***Hotel Recommender System Using Hybrid Filtering***



**Ramya Hunasghatta Manjunatha**

Dissertation submitted in partial fulfilment of the requirements for the degree of

Master of Science in Data Analytics

at Dublin Business School

**Supervisor: Basel Magablah**

August 2019

**Declaration**

I hereby declare that this dissertation that I have submitted “Hotel recommender system using hybrid filtering” to Dublin Business School for the award of MSc. in data analytics under the guidance of supervision of Basel Magablah is solely the result of my own work; collaboration contributions have been acknowledged and are explicitly referenced in the text. This work has not been submitted to any university or college for the award of Degree.

Signature: Ramya Hunasghatta Manjunatha

Student Number: 10388022

Date: 26th August 2019

**Acknowledgement**

I would like to express my deep gratitude to my research supervisor Basel Mageblah from Dublin Business School for continuous support, guidance and enthusiastic encouragement throughout my research. You have always been a tremendous mentor for me offering sincere assistance, motivation and suggestions in my quest for knowledge. You have given me all the freedom to explore my research, while quietly and non-obtrusively ensuring that I focus on research and do not divert from the core of the research. Without his guidance, this thesis would not have been feasible, and I shall eternally be grateful to him for steering me in right direction. Secondly, I would like to thank my family for supporting me throughout this experiment work.

Author

Ramya Hunasghatta Manjunatha

# **Abstract**

Recommender System is a computer system in which the algorithms recommend a set of items or products to the user based on the information provided by the user and the extracted from the data. Recommender Systems are classified into two broad categories – Content-based and collaborative method. These two categories are different in the way the input is taken and processed. Content-based system processes the data inputs taken from the users whereas the collaborative method takes the data from the user/item profiles and then recommends the appropriate item/product. The research gap has been identified as the lack of accuracy in either approaches. This research proposed a hybrid system using natural language processing, sentiment analysis and machine learning algorithms. A dataset of hotels in Berlin has been taken to test the designed system. The proposed system was implemented using Python platform and libraries such as NumPy, SciPy, and joblib. The performance of the results was boosted using XGB, LightGBM and Catboost algorithms. The performance of the system was evaluated, and results were analysed.

*Keywords* – Hotel recommendation system, Recommender Systems, Content-based System,

Collaborative System, Machine Learning

**Table of Contents**

[**Abstract** …4](#_Toc17908384)

[**Chapter 1: Introduction** 9](#_Toc17908385)

[1.1 Business Problem: 9](#_Toc17908386)

[1.2 Scope 10](#_Toc17908387)

[1.3 Research Objectives 10](#_Toc17908388)

[1.4 Research Question 11](#_Toc17908389)

[1.5 Aim 11](#_Toc17908390)

[1.6 Dissertation Blueprint 11](#_Toc17908391)

[**Chapter 2: Literature Review** 13](#_Toc17908392)

[2.1 Introduction: 13](#_Toc17908393)

[2.2 A review of past research on recommendation systems: 13](#_Toc17908394)

[2.2.1 Past research on Hotel Recommender System 15](#_Toc17908395)

[2.2.2 Past research on the Content Based Recommender Systems 18](#_Toc17908396)

[2.2.3 Past research on recommendation systems based on Collaborative Filtering method 19](#_Toc17908397)

[2.2.4 Past research on Knowledge-based Recommender Systems 23](#_Toc17908398)

[2.2.5 Past research on Demographic Systems 24](#_Toc17908399)

[2.2.6 Past research on Critiquing approach 25](#_Toc17908400)

[2.2.7 Past research on Hybrid Systems 25](#_Toc17908401)

[2.3 Algorithms used for recommender systems 25](#_Toc17908402)

[2.4 Challenges of Recommender System 27](#_Toc17908403)

[2.5 Summary 29](#_Toc17908404)

[**Chapter 3: Methodology and Design** 31](#_Toc17908405)

[3.1 Introduction: 31](#_Toc17908406)

[3.2 Dataset 34](#_Toc17908407)

[3.3 Use Case Diagram 35](#_Toc17908408)

[3.4 Summary 36](#_Toc17908409)

[**Chapter 4: Implementation** 37](#_Toc17908410)

[4.1 An insight over a dataset 37](#_Toc17908411)

[4.2 Graphical representation of the data 37](#_Toc17908412)

[4.3 Feature Selection 40](#_Toc17908413)

[4.4 Evaluation of the System 43](#_Toc17908414)

[4.5 Software and Packages: 44](#_Toc17908415)

[4.6 Summary: 45](#_Toc17908416)

[**Chapter 5: Testing And Results** 47](#_Toc17908417)

[5.1 Flow chart of the whole process 47](#_Toc17908418)

[5.2 Results 48](#_Toc17908419)

[1. Listings\_summary.csv 48](#_Toc17908420)

[2. Remove Duplicates 51](#_Toc17908421)

[3. Feature Engineering 51](#_Toc17908422)

[4. Reviews 54](#_Toc17908423)

[5. Results of Collaborative Filtering 54](#_Toc17908424)

[6. Results of SVD Matrix Factorization 56](#_Toc17908425)

7. Results of Content Based Filtering…………...………………………………………58

8. Results of Hybrid Model……………………………………………………………..59

[5.3 System Evaluation: 65](#_Toc17908426)

[5.3.1 Root Mean Squared Error (RMSE) 66](#_Toc17908427)

[5.3.2 Precision 66](#_Toc17908428)

[5.4 Summary 67](#_Toc17908429)

[**CHAPTER 6: Conclusion** 68](#_Toc17908430)

[6.1 Introduction: 68](#_Toc17908431)

[6.2 Conclusion 68](#_Toc17908432)

6.2.1 [Advantages 69](#_Toc17908433)

6.2.2 [Applications 69](#_Toc17908434)

[6.3 Future Research Work 70](#_Toc17908435)

[**References** 71](#_Toc17908436)

**List of Figures**

Figure 2.1. Articles on recommender systems from 2001 to 2010

Figure 2.2. Working of KNN model

Figure 2.3. Working of KNN - result

Figure 3.1. Block diagram of a typical recommender system

Table 3.1. Description of variables of Hotel listings in Berlin

Figure 3.2. Use case diagram of hotel recommendation system

Figure 4.1. Visual Representation of Number of beds, price, square feet, and Number of guests

Figure 4.2. Visual Representation of Extra people, bedrooms, minimum nights, and number of reviews

Figure 4.3. Graph of Number of bedrooms and average price.

Figure 4.4. Graph of Average price Vs. Square feet.

Figure 4.5. Languages and their frequency

Figure 5.1. Flowchart of the Content-based process

Figure 5.2. Flowchart of the whole process

Figure 5.3. Missing Values

Figure 5.4. List of amenities ranked on the basis of number of searches by the users.

Figure 5.5. Score evaluation using SVD matrix factorization

Figure 5.6. Evaluation results using SVD factorization for test and train data

Figure 5.7. Score evaluation using SGD Regressor

Figure 5.8. List of selected features

Figure 5.9. Score evaluation using XGBoost

Figure 5.10. Feature Importance – F Score

Figure 5.11 Score evaluation using LightGBM

Figure 5.12. Feature Importance 1

Figure 5.13 Score evaluation using AdaBoost

Figure 5.14. Feature Importance 2

Figure 5.15. Score evaluation using CatBoost

Figure 5.16. Feature Importance 3

Figure 5.17. Model Evaluation Results

# **Chapter 1: Introduction**

## 1.1 Business Problem:

Recommender systems have become an integral part of several commercial systems in this modern era. Recommender system is a computer system that takes the data of the user activity, processes it, and then predicts the events/items that are likely to be the nearest choice of the user. The major reason recommender system is a need is that the availability of thousands of options for the user to choose from. The users are hit with more options than they can select from. In such a scenario, it is easy if some of the best choices for the user are recommended. This is similar to a scenario when the expert guide offers you tips to choose a better product for your need instead of searching the entire store. The resources available online are enormous, these resources have variety of choices, preferences, and features. However, it is not an easy task to select the right choice with at a glance. The online facilities and products available are not usually customized to the needs of the user. This is when the recommender system comes into picture and helps the user to get the best choice. (Mall, 2019)

The need for the hotel recommendation system using different approaches is that the huge number of datasets and the hotel information available for the user is confusing and it is not customized to the taste of the user. Thus, an advanced recommendation system using content-based and collaborative based system is proposed in this project.

Some of the areas the recommender systems are used include tourism, hotel booking, movies, e-commerce, etc. These systems take the preferences of the user based on the previous use and then recommend one or many products which are the best match for the user’s preferences. A set of parameters may be obtained by the user to shortlist the choices. The system then recommends the best choices for the selected preferences. This method is called content-based recommendation system.

In another approach, a set of common parameters are taken from the users that share interest or location, this data is used by the system to recommend products to the users. This approach assumes that some of the interests of people from similar location (or any other parameter) also share the same interests of choices. This approach has also been effective and is called collaborative recommendation system. In this research, a design of hybrid hotel recommendation system is proposed which uses both content-based and collaborative method for improved accuracy and efficiency.

## 1.2 Scope

The scope of this research investigation is limited to the application of hotel recommendation. The hybrid method uses the data preferences of the user and the collective data available about the users who share common interests to recommend a list of choices.

## 1.3 Research Objectives

The aim of the project is to provide accurate results of hotels for the user using numerous parameters and systems. This research proposes the design and implementation of the hybrid design with the following objectives.

* Conduct and literature survey and analyze the available research on the hotel recommendation systems.
* Design a content-based and collaborative system individually and test the results.
* Design a hybrid mechanism and implement the same.
* Test the results with a dataset and analyze the results.

## 1.4 Research Question

Hospitality industry is one of the most popular commercial industries in the world. Tourism is also on the rise in the recent years because of the affordable air tickets and easy transportation. Millions of people travel every day for leisure and business. Either way, the accommodation of the users is the first of their choices. The number of hotel booking sites of the internet is on the rise due to the demand. There are many search engines exclusively for booking hotels which uses different approaches to search the best hotel from all the available booking sites and then recommend the best choice for the user. Most of these search engines and recommender systems works on content based or collaborative methods. The research question is the lack of hybrid systems with various parameters used in recommendation system. This project solves this problem by proposing a hybrid system with reviews, user inputs, and other data for recommendation.

## 1.5 Aim

The research aims to design and implement a hotel recommendation system using hybrid concept of content-based and collaborative methods.

## 1.6 Dissertation Blueprint

This report is organized in the following order.

**Introduction** – This chapter sets the understanding of the project and its need briefly. This chapter sets the stage for the project.

**Literature Review** – This chapter includes the discussion on the research available on the topic with classification, design and comparison of the research methods. This chapter answers the question *why this project.*

**Design Methodology** – This chapter includes the various methods and approaches used to design the project and how each method is modelled. This chapter answers the question *how.*

**Implementation** – The details of the tools or techniques used along with modelling are discussed in this chapter.

**Findings/Results** – A detailed description of the results obtained and how the results are analysed, is presented in this chapter.

**Conclusion** – This chapter explains why the design results are important, discusses the advantages and applications of the system.

# **Chapter 2: Literature Review**

## 2.1 Introduction:

Literature review is the most important part of any research. The literature review provides a solid foundation for the research in the following manner.

* LR allows the researcher to understand the current state of research on a particular topic.
* LR helps understand the research available on the topic since the inception of the system.
* A survey of the existing research material gives an in-depth knowledge of the research including the issues.
* Literature Survey helps identify the research gap in the specific field of interest.
* Analyzing the existing methods is the foundation of the design of a new method.

## 2.2 A review of past research on recommendation systems:

Recommender systems have been popular for their business value and the accuracy of customization based on the preferences of the user. A recommender system is a subset of the information filtering system that uses different concepts of rating, reviews, and interests/preferences between the user and the item. There are numerous applications of the recommender systems in digital entertainment and other commercial applications. This project is dedicated to the hotel recommendation system. The uniqueness of this project is that it uses both content-based and collaborative based methods in a hybrid way to design an efficient method. This literature review is organized as follows – A list of latest work on collaborative method is described and analysed, then the work on content-based systems is discussed. The research gap of each work is highlighted and the need for an advanced method using the combination of these two methods is presented.

The academic research on the concept of recommender systems has significantly increased in the recent years. It is said that the research on this topic has not reached its maturity because the system is relatively new and still the research methods from other fields is being experimented. A research study on the articles on the recommender systems for a period of 10 years reveals that the concepts of collaborative and content-based systems are the most popular among the researchers. The study of 187 articles by a researcher provided the classification for every year as shown in figure 1 (Park, Kim, Choi & Kim, 2012).

The concepts and the approaches used for the design of the proposed solution provides accurate information on the research available on the topic. This literature review provides the pathway to understand the research gap and answers the question *why this research* for the proposed system.



Figure 2.1. Articles on recommender systems from 2001 to 2010

A classification of the articles on the recommender systems is presented by using the classification framework for two fields - recommendation fields and data mining techniques.

Recommendation fields – The use of recommendation fields as inputs to the recommender systems is a popular method used in the literature. The inputs are given by the user. These input fields include the modelled version of the choices of the users. In case of the hotel recommendation system – the recommendation fields may be (but not limited to) location of the hotel, price of the hotel, facilities such as Air conditioning, hot water, Wi-Fi, etc. These inputs can be provided by the user when he/she searches for the hotel.

These inputs are strong foundation for the next steps of the design. The recommendation system is designed based on the available and mined information regarding the user. The inputs that are given by the user are the first and the best choice of fields for the system. Thus, they are important and need not be mined or processed again for extracting any information. They are direct and have more weight in the design parameter.

Data Mining techniques – There are several other parameters used as the input to the recommender systems apart from the ones provided by the user. These inputs are not available directly. They are mined using different data mining techniques before they are used in the design.

## 2.2.1 Past research on Hotel Recommender System

Bell and Koren, (2007) discussed that the recommender system proposed in their journal is related to the collaborative filtering using collaborative filtering method. This method uses the learning approach in which the relationships between the user and the item acts as the foundation for the system. One of the most popular method is based on the neighbourhood approach. In this approach the preference is based on the similar items which were rated previously. This method is effective but needed to be improved for better results. The prediction accuracy should be improved and the same was performed by the proposed method in this research. There are different steps used in this method. The first step is to remove the effects that are a result of the global/universal patterns. The interpolation accuracy is improved with this step. The interpolation weights are then compared. The difference is that the weights are measured separately and not as combined variables. The optimization problem is overcome by improving the accuracy of the system. This method has been fast compared to the existing methods and has also been proven to be efficient. The number of parameters is less in this method which will result in fewer iterations and less processing time. The results of this method have been experimented using different databases and the solution was found to be effective. This method is the basis of this project in which the design is based on the neighbour approach partially. The research gap is that this method does not include several other parameters of content-based approach. (Bell & Koren, 2007).

Koren (2008) argues that the personalization of the user inputs and recommendations based on accuracy is the result of this research paper. The collaborative filtering method has been quite popular among research professionals and industry experts for various reasons. In this method, the past transactions were analysed, and a unique relationship was established between the user and the products. This relationship is based on numerous parameters which defined the accuracy of the design. The design also works on the models that define the mathematical correlation between any two parameters. There are two subsets of the collaborative method, collaborative filtering and the latent factors. Both these methods are well analysed and experimented in this research. There are innovative methods that use combined approaches in this model. The accuracy has been improved with the use of both implicit and explicit feedback from the clients. A dataset was obtained, and the method was tested before it could be approved. The evaluation parameters are used to assess the efficiency of the system and it was found that the performance has surpassed the expectations and works well for recommendation systems. This method however lacks in the content-based approach because the latent factor and the neighbourhood methods do not offer to solve the issues of content used for recommendations (Koren, 2008).

Hu, Koren and Volinsky (2008) describes the personalization of the recommender systems as the fundamental requirement. However, the lack of feedback on certain systems is causing the issues with accuracy. The proposed method of this paper dealt with the explicit feedback and the preferences of the user and the item relationship. It is important to note that the preferences are mostly designed for the likes of the user but not the dislikes. This model also proposed the method in which the preferences were calculated for the dislikes. The data was categorized into positive and negative aspect which provided with the required relationships and preferences for both likes and dislikes. The factor method works on the feedback of the users as the input to improve the accuracy. This method is also scalable and works well with the linear dataset of a large size. The method was tested for the data of the television shows. The same could be extended for the hotel recommendation system of this project. However, the gap is the same as the previously discussed papers which does not use the content-based method. The collaborative method has its advantages but does not improve the accuracy without the content-based method (Hu, Koren & Volinsky, 2008).

Takacs, Pilaszy, Nemeth and Tikk (2008) recommends that the recommender and rating-based systems are efficient for a large number of applications. This research work proposed a matrix factorization method which was termed hybrid model because it had multiple approaches. The accuracy of the model was highly improved when compared to the previously discussed models. The research work is arranged in the order of advancement in the method of recommender systems in this literature review. The factor models have also been effective when combined with the neighbour-based methods. At this point of this literature review, it is noticeable that the research on this topic has been focusing on the accuracy and different approaches have been used in different combinations. The data of the TV shows is a popular choice among the researchers because of the large amount of data and how different preferences of the user can be modelled for that data. The same approach can be experimented for the hotel recommendation system by modelling the feedback and the neighbour-based systems (Takacs, Pilaszy, Nemeth & Tikk, 2008).

## 2.2.2 Past research on the Content Based Recommender Systems

The working of the content-based recommender system is explained in this subsection. This approach is the first and yet efficient method used by several search and recommender systems in practice. The basic functioning of the recommender system works on the inputs provided by the user. The data provided by the user is used to design and create a profile of the user. This data can also be used to mine hidden patterns from the inputs. The recommendations are then provided based on the inputs from the user and processing. The accuracy of the recommender system is dependent on the number of inputs given by the user. More the inputs, more accurate are the results. There are two concepts extensively used in the content-based systems. They are Term frequency, and inverse document frequency. These terms are the foundations for the design of content-based method. The term frequency and the inverse document frequency are related and are used to check the relationship and frequency of the terms provided by the user. If the term *‘the best hotel with jacuzzi’* is searched. The inputs given by the user should be measured based on each word. The use of *the* has higher frequency compared to *jacuzzi.* However, it is important to note that the weight of the word *jacuzzi* is more because the term is what user specified with respect to the facilities in a hotel.

The content-based systems consider the similarity of the user inputs and the other qualities of the user in terms of behaviour to offer recommendations. On the other hand, the details of the user inputs from his/her past recommendations is also an input for the system. The weightage of these terms is defined in the design of the system. The similarity index is designed in which the previous terms and the relationship between the previous and the current term is compared.

Content-based system works based on features. The available features, be it the inputs taken from the user or the default variables of the system, are used and a model is built. An example can be used to understand the system. In case of a hotel recommendation system, it is possible, based on the data, to derive some of the model features about the users. Young tourists may like the technological features such as Wi-Fi in a hotel whereas elderly people may like leisure amenities such as swimming pool. This understanding is called modelling in content-based system. The parameters of the user and the parameters of the hotel are synchronized before they are modelled. Each parameter is given a set of boundaries to define them.

## 2.2.3 Past research on recommendation systems based on Collaborative Filtering Method

Çano and Morisio (2017) debates that the collaborative system is different from the content-based system. It considers the parameters from similar user and uses it for filtering the results. It is common that the users from similar location or area of work or any parameters share the interests. This method considers this parameter and recommends the results. The behaviour of the users is stored in the form of profile. This data is then used when a user from similar profile searches for any hotel. The collaborative method is a preferential model. If prefers the existing data of the users to profile the results for the current user.

If user A has booked Hotel X and has rated positively. The user B who is in search of a hotel and shares the common interests with user B is recommended the Hotel X. The collaborative system is based on the preferences and not on the inputs entered by the user for search terms. The preferences can be either on the items or on the users. In a different example, if user A has purchased Item X and Y. The recommender system recommends Item Y to the user who purchases Item X. Thus, the preference could be on the user or on the product the system recommends. In the first case where the preferences of the user are taken as input, the system profiles the user behaviour and uses the same to recommend the item for the next user. However, in the second case, the item is profiles and used as preferences in which the user profiling is not performed (Çano & Morisio, 2017).

Rocca (2019) states that the collaborative systems have a major advantage that are in no need of the input from the users. Such system can be used for several applications. The interaction of the users is the major input for the system. The interaction also manages the accuracy of the system. This system takes in all the interactive activities of the user and improves the accuracy of the system. Each time a user interacts in terms of any input of the system, any other changes in the search parameter or any other system variable, the data is stored, and the input can be used for the recommender system. Thus, the data provides more and more inputs to the recommender system. This way, the system becomes more accurate. The past interactions of the user are a great input. However, there is an issue to be addressed in this method. The problem of cold-start occurs. Since the recommender systems works on the user inputs and interactions, if the user is using the system for the first time, there is absolutely no input to the system. Thus, the recommender system will have no results based on the data. At this point, the system is just a search engine which does not use any of the models designed but provides search results only based on the default conditions. This problem is one of the major disadvantages of the recommender system. These issues will always occur because the system is completely based on data. One alternative way to solve this issue is that the system randomly recommends a set of search results with random parameters considered.

A digital user who is displayed a set of hotels will not be satisfied at the first attempt because the search parameters are not set in place. On the other hand, it is also possible that user understands the need for the variables to customize the results. There are other ways this cold-start problem is handled. The users may be recommended a set of results displayed to the most common users. These results are the popular result in the particulars are (if the demographic information is taken) with default conditions on the parameters. However, once the user changes any of the parameters and interacts with the system, the results will change instantly, and a set of customized results are provided. The user profiling is also initiated to understand the improvise the results. (Rocca, 2019).

Two types of collaborative methods are explained for better understanding of the recommender systems – memory based, and model based.

Memory based collaborative method – In this type of method, the concept of latent model is ignored. The algorithms take the input directly from the user. There is no modelling of the data after the user interactions are considered. The representation of the users is directly taken in the form of interactions. The interactions are defined as the relationship between the user inputs and the parameter of the item. This relationship is unique and is customized to the user. This data cannot be tampered or manipulated. The neighbouring method is used in this case. The interaction of the user representing the variable and the product is checked with the neighbours. The mathematical modelling is performed only after the user and the item parameter has been established with a relationship called interaction.

In the memory-based approach, there are different types – user-user and item-item. The user-user method is a typical method used by most of the recommender systems. In this method, the interaction of the user is taken into consideration and the users with similar interaction profile is considered the closest. In this way, the user will be collaborated with the nearest neighbours. The neighbours in this system are the ones who have similar profiles that have been the result of similar interactions. This method concentrates more on the user and not the item. The interactions and the profiling of the users is more important, and the item is given least importance. This method takes the similarity value between the users which is then computed to check if the user interactions are similar to others in the same area. As mentioned, this method gives least importance to the item/product. This method is suitable when there are huge number of users who are interested in similar items. This method can be used when there are large number of users in a particular tourist spot. In such localities with similar users and their choices, it is obvious that the facilities are usually irrelevant because of limited choices. Thus, the user interactions are based on the profiles of others which is also similar because of the limited parameters available on the items. The modelling of the parameters is usually common to the interactions.

In the item-item method of modelling, the product/item is important than the user. If the user has rated or reviewed two items positively with good number of interactions. These two items have close relationship and many parameters are matching between the two. Thus, the model assumes that the similarity between these items is maximum. To recommend an item to the user, this concept is used. If a user searches for Item 1, he will be recommended Item 2 because many users had interacted with Item 1 and Item 2 in similar way and it was found that these two items are closely related. Thus, the recommendation becomes easier but effective. This method is efficiency when there are many items and the user choices are limited. This method concentrates on the items and not the users. The neighbouring user interaction is not considered in this mechanism.

Model based methods are slightly different. In this method, there is a modelling that processes the data after the relationship between the user and the item is established. The interaction is constructed repetitively to train the data. This modelling is useful because the data needs some processing before it can be used. This process is necessary when there are multiple inputs and relationships between the user and the item. Also, if the variables increase, it is not easy to use the memory-based method because of the non-processing. Memory based method works well for limited number of interactions and processing whereas the modelling method is suitable for high number of variables and complex data. Whenever the user interacts with the system, he/she does not contribute to the inputs of the system directly. Instead, the inputs are added to the mathematical model. All the processing of the interaction will be performed by the model before it is added to the design of the system. (Rocca, 2019).

## 2.2.4 Past research on Knowledge-based Recommender Systems

Çano and Morisio (2017) also argues that the third and another important approach used in the recommender systems is knowledge based. In this system, the inputs of the users and the items is profiles as in collaborative filtering, then a set of unique parameters are extracted from the inputs – called knowledge. This knowledge on the users and the items can be used by the recommender system to provide results accurately on the choice of the user. The difference between the collaborative and knowledge-based system is that there is no knowledge extracted in terms of collaborative system. The profiles are directly used unlike in this system. The knowledge extracted is not the directly available information from the user. It could be similar to the frequency of the user purchasing the item. There are two types in the knowledge system – case and constraint. The case system uses a comparison between the requirement and the recommendation. If the user requirement and the recommendation is matched, the accuracy is said to be higher. The matching can be performed by checking how many proposed results users have approved of and used for booking the hotel. If the user has searched for a specific requirement and the case model has presented the accurate result, then it is likely that the user selects the first item of the results. The constraint system works well for the user who does not have strict constraints and is looking for a recommendation. If the user profile has not been created yet because the user is searching for an item for the first time, then the constraint system works well (Çano & Morisio, 2017).

## 2.2.5 Past research on Demographic Systems

This approach uses preferences of the users of a demographic area to recommend the results. For example, the users of a Hindi speaking state in India are recommended Hindi movies as the first choice. In this case, the demographic parameter used in the language. Likewise, there could be several demographic parameters including food choices, hot or cold weather, fashion choice, etc. that could be used for designing the recommender system. This method provides accurate results when the demography is chosen as more important than the content-based approach. This method also has disadvantages when the user is using the recommendation system for the first time because the user could be from a different demographic region and has different preferences. This method, however, works well if it is used in combination with user profiling and knowledge-based system (Çano & Morisio, 2017).

## 2.2.6 Past research on Critiquing approach

The preciously discussed approaches do not perform well if the user inputs and the profile details are not available. For the first time user, a set of default parameters are used. Then the user inputs are used to modify/refine the results. In this case, if a first-time user searches for a hotel, and then after the results are displayed, changes the location parameter, it is considered as a good input for the system (Çano & Morisio, 2017). This method of using the feedback is referred to as critiquing approach.

## 2.2.7 Past research on Hybrid Systems

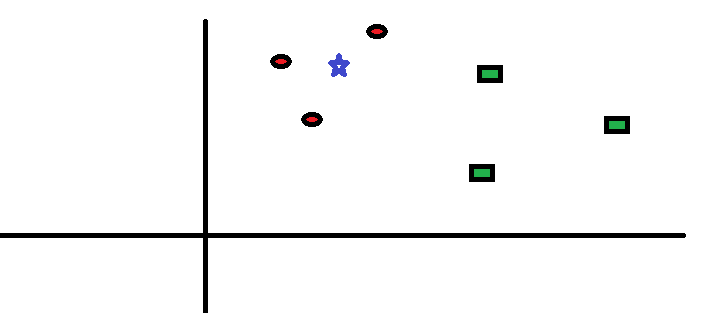
It is possible that the recommender system requires different types of designs for improved accuracy. In such cases, two or more approaches are combined, and the resultant design is called a hybrid system. It is also possible that the design combines the different approaches based on the application and the customization required for the specific case (Çano & Morisio, 2017). This project also proposes a hybrid system in which content-based and collaborative approaches are combined for improved accuracy.

## 2.3 Algorithms used for recommender systems

**Association Rules –** This technique of data mining is a popular and common algorithm used for recommender systems. The association rules take the input from the data and then extract one or many rules that associate the data with the parameters. When any similar data is found to be associated with the parameters, the result is positive. The applications of the recommender system play an important role before selecting association rules for the design. This approach works well if the association rules are limited to a few parameters. If there are several parameters associated with other parameters, the accuracy of the system decreases.

**Classification –** The classification method is also a popular method used for classifying the data based on any variable. Bayesian classifier is popular and usually found in many articles in the research literature. Bayesian classifiers work based on data by using the posterior probability of the class of items. This method can be applied in the hotel recommendation system for the ratings and reviews of the hotels. The user may choose the hotel above a specific rating. The Bayesian classifiers can be modelled to search the hotels only above the rating specified by the users. In this method, there is minimal relationship between the collaborative method and the user profiling. The collaborative method defines the relationship between the user and the item. However, the Bayesian classifier does not relate to this relationship and only adds to the existing methods. This method can enhance the accuracy of the search results and provides suitable ratings and other classification parameters in addition to the design parameters of the recommender system (Çano & Morisio, 2017).

**KNN –** One of the most commonalgorithms is the k-nearest neighbour algorithm. This method is used to generalize the association rules. The profiles of the nearest data item is considered to improve the accuracy of the result. This method can be combined for small datasets with limited parameters. It is not preferred as an independent algorithm for hotel recommendation system. KNN is the fundamental system used in most of the artificial neural network-based systems. If there are different data point that has been categorized in to various classes, predicting the results in the same point is difficult, which is where the KNN algorithms come in handy.



*Figure 2.2 Working of KNN model*

In the example shown in figure 2.1, the red circles and green squares are used. If the class of blue star should be determined, it can either fall into red circle or blue square category. However, it is nothing else but one of these. The KNN works on the nearest neighbour and takes the votes from them. If K=3, then the result is as shown in figure 2.2.



*Figure 2.3 Working of KNN – result*

## 2.4 Challenges of Recommender System

The recommender systems work on data. Large datasets and the accurate design of model will result in good results. Any changes in these parameters will result in inaccurate results. The following challenges are described as relevant to the proposed design.

**Initialization**

This is a common challenge for the recommender systems. This issue is also called Cold-start. This problem occurs when the user is searching for a hotel for the first time. At this point, the following issues arise that influence the accuracy of the results.

* Lack of user profile – The user profile is not available when the user is searching for the first time. Thus, the details of his/her likes and dislikes is not available for any recommender system.
* Lack of past searches – The system has no information on the past purchases or the hotels. This is also a major problem for the recommender system since the system provides more accuracy with the data of the users.

**Scalability**

The growing number of users and the number of choices available for the users create a massive database of various sets. The collaborative approach works with the inputs of users from same interests. The system is accurate only when the users of the same background search for the results. As the number of users and their choices increase there are huge number of parameters to select from. Also, the issue increases when the increasing number of users deviate from the common interests and do not fall in the same group as before. This scalability issue is also increasing every day.

On the other hand, the growing number of hotels with different options and choices to choose from, is creating a huge database of the hotels. The data should always be updated every day to keep the accuracy high. It is extremely difficult to update the data every day because of the preprocessing steps involved in the design. Another difficulty is scalability is that the system should always be up and running. It is not advisable to update the system offline. This is a challenging task for the designers and engineers (Jain, Grover, Thakur & Choudhary, 2015).

**Privacy**

One of the most debated topics in these times is privacy. The issue with the digital data stored is a big issue. A large amount of data is usually collected by the recommender systems about the users to process and provide accurate results. This data is collected using the digital systems. Thus, the data is also stored by the service provider. This data may include personal and sensitive information. It is important that the data collected by the recommender system is stored and processed carefully without any leakage to any third-party service providers. The new regulations for the data protection should comply with the *General Data Protection Regulation (GDPR)* proposed and implemented by the European Union.

**Predictability**

The vast number of items and the products available in the market along with the familiarity of the user with the products has led to the predictability issue. The users are now familiar with the results. The recommender system may offer same results to the repeated user in case the search parameters and the user profile has no changes. If a frequent traveler visits the same place again with the same search parameters, it is highly possible that the results of the recommender system are the same in the same order. It is possible to solve this issue by analyzing large amount of data on a regular basis. The issue may be solved by updating the design and the dataset regularly to include new parameters along with new models to the algorithm (Jain, Grover, Thakur & Choudhary, 2015).

## 2.5 Summary

Literature Review details every important concept, model, and the research available in the literature about the specific topic. In this chapter, the recommender systems are described in different segments. At first, the topic and the working of recommender systems is described. The approaches used in the recommenders is then listed and discussed. Later, some of the popular algorithms used in the recommender systems are briefed. A set of peer reviewed articles are then reviewed as part of the survey. Literature Review chapter offers an in-depth knowledge of the recommenders and provides an insight to the research gaps in the articles and the peer reviewed research in the literature. In the LR, we have discussed that there are different approaches for the recommender systems. Content-based and collaborative systems are the common and used in this proposed solution. The hotel recommendation system has huge demand in the developed and developing countries alike. There are over 450000 listing under one online recommended and booking systems called OYO. The demand for the recommender systems is growing every day. The academic research is detailed in this section, used in the design of this proposed system. The next chapter, Design Methodology, presents the different methods used in the design of the solution in connection with the content discussed in the literature review chapter.

# **Chapter 3: Methodology and Design**

## 3.1 Introduction:

The design of the recommender system is analysed in different modules. There are various methods in which the recommender systems are designed. A typical recommender system and its different modules are explained in this section before explaining the design of the proposed hotel recommendation system. The proposed hotel recommendation system is a hybrid model with content-based and collaborative system with feature selection.



Figure 3.1. Block diagram of a typical recommender system

The modules of the recommender systems are explained in detail before the modelling with parameters (Hirakawa, Satoh, Hisazumi & Shibata, 2015).

One of the most important blocks in which there are three parts – external, user, and social. The context block is the one that takes the input and feeds it to the recommender engine.

**User context** – The context or the details of the users in included in this part. The details of the user such as the personal details and the inputs given to the recommender engine such as the information of the past visit to a particular hotel. This context is exclusive to the user. It is important to note that this is the context and not the profile. The environment in which the user and the external environment is separated.

**Social context** – The details of the hotel that is obtained from the external sources such as internet sources or social media. This information is also crucial for the engine because there are inputs about the hotel and the tourist place that is not given by the user or the hotel. This information may be provided by other tourists, government authorities, local people etc. This data is often added to the recommender system to improve the informational accuracy.

**External context** – Another important context in which the environmental parameters such as climate and road conditions are sent to the engine. As mentioned above, this information is not related to the user or the item but additional data to support the engine with more inputs for improved accuracy (Hirakawa, Satoh, Hisazumi & Shibata, 2015).

The profiling of the users in different ways is an important part of the design. For accurate results, the profiling of the user is classified into the following types.

**Explicit** – The profile information of the user that is explicitly available such as the data provided by the user in the form of digital data on the platform. This data may include the personal details such as the name, age and other basic details. This data allows the system to perform a basic profile of the user. As simple as profiling a young male tourist who is travelling alone for a hill station. These details will be provided by the user.

**Implicit** – This data to this profile is not provided by the user but the system generates this. This detail is not available readily, but data processing should be done to obtain details such as the information of the previous visits and any ratings to a particular hotel.

**Requirement** – After the implicit and explicit profiling, the user may still have specific requirement for that particular trip. This information is not provided at the registration stage and cannot be obtained by the system. This data is entered by the user at the later stages after a set of hotels are recommended. This data will complete the profiling process because the basic details of the user, extracted data, and the specific requirements are all available to be fed to the engine.

**Repository** – The data of the hotel along with the facilities and the public information is contained in the repository of the database. This block classifies the data into separate parts for easy modelling and analysis. Firstly, the information of the destination is included. This information is relevant and is required for modelling the design. The content includes the multimedia information and any digital material about the place and the hotel (Hirakawa, Satoh, Hisazumi & Shibata, 2015).

There are many steps involved in the design of the proposed system. The data to be loaded to the system to test the performance is taken from the datasets available from different sources. For this project, a list of hotels available in Berlin are taken. Also, a dataset containing the reviews of the hotels is taken. These two files are taken in CSV format. The CSV format includes the data values in a particular format with delimiting values. This format is easy for the system to process and analyse because there is no issue with the format of the data as it is already formatted and available for any processing.

## 3.2 Dataset

The dataset with the variables will be discussed and analysed in this section. There are 82 variables in the dataset. However, a few of the most important ones are explained in this section.

***Table 3.1 Description of variables of Hotel listings in Berlin***

|  |  |
| --- | --- |
| **Variable** | **Description** |
| Id | A unique id for the hotels |
| Name | Name of the listing |
| Summary | A brief description of the listing |
| Space | Type of the space |
| Experiences Offered | Types of accommodation services offered |
| Neighborhood overview | A brief description of the neighborhood of the listing |
| Transit | Ways to reach the listing place |
| House rules | Any rules for the users are listed here |
| Host id | A unique number that identifies the user |
| Host since | Number of months the host has been listed |
| room\_type | Type of room |
| accommodates | Number of people listing can host |
| bathrooms | Number of bathrooms |
| bedrooms | Number of bedrooms |
| beds | Number of beds |
| bed\_type | Type of bed |
| amenities | List of amenities provided |
| square\_feet | Area of the hos space |
| price | Price of the hosting per day |
| weekly\_price | Price of the hosting per week |

## 

## 3.3 Use Case Diagram

Use case diagram is a pictorial representation of the interactions between the user and the system. The use case diagram lists all the possible ways the user can interact with the system. It can identify and analyse all the system requirements and the interconnections between each module of the system. The proposed hotel recommendation system is divided into frontend and backend. The frontend is the first layer that takes the input from the user and processes it at the first layer, then sends the data to the backend. The backend processes the data as per the programmed algorithms before the data can yield results. The user data is taken and fed to the data pre-processing unit in the backend. This data is then sent to the content-based and collaborative system before it is combined in the hybrid system. The results of these algorithms are then boosted for performance using the machine learning algorithms. The results are sent back to the user as shown in figure 3.1



Figure 3.2. Use case diagram of hotel recommendation system

## 3.4 Summary

The Design chapter discussed the methods and algorithms used in the proposed design. The chapter also explains the dataset used in the proposed system for analysis. The use case diagram helps understand the interactions between the user and the system.

# **Chapter 4: Implementation**

This chapter describes the implementation of the proposed hotel recommendation system. Description of data and pre-processing of the data is discussed in this chapter

## 4.1 An insight over a dataset

The dataset consists of various parameters that are used directly or indirectly to filter the results using content-based and collaborative method. This dataset is analysed and the inputs of the data are fed to the data pre-processing module before it is processed for other patterns.

## 4.2 Graphical representation of the data

A few of the data parameters of the dataset are visually represented. Only 4 of the 82 parameters are visually represented in this section. However, it is important that the average value of each of these parameters is what needs to be analysed. This data representation for all the variables in the dataset can help understand the data patterns in the data.

There are various parameters that are plotted and analyzed. A graph of two parameters has been shown in figure 4.3. The number of bedrooms and the average price is plotted. It was found that the dataset includes the average price which is mostly linear as the number of bedrooms increase except for the average price of 5 bedrooms which as been on the higher side at 572$ average. This data could also prove that if there is a higher demand for 5-bedroom hotels, the prices can be increased.

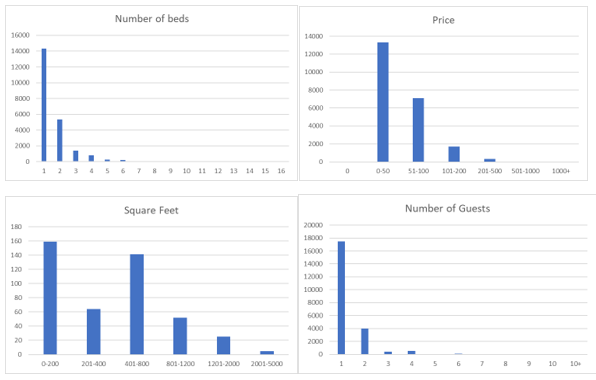


Figure 4.1. Visual Representation of Number of beds, price, square feet, and Number of guests

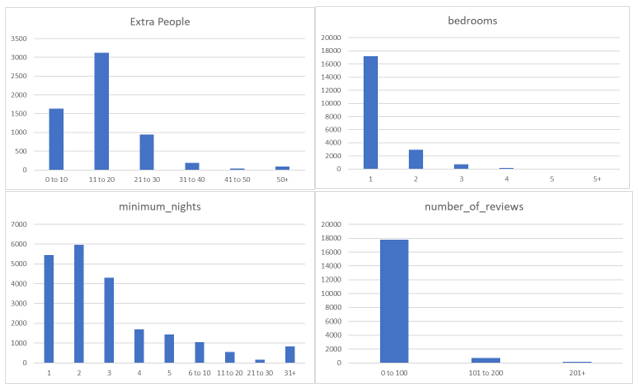


Figure 4.2. Visual Representation of Extra people, bedrooms, minimum nights, and number of reviews

In the figure 4.4, a graph of average price and square feet area of the room has been plotted. It is found from the graph that the price increases linearly as the area increases except for the 0-100 square feet room because of the high demand.

*Figure 4.3. Graph of Number of bedrooms and average price.*

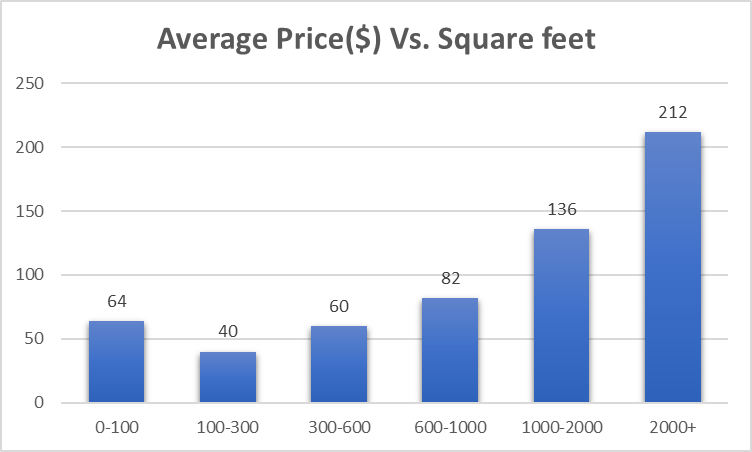


Figure 4.4. Graph of Average price Vs. Square feet.

## 4.3 Feature Selection

**Language Detection –** The first and foremost step is to detect the language in which the comment is written. There are different Word Sense Disambiguation and Sentiment Analysis tools that can be directly used to detect the language. In this design, *langdetect* module is used in Python. This value is added in the column called *language.*

Analysis of a set of data reveals that English is the most popular language among reviewers. Figure 3 shows a list of languages and their frequency.

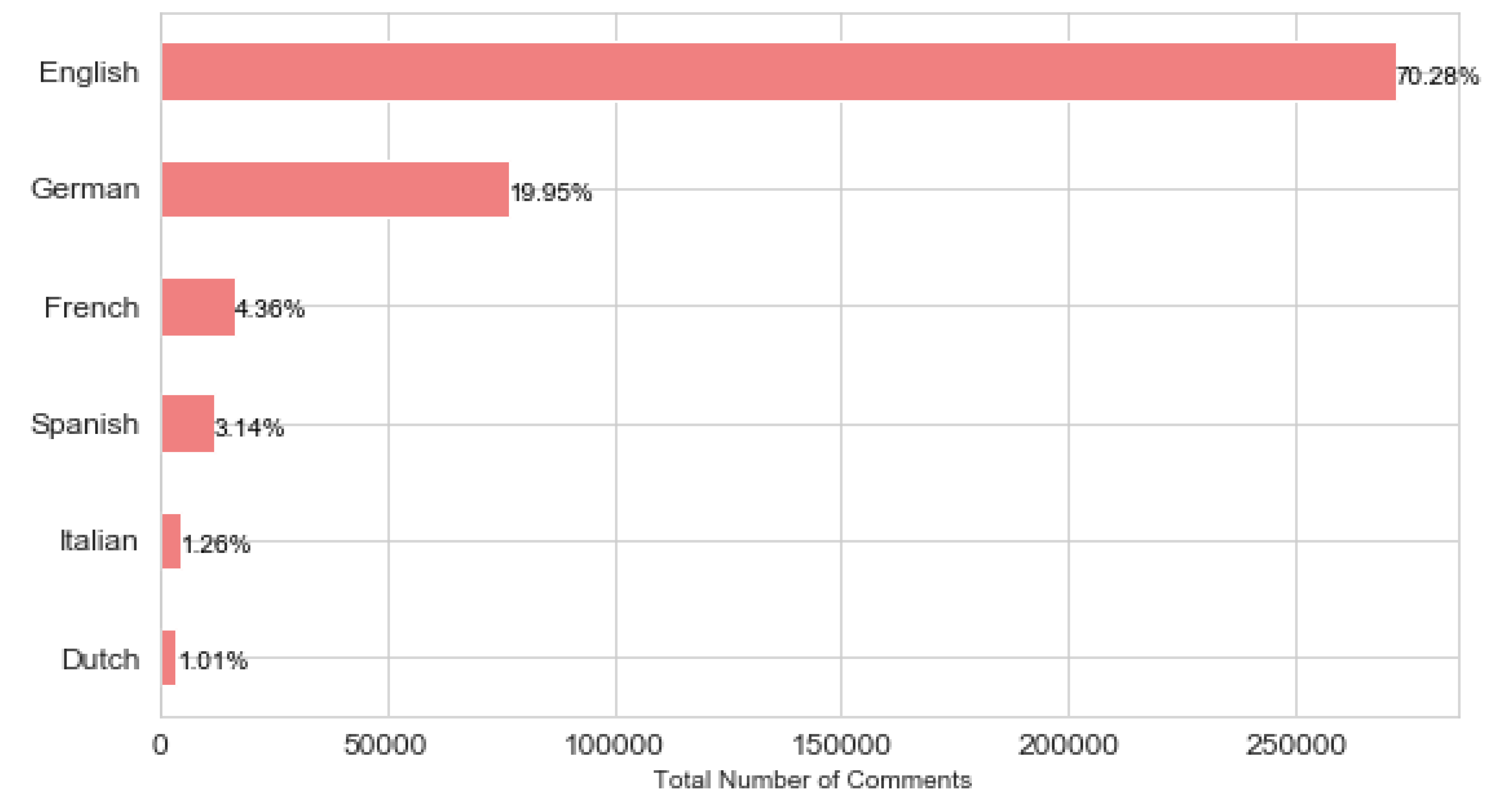


Figure 4.5. Languages and their frequency

If the review is not entered in English language, it must be translated to English because the sentiment analysis used works well only with English. It is possible to analyse the sentiment of the reviews in other languages. However, the research and the accuracy of the English language sentiment analysis is well above that of other languages.

For example,

*“Un appartement tres bien situé dans un quartier historique de l'est tres agréable, le métro est*

*tout proche. Pas de connexion internet ni de télé mais cela nous pousse encore plus a découvrir Berlin. Jan est tres accueillant et nous a donné quelques conseils pour nos visites.”*

**Sentimental score for the above comment: Sentiment (polarity=0.0, subjectivity=0.0)**

**(scale will be from -1 to +1)**

**-Sentimental analysis is done using text blob.**

**Translated sentence of the above non-English sentence**

An apartment very well located in a historic district of the east very nice, the metro is nearby. No internet connection or TV but it pushes us even more to discover Berlin. Jan is very welcoming and gave us some tips for our tours.

**Sentiment score for above translated comment: Sentiment (polarity=0.33599999999999997,**

**subjectivity=0.42000000000000004)**

In this way, the sentimental score is calculated for every review using text blob library.

The conventions used for sentiment analysis is as follows.

Sentiment (polarity=0.33599999999999997, subjectivity=0.42000000000000004)

Select polarity.

Polarity > 1 - positive review

Polarity ~ 0 - neutral review

Polarity <0 - negative review

Polarity range [-1,1]

Scale the polarity score to [1, 5]

Model overfits the rating if the scale is between [-1,1]

So, scale them to [1,5]

Formula

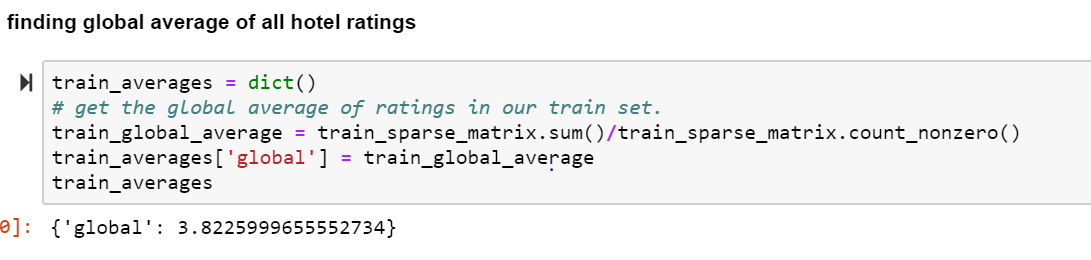
Scaled rating = ((new\_max - new\_min) \* (current\_rating - old\_min) )/ (old\_max - old\_min) + new\_min

**Feature Engineering 2**

With the sentimental scores of a large number of reviews, it is possible to get a global average rating. This feature is engineered as part of the proposed system. This feature is not available in the existing systems. As mentioned above, it is possible to engineer as few more features for each module to improve the accuracy of the system.

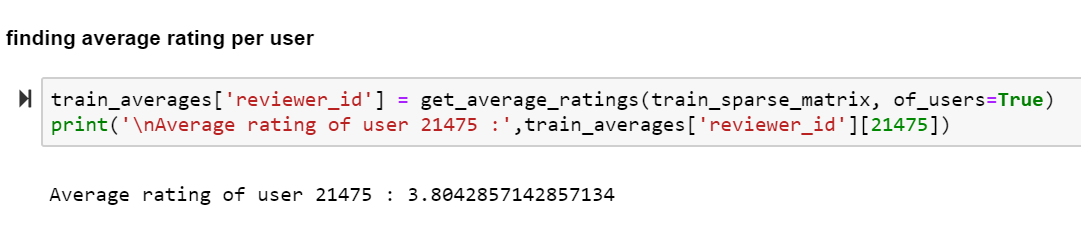
Sparse matrix is implemented for both test data and train data. It is implemented to make the storage efficient as sparse matrix will store index of one, so the zeroes will be implicitly ignored.

**Global average rating-** The average of all the ratings available for a particular hotel. The ratings in English and other languages is calculated.



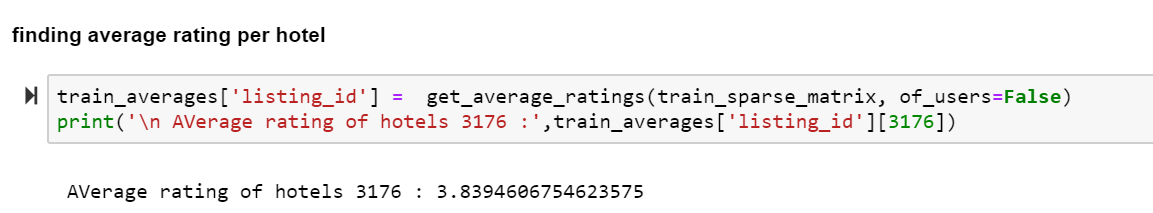
*Figure 4.6 Global average rating of all hotels*

**User Average Rating -** The average of all the ratings given by the users for a particular hotel. The rating of one hotel given by many users.



*Figure 4.7 Average rating per user*

**Hotel Average Rating –** The average value of the ratings for the hotel in comparison with other hotels.



*Figure 4.8 Average rating per hotel*

**Merge listings and reviews data frame of listings-id**

At this point, the listings and the reviews are to be listed in the listings\_id column. The task is to use the past ratings and reviews on a particular hotel and predict the future bookings. This method ensures that the data modelled can also be used predicting the future bookings. The feature engineered values play an important role in this regard. The overall system block diagram is shown in figure 4.

## 4.4 Evaluation of the System

**Metrics**

**Rmse and mape are evaluation metrics used for this project**

## 4.5 Software and Packages:

The following set of software tools and packages are used in this project. The code is run in Jupiter notebook and requires following packages

* Python (>= 3.5) - Python is a great tool for data analysis and for backend processes. There are many popular companies and web platforms using python for development and analysis including Google. Python version 3.5 and above works well for the implementation of this project. The web-based installers are part of the Python tool.
* NumPy (>= 1.11.0) – NumPy is a library for Python used for processing of large data including arrays and matrices. The non-optimizing part of Python can be implemented using NumPy.
* SciPy (>= 0.17.0) – an open-source Python library used for computing data. Image processing, data computing, and other signal processing operations can be performed using this library. SciPy version 0.17.0 and higher can be used to implement this project.
* joblib (>= 0.11) – A set of tools used in Python for parallel computing. Version 0.11 and higher can be used.

For Light gbm VC runtime is needed if Visual Studio (2015 or newer). Other libraries include seaborn, scikit-learn, langdetect, geopy, textblob, catboost, lightgbm, xgboost

RMSE, Catboost, LightGBM and XGBoost -These tools are used to improve the performance of the results. These machines learning based methods are used for optimization and performance enhancement.

**RMSE** – The Root Mean Square Error is one of the most common standards used for predictions. This method can also check if the regression and the data points are located far from the data. The major advantage of the RMSE is that the method can give accurate understanding of the data points that fit nearest the best fit line. Thus, if the standard and the best fit data is deviating, this method can ensure the distance and the deviation can be determined.

**Catboost** – An open source tool used to check the gradient boosting the decision trees/ This method is known for its accuracy, robustness, easy, practical and also extensible. This method is used in this project for the results on decision tree methods.

**LightGBM** – Machine learning is an advanced technology that can be used for understanding and programming the tree-based machine learning algorithms. The tree-based method explains how the algorithm can be boosted for better performance. This method is suitable for high speed applications like the hotel recommendation system and the results of the LBGM are accurate and supports some graphical way of learning system too.

**XGBoost** – XGBoost is the algorithm that can be used for the structured and the tabular data like the data in the Hotel Recommendation System. This method has a model that can be used for gradient, stochastic, and regularized boosting. Also the system can help program the parallelization, distributed computing, out-of-core computing, and cache optimization.

## 4.6 Summary:

The implementation process is simple and straightforward. The implementation is done in different parts. Content-based System – The content-based system is implemented first using the data parameters in the dataset. This is done by fitting a linear model as the above features using SGD Regression. Collaborative system – The dataset of the reviews is processed by translating and analysing the polarity of the reviews using sentiment analysis tools. The system is the result of the processed data. This process improves the accuracy of the results. The feature extraction system is part of this stage of implementation Hybrid system – The results from content-based and collaborative system are combined and then the results are displayed.

The system is implemented by data pre-processing, content-based and collaborative methods, and the software tools used for the process are explained.

# **Chapter 5: Testing and Results**

The testing and results chapter details the processes of implementation in terms of operation. The system is tested, and the results are analysed. The problems discussed in the literature review section is implemented and tested. The process is explained in the flowcharts.

## 5.1 Flow chart of the whole process

The flowchart of the entire process is shown in figure 5.1. There are three levels that the data from the dataset will be input to, as shown. The content-based filtering method, pre-processing module and collaborative filtering method is loaded with data. The data from the content-based method is processed and the features are added to the data. Similarly, the data from collaborative method is added with features. The central block has the data from these methods along with the features. These features are then sent for evaluation for results of the hotels. Each of these models are evaluated and the performance of these methods is boosted using the tools such as XGBoost, Light GBM, Catboost, and Adaboost.

*Figure 5.1 Flowchart of the Content-based process.*



*Figure 5.2 Flowchart of the whole process*

## 5.2 Results

**Data Pre-processing**

## Listings\_summary.csv

**Define the columns we want to keep from Listings\_summary.csv**

columns\_to\_keep = ['id','name', 'space', 'description', 'neighbourhood\_group\_cleansed',

'latitude', 'longitude', 'property\_type', 'room\_type', 'accommodates', 'bathrooms',

'bedrooms', 'bed\_type', 'amenities', 'square\_feet', 'price', 'cleaning\_fee',

'security\_deposit', 'extra\_people', 'guests\_included', 'minimum\_nights',

'instant\_bookable', 'is\_business\_travel\_ready',

'cancellation\_policy','host\_is\_superhost','number\_of\_reviews']

The following columns or the details of the listings are used.

* Name – The name of the hotel/accommodation
* Space – The type of space in which hotel is located
* Description – A brief description of the hotel accommodation including basic details.
* Neighborhood - The name and description of the neighborhood with location details
* Latitude – Enter the latitude of the hotel
* Longitude – Enter the longitude of the hotel
* Property type – Type of the property – hotel, homestay, private house, etc.
* Room type – Single or double room.
* Accommodates – How many people can one room accommodate?
* Bathrooms - How many bathrooms does it have?
* Bedrooms – How many bedrooms does it have?
* Bed type – Single or double bed
* Amenities – Hot water, Wi-Fi, etc.
* Square feet – Area of the room
* Price – Price of the room per night.
* Cleaning fee – Is there any cleaning fee, how much?
* Security deposit – Is there any security deposit, how much?
* Extra people – How many extra people can it accommodate?
* Guests included – How many people are staying in the room?
* Minimum nights – How many nights is the minimum?
* Instant bookable – Is the room ready to be bookable?
* Is business travel ready? – Is the room ready for any business travelers?

The above-listed values should be entered by the guest user. All the values are important and are required for the hotel recommendation system to be accurate. However, it is not easy to get the user to fill all the details. Therefore, there is no mandatory rule for all the values to be entered. If a few values are missing in the above list, the values are normalized to the standards. For example, the number of beds offered, if not entered, will be usually two. A list of such normalized average values is researched and added to the system in case the user does not enter them. In case there are more than 50% of the value missing, then the columns that host these values are dropped from the dataset. The result of the recommender system will be accurate if a greater number of values are entered. Otherwise a default value of the results will be obtained.

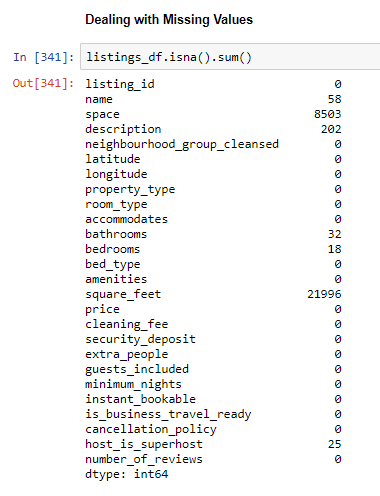


Figure 5.3 Missing Values

## Remove Duplicates

Any duplicate values are removed from the list. It is possible that the default values are present when the user enters the values. Also, the duplicate values entered by the user should be removed before processing that data.

## Feature Engineering

There are certain features that are obtained from the entered data. These features are not directly available from the data, but they are processed and extracted from the process called feature engineering.

**TF-IDF**

The term frequency and Inverse document frequency algorithms are used for the natural language processing operation of the language analysis. The language detection and translation can be performed using TF-IDF. The features of the reviews (text) is extracted using the algorithms.

The higher values of the words in a document would obviously mean the importance of the word is higher. This way the number of words in the document is determined.

The term frequency is combined with the inverse document frequency to determine the number of documents in which a particular word is used frequently. These operations are the first steps of natural language processing.

**Feature Engineering 1 – Distance to Central Berlin**

It is possible that the location of the neighborhood is measured from different areas of the city. It is therefore necessary that a central location is used to measure the distance. Central Berlin or the City Center is used as the default location instead of the neighborhood distance to measure the distance. This distance is calculated using the feature engineering process. This distance is not entered by the user or the hotel management, but it is measured as the distance between the latitude and longitude values of the city center and the values of the location.

**Feature Engineering 2 – Amenities at the hotel**

There is a list of amenities that are common among several hotels. Such amenities need not be engineered. Instead there are certain other amenities that are not usually mentioned by the hotel management. These amenities are again engineered using data processing methods. The following amenities are engineered. Any other amenities may also be engineered provided they can be mathematically modelled, and a value determined.

* Laptop friendly workspace- If a traveler is on a business trip and requires a space to work using a laptop, the system should check different parameters and determine if the hotel has a laptop friendly space for business work. This amenity can be engineered based on values of square feet are of the room, number of beds, and other amenities.
* TV – This amenity can be directly known if the hotel management has entered it. Otherwise, it can be feature engineered using a set of values.
* Kid friendly Accommodation – This feature is an engineered result. It is possible to determine if the accommodation is kid friendly or not using a set of parameters. The modelling of the features can be determined the value of this amenity.
* Smoke friendly – This is another amenity, which is feature engineered. The number of smoking areas, or the interiors of the room that can allow smoking, or any other smoke alarm system data can be used to engineer the value of this amenity.
* Greeted by the host – This is a friendly gesture by the host. It is a custom in many tourist places to greet the guests with traditional welcome. This can be a great delight to many guests. Based on the data provided by the hotel management this feature can be engineered into another category called *Delights.* or similar.

It is important to feature engineer a set of amenities. There are several hotel recommendation systems with default list of amenities to choose from. However, the proposed system is unique in this sense. The system is so designed that many more amenities can be included, engineered and categorized before they are fed in to the recommender system.

The following amenities are considered most important, based on the data collected through research, shown in figure 5.4.

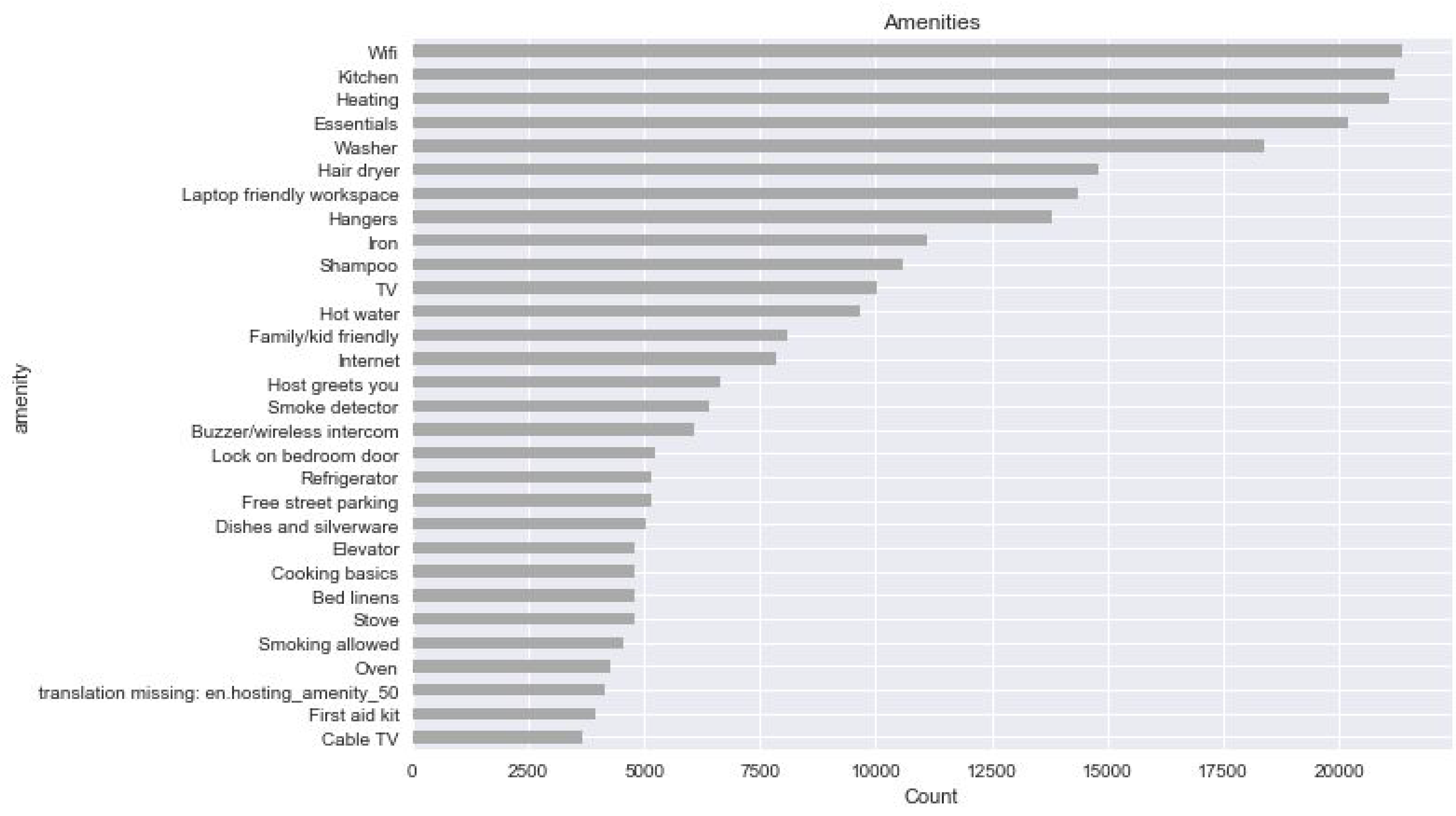


Figure 5.4. List of amenities ranked based on number of searches by the users.

## Reviews

The next step is to use the reviews of the hotel from various users to model the recommender system. These reviews offer valuable information based on different users and values. However, the reviews are not available directly in the form of values. They are entered in the text form in any language. The text is descriptive in common language and style. This data entered by many users is taken and then translated, analysed and used for recommended systems.

The following values are taken for the analysis of reviews.

* Listing ID – The unique ID of the hotel. This represents the hotel and its details.
* Reviewer ID – The unique ID assigned to the reviewer/user at the time of booking. This ID ensures that the reviewer is genuine, and the stay is verified.
* Reviewer Name – The name of the reviewer is entered
* Date – The date of writing the review
* Comments – The review of the accommodation. This column may be further divided into different values to get the specific details in the form of ratings for amenities, location, food, Wi-Fi, etc.

If there are any missing values, a set of normalized are replaced and any duplicate values are deleted. If there are more than 50% of the values missing, the entry is considered invalid.

## Results of Collaborative Filtering

## For the processing of collaborative filtering, a python library is used for building recommendation system. The data from the surprise library should be transformed using a reader. Reader class is used to parse a file containing ratings. Such a file is assumed to specify only one rating per line, and each line needs to respect the following structure:

## user; item; rating; [timestamp]

## Utility functions for Surprise models

## It is just to make sure that all our algorithms should produce same results for each time they run

## def get ratings(predictions): It gets(actual\_list,predicted\_list) ratings for the given list of predictions(prediction is a class in Surprise).

## def get\_errors(predictions): It gets “rmse” and “mape”, gives list of prediction objects.

## def run\_surprise(algo,trainset,testset,verbose=True):It will return predicted ratings, rmse and mape of both train and test data.

## Return train\_dict,test\_dict : It returns two dictionaries, one for train and the other is for test. Each of them has 3 key-value pairs, which specify “rmse”, “mape”, and “predicted ratings”.

## Execution Steps

## Train the algorithm with the trainset

## Evaluating train data

## Get the train predictions (list of prediction class inside Surprise).

## Get predicted ratings from the train predictions.

## Get “rmse” and “mape” from the train prediction

## Store predictions, rmse, mape in train dictionary.

## Evaluating test data

## Get the predictions (list of prediction classes) of test data

## Get the predicted ratings from the list of predictions

## Get error metrics from the predicted and actual ratings

## Store them in test dictionary

## Example

## {‘rmse’:0.4162251500802891,

## ‘mape’:8.344213703388602,

## ‘predictions’:array([3.70175047,3.82384748,3.90888846,…,3.80178785,3.8017875,3.80178785])}

## Results of SVD Matrix Factorization

## The Singular-Value Decomposition or SVD for short is a matrix decomposition method for reducing a matrix to its constituent parts in order to make certain subsequent matrix calculations simpler

## Execution Steps

## Initialize the svd model with parameters (n\_factors:number of latent factor)

## Run the model using utility function rnn surprise mentioned above

## Just store these error metrics returned in our models evaluation datastructure

## Score

## 

*Figure 5.5 Score evaluation using SVD matrix factorization*

## Training the model

## 

## Train Data and Test Data

## 

*Figure 5.6 Evaluation results using SVD factorization for test and train data*

## No Considerable bias and variance problem considering it’s a recommender system.

## Results of Content Based Filtering

## Hotel Names and Ratings are used for content based filtering and other columns are ignored due to missing values of more than 50%. Term frequency -inverse document is a weighing factor used as a search tool in scoring and ranking a file given a user query. Simple ranking function are computed by adding tf-idf for each term

## tf-idf = TF (w)\* IDF(w)

## TF(w) = (Number of times term w appears in a document)/(Total number of terms in a document)

## IDF(w)= log e (Total number of documents/Number of documents with term w in it)

## SGDRegressor (Stochastic Gradient Descent) are used in large scale machine learning problems commonly used for text classification and natural language processing.

## Score

## 

*Figure 5.7 Score evaluation using SGD Regressor*

## Results of Hybrid Models

## Total Features = (Initial features + feature1 + feature2)

## Considering the Total features

## 

*Figure 5.8 List of selected features*

|  |  |
| --- | --- |
| Feature1 | “svd” |
| Feature2 | “content\_sgd” |
| Target | “Rating” |

## Utility functions for running regression models

## def get\_error\_metrics(y\_true,y\_pred) : It gets rmse and mape given actual and predicted ratings.

## Def run\_boosting(algo,x\_train,ytrain,x\_test,y\_test,verbose=True) :

## Execution Steps

## Fit the model

## Get the rmse and mape of train data

## Store the results in the dictionary

train\_results={'rmse':rmse\_train,‘mape':mape\_train,'predictions':y\_train\_pred}

* Return Test and Train results.

**Gradient Boosting Algorithm:**

The Gradient boosting algorithms are used along with classification or regression problems which produces an ensemble of weak prediction models. This model also follows stage wise development. A gradient boosting algorithm can be treated as optimization algorithm for a suitable cost function.

**XGBoost (Extreme Gradient Boosting):**

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

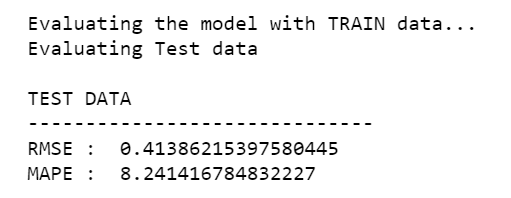
* Initialize our XGBoost model using XGBRegressor

Parameters:

Learning Rate: Slow or Speed up the learning of gradient boosting model

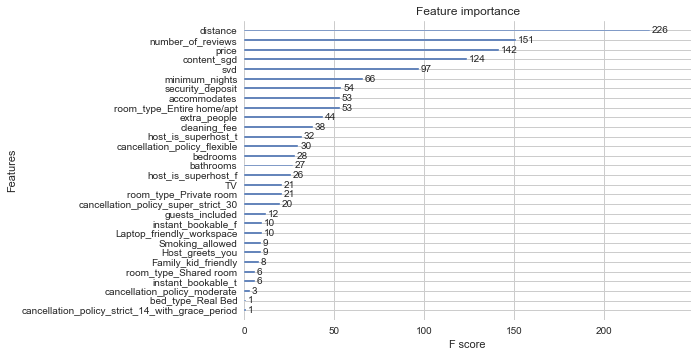
N\_estimators: The number of trees (or rounds) in an XGBoost model.

* Store the results in models\_evaluation dictionaries



*Figure 5.9 Score evaluation using XGBoost*

**Feature Importance**



*Figure 5.10. Feature Importance – F Score*

**LightGBM Model (Light Gradient Boosting):**

Light GBM is a gradient boosting framework that uses tree-based learning algorithm. Light GBM is prefixed as “Light” because of its high speed. Light GBM can handle the large size of data and takes lower memory to run. Another reason of why Light GB is popular is because it focuses on accuracy of results. LGBM also supports GPU learning and thus data scientists are widely using LGBM for data science application development.

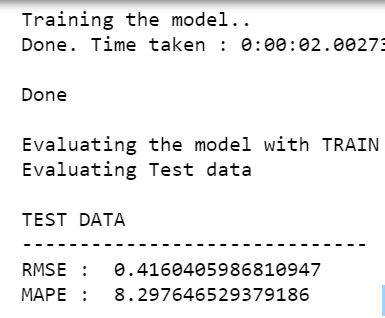
* Initialize our LGBM model using LGBMRegressor

Parameters:

Learning Rate: Slow or speed up the learning of gradient boosting model

N\_estimators: The number of trees (or rounds) in an lgbm model

* Store the results in model’s evaluation dictionaries



*Figure 5.11 Score evaluation using LightGBM*

**Feature Importance**

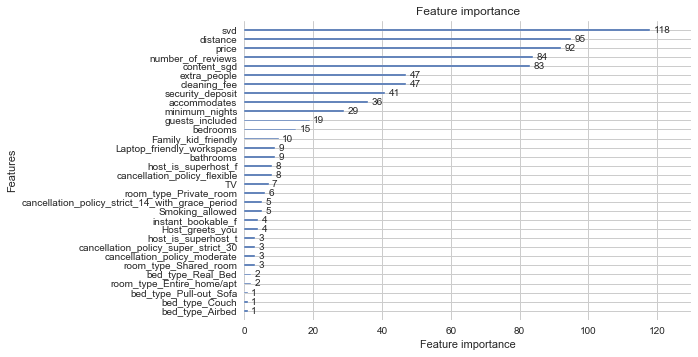


Figure 5.12. Feature Importance 1

**AdaBoost Model:**

It is one of the best boosting algorithms. Adaboost helps you combine multiple “Weak

Classifiers” into a single “strong classifier”

* Initialize adaboost model using adaboostregressor

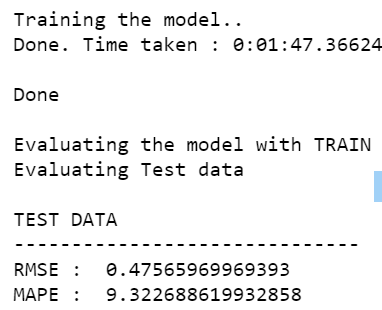
Parameters:

Learning Rate: Slow or speed up the learning of gradient boosting model.

N\_estimators: Learning rate shrinks the contribution of each regressor by learning rate. There is a trade-off between learning rate and n\_estimators.

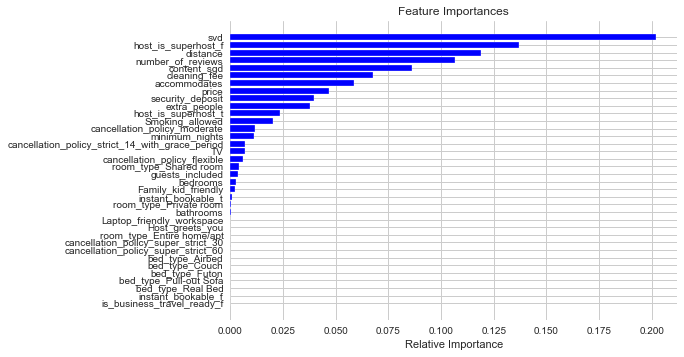
Loss: The loss function is used when updating the weights after each boosting iteration.

* Store the results in models \_evaluation dictionaries.



*Figure 5.13. Score evaluation using AdaBoost*

**Feature Importance:**



*Figure 5.14. Feature Importance 2*

**CATBoost Model:**

“CatBoost” name comes from two words “Category” and “Boosting”

* Initialize CatBoost model using CatBoostRegressor

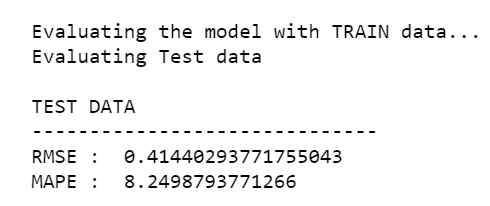
Parameters:

Learning rate: Used for reducing the gradient step.

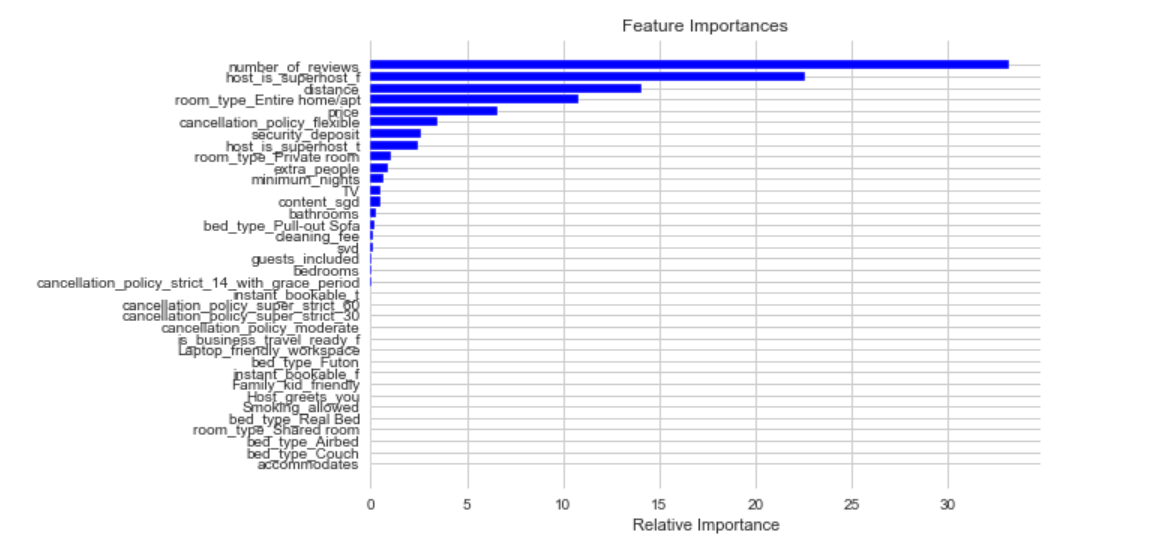
N\_estimators: The maximum number of trees that can be build on solving machine learning problems.

Depth: Depth of the tree.

Loss: The loss function is used when updating the weights after each boosting iteration.



*Figure 5.15. Score evaluation using CatBoost*



*Figure 5.16. Feature Importance 3*

**5.3 System Evaluation:**

The research has been implemented to check the performance of the results of a hotel recommendation system. The performance of the system is evaluated. The research has used content-based system and collaborative method and then the results are merged. The performance is boosted using the algorithms such as light GBM, Catboost and Adaboost.

### **5.3.1 Root Mean Squared Error (RMSE)**

The standard deviation of the errors is measured using RMSE. The deviation is also a measure of how far the errors are operating from the given set of standard data. This value and the analysis will ensure that the data lies between the predicted and the actual values. The difference between actual and the predicted value also determine the accuracy of the project results. RMSE can be calculated using this formula.

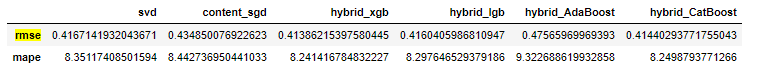
where recommendations = The number of outputs generated by the system

user preference – The number of inputs by the user.

N = The total number of recommendations.

### **5.3.2 Precision**

After the implementation of content-based and collaborative methods, and the performance boosting algorithms, it was found that the hybrid XGB algorithm has lowest RMSE. The MAPE value of the hybrid Catboost algorithms has found to be the lowest. Thus, the hybrid recommendation system and the performance boosting algorithms have been analysed and the precision of these systems is compared as shown in figure 5.9.



*Figure 5.17. Model Evaluation Results*

## 5.4 Summary

This chapter discussed the results of all the components of the design. The content-based system and collaborative system designed are discussed along with the results. The results are each algorithm is presented. Also, the RMSE and the results of evaluation are discussed.

# **Chapter 6: Conclusion**

## 6.1 Introduction:

Recommender systems are not new to the field of research. There is enough research on the recommender systems in the literature. The evolution of the recommender systems is a long journey from basic content-based systems to the advanced systems with several parameters and variables. The most popular types are content-based and collaborative based. Each method has its own advantages and disadvantages. Content-based method works on the inputs from the user whereas collaborative system works on the profile of the users and the data generated from similar users to recommend an item. However, there is limited research and data available on the hybrid systems which use both content-based and collaborative based systems. Also, the lack of feature engineering of different parameters and values to collectively improvise the features and in-turn the overall accuracy of the results is the research gap addressed in this project.

## 6.2 Conclusion

This research proposed a hybrid mechanism in which content-based and collaborative systems are combined. Datasets have been obtained from different sources. Data has been preprocessed before it is fed to the system. A set of details on the hotel accommodations in Berlin has been taken for the implementation of this project.

The additional features of the system are modelling a set of parameters into a feature. This concept is new and unique to the project. The values of the feature engineering are fed in to the hybrid system. Sentiment Analysis of the reviews is performed. Reviews from English and non-English language are taken. Non-English reviews are translated to English for better results of sentimental analysis.

At first, the content-based and collaborative systems are modelled separately. The results from both these systems are combined in the next phase. The hybrid system uses different tools to analyse the results. Finally, a comparison of the results from content-based and collaborative system is presented.

## 6.2.1 Advantages

The proposed design of the recommender system has the following advantages.

* The system is a hybrid system with the features of both content-based and collaborative systems.
* The system has high accuracy because of featured engineering.
* The system can be customized for other applications.
* The accuracy of the system can be improved with larger dataset and training
* The system can be improved using machine learning algorithms.

## 6.2.2 Applications

Recommender systems are popular among several applications. The proposed system has been designed for hotel recommendation system. However, with appropriate design and parameter changes, this system can be customized for the following applications.

* Entertainment – The recommender system can be designed for recommendation of TV shows on the online streaming platforms and movies. The system can also be customized for recommending on-demand services in the field of entertainment.
* E-commerce – The recommender system can be used for recommending the products on the e-commerce platforms. The design of the recommender system for e-commerce websites is a challenging task because there are a large number of items with several variants, colors, prices, etc. On the other hand, there are numerous categories and sub categories of items.
* Digital Content – The recommender system can be designed in the similar way for recommending the users with digital content in the form of news, entertainment, videos on YouTube, applications, etc.
* Service providers – The proposed recommender system can be used for offering various services such as financial services, rental sites, real estate, bus/flight booking, etc.

## 6.3 Future Research Work

The project proposes a hybrid model of hotel recommendation system. This system has been designed using different steps. The following points can be the inputs for further design of this project. Each part of the proposed system can be improved using advanced techniques and algorithms. This process can be performed using the tools with

Dataset – Larger dataset with different points and deviations should be used to improve the accuracy of the system. Different datasets may be used to test and compare the results

* A few more parameters may be used for feature engineering the amenities

The parameters of actual hotels should be used to check if there is a deviation between the results of datasets and real data.

* Testing – Various test cases can be used to test the performance, efficiency and robustness of the system.

# **References**

#### Ieeexplore.ieee.org. (2019). Trends, problems and solutions of recommender system - IEEE Conference Publication. [online] Available at: https://ieeexplore.ieee.org/document/7148534 [Accessed 2 Sep. 2019].

#### Ieeexplore.ieee.org. (2019). Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights - IEEE Conference Publication. [online] Available at: https://ieeexplore.ieee.org/document/4470228 [Accessed 2 Sep. 2019].

#### Çano, E. and Morisio, M. (2017). Hybrid recommender systems: A systematic literature review. Intelligent Data Analysis, 21(6), pp.1487-1524.

#### Ieeexplore.ieee.org. (2019). Data Gathering System for Recommender System in Tourism - IEEE Conference Publication. [online] Available at: https://ieeexplore.ieee.org/document/7350670 [Accessed 2 Sep. 2019].

#### Ieeexplore.ieee.org. (2019). Collaborative Filtering for Implicit Feedback Datasets - IEEE Conference Publication. [online] Available at: https://ieeexplore.ieee.org/document/4781121 [Accessed 2 Sep. 2019].

#### Medium. (2019). Recommender System. [online] Available at: https://towardsdatascience.com/recommender-system-a1e4595fc0f0 [Accessed 2 Sep. 2019].

#### Pdfs.semanticscholar.org. (2019). [online] Available at: https://pdfs.semanticscholar.org/40e1/53460564ed0dbb2523394438c7a1172ca9dc.pdf [Accessed 2 Sep. 2019].

#### Medium. (2019). Introduction to recommender systems. [online] Available at: https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada [Accessed 2 Sep. 2019].

#### Ieeexplore.ieee.org. (2019). A unified approach of factor models and neighbor based methods for large recommender systems - IEEE Conference Publication. [online] Available at: https://ieeexplore.ieee.org/document/4664342 [Accessed 2 Sep. 2019].