/machine-learning/glossary?hl=en#long-short-term-memory-lstm

Cheng, T. (2019). Product recommendation system design. *ACM International Conference Proceeding Series*, 71–74. https://doi.org/10.1145/3357292.3357314

Ge, J., Lv, S., Wr, X., Xvhuv, H. D., & Suhihuhqfh, U. (2019). *Recommendation algorithm based RQ XVHUV ¶ interest preferences and Restricted Boltzmann machine*. 37–41.

Gomathi, R. M., Ajitha, P., Krishna, G. H. S., & Pranay, I. H. (2019). Restaurant recommendation system for user preference and services based on rating and amenities. *ICCIDS 2019 - 2nd International Conference on Computational Intelligence in Data Science, Proceedings*. https://doi.org/10.1109/ICCIDS.2019.8862048

Hong-Xia, W. (2019). An Improved Collaborative Filtering Recommendation Algorithm. *2019 4th IEEE International Conference on Big Data Analytics, ICBDA 2019, 2019*, 431–435. https://doi.org/10.1109/ICBDA.2019.8713205

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), 261–273. https://doi.org/10.1016/j.eij.2015.06.005

Janakiev, N. (2018). *Practical Text Classification With Python and Keras*. Real Python. https://realpython.com/python-keras-text-classification/#what-is-a-word-embedding

Kumar, P. (2020). *Movie Recommendation System | Python & Flask | Web Application | Heroku Deployment*. https://medium.com/analytics-vidhya/movie-recommendation-system-python-flask-web-application-heroku-deployment-7e39492b640c

Malik, U. (2019). *Python for NLP: Movie Sentiment Analysis using Deep Learning in Keras*. Stackabuse. https://stackabuse.com/python-for-nlp-movie-sentiment-analysis-using-deep-learning-in-keras/

Mishra, R. K., Urolagin, S., & Jothi, A. A. J. (2019). A Sentiment analysis-based hotel recommendation using TF-IDF Approach. *Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019*, 811–815. https://doi.org/10.1109/ICCIKE47802.2019.9004385

Mohammad-Fawaz-Siddiqi. (2020). *Build a recommendation engine with a restricted*

Boltzmann machine using TensorFlow. https://developer.ibm.com/technologies/deep-learning/tutorials/build-a-recommendation-engine-with-a-restricted-boltzmann-machine-using-tensorflow/

Nnach, G. A., & Ja, D. (2016). *Recommendations with a Purpose*. https://dl.acm.org/doi/10.1145/2959100.2959186#:~:text=The purpose of recommenders is,in accordance to the estimated

Ramzan, B., Bajwa, I. S., Jamil, N., & Mirza, F. (2019). An intelligent data analysis for hotel recommendation systems using machine learning. *ArXiv*, *2019*.

Saraswat, M., Dubey, A., Naidu, S., Vashisht, R., & Singh, A. (2020). Web-Based Movie Recommender System. *Advances in Intelligent Systems and Computing*. https://doi.org/10.1007/978-981-15-1518-7_24

Sarkar, D. (2016). Text Analytics with Python. *Text Analytics with Python*. https://doi.org/10.1007/978-1-4842-2388-8

Shakirova, E. (2017). Collaborative filtering for music recommender system. *Proceedings of the 2017 IEEE Russia Section Young Researchers in Electrical and Electronic Engineering Conference, ElConRus 2017*, 548–550. https://doi.org/10.1109/EIConRus.2017.7910613

Sharma, A. (2018). *Aditya Sharma Building a Book Recommender System using Restricted Boltzmann Machines*. https://adityashrm21.github.io/Book-Recommender-System-RBM/

SHARMA, A. (2020). *A Beginner's Guide to Exploratory Data Analysis (EDA) on Text Data (Amazon Case Study)*. Analyticsvidhya. https://www.analyticsvidhya.com/blog/2020/04/beginners-guide-exploratory-data-analysis-text-data/

Sharma, Y. (2015). *A Multi Criteria Review-Based Hotel Recommendation System*. 687–691. https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.99

Tavana, M., Mousavi, S. M. H., Mina, H., & Salehian, F. (2020). A dynamic decision support system for evaluating peer-to-peer rental accommodations in the sharing economy. *International Journal of Hospitality Management*, *91*, 102653. https://doi.org/10.1016/j.ijhm.2020.102653

THANDAPANI, S. P. (2019). *Recommendation Systems: Collaborative Filtering using Matrix Factorization — Simplified*. https://medium.com/sfu-cspmp/recommendation-systems-collaborative-filtering-using-matrix-factorization-simplified-2118f4ef2cd3

Tyrolabs. (2019). *Introduction to Recommender Systems in 2019*. Introduction to Recommender Systems in 2019. https://tryolabs.com/blog/introduction-to-recommender-systems/

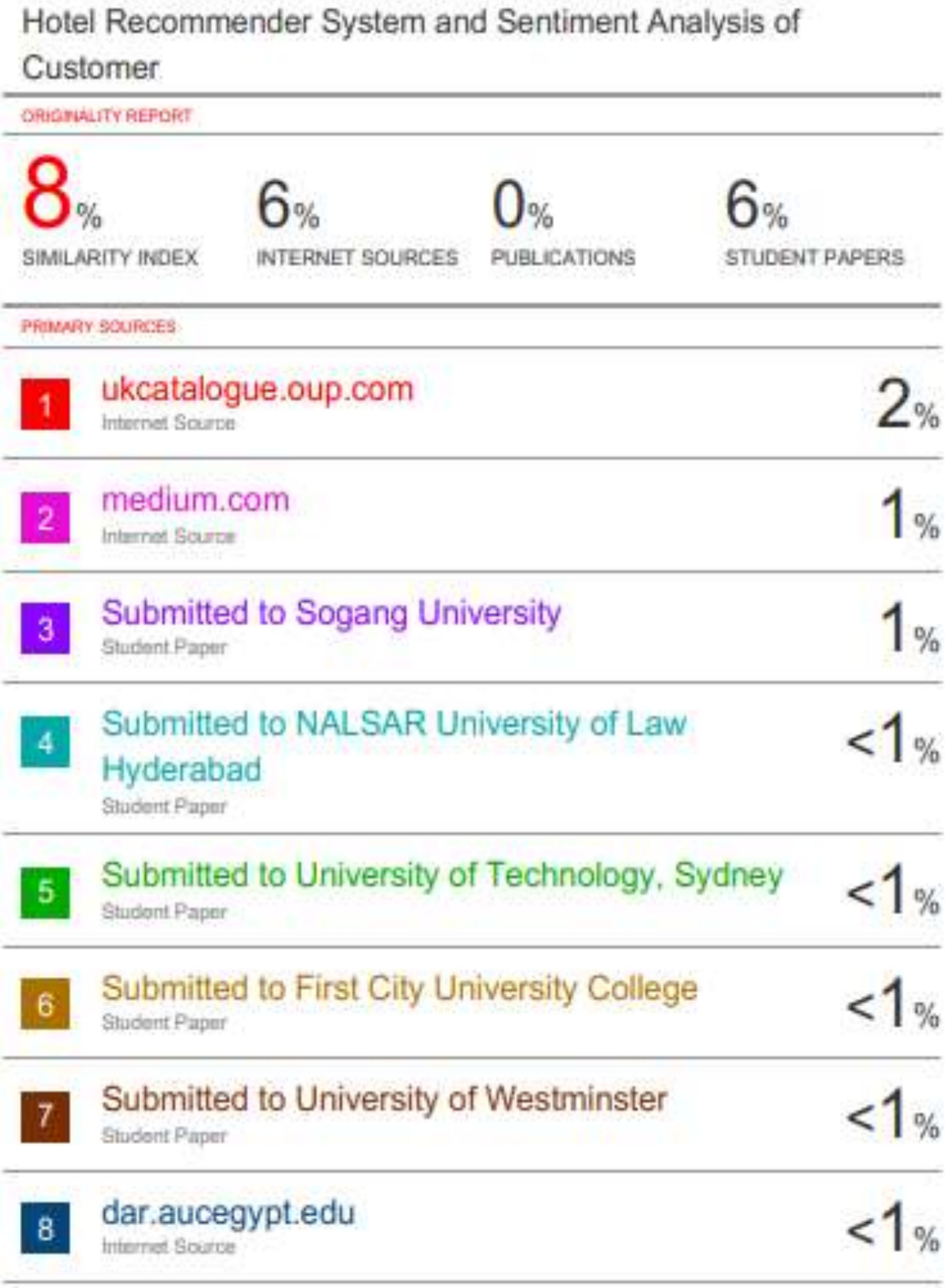
Victor. (2017). *ALS Implicit Collaborative Filtering*. https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe

Appendix

Plagiarism Report[1]

[1] Turnitn report from the University is attached.

Plagiarism Report



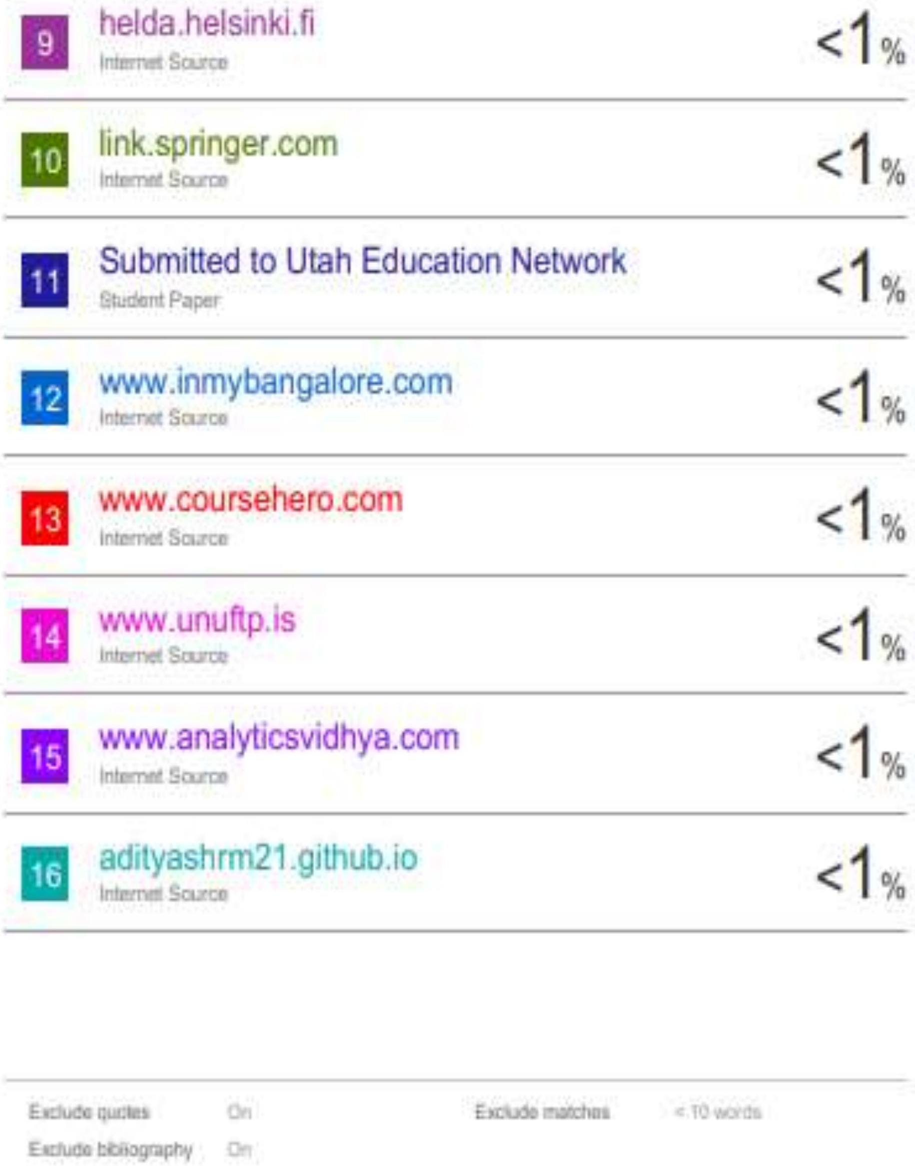
Hotel Recommender System and Sentiment Analysis of Customer

ORIGINALITY REPORT

8%	6%	0%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ukcatalogue.oup.com Internet Source	2%
2	medium.com Internet Source	1%
3	Submitted to Sogang University Student Paper	1%
4	Submitted to NALSAR University of Law Hyderabad Student Paper	<1%
5	Submitted to University of Technology, Sydney Student Paper	<1%
6	Submitted to First City University College Student Paper	<1%
7	Submitted to University of Westminster Student Paper	<1%
8	dar.aucegypt.edu Internet Source	<1%



9	helda.helsinki.fi Internet Source	<1%
10	link.springer.com Internet Source	<1%
11	Submitted to Utah Education Network Student Paper	<1%
12	www.inmybangalore.com Internet Source	<1%
13	www.coursehero.com Internet Source	<1%
14	www.unuftp.is Internet Source	<1%
15	www.analyticsvidhya.com Internet Source	<1%
16	adityashrm21.github.io Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		