Airbnb recommendation system using Aspect-based sentiment analysis: hybrid approach

Manav Mishra
STUDENT NUMBER: 2038836

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

Supervisor
dr. Afra Alishahi
Second Reader
dr. Boris Cule

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
May 2021

Preface

Dear reader, I would like to thank you for taking your time to go through my thesis about Airbnb recommendation system using Aspect-based sentiment analysis. I hope this thesis will provide you with proper knowledge about using different methods used in text analysis. Making this preface as a medium, I would like to send my kind regards and thanks to my supervisor dr. Afra Alishahi for guiding me and helping me to go in right direction through our meetings.

I would also like to extend my thanks to my mother, father and my girlfriend to support me in the difficult time when being far away from them in different country.

I hope you will enjoy reading this thesis as much as I enjoyed writing and developing the program on the topic,

Manav Mishra

Airbnb recommendation system using Aspect-based sentiment analysis: hybrid approach

Mishra

This paper will focus on conducting an Aspect based Sentiment Analysis (ABSA) of visitor reviews/feedback on the different room types they stayed via Airbnb.Therefore, the main aim of this thesis is to use customers' emotional expressions as the basis of their experience to extract the aspects and their related sentiments for the comments posted predict and then provide a personalized recommendation of the best room present on Airbnb accommodation. The recommendation system is built using Content-based filtering and Collaborative filtering method. However, the best performing model resulted to be the weighted hybridized model. To measure the performance for this conclusion recall, precision, and f-1 score is used to evaluate the performance of the model, the evaluation metrics are calculated for 5 and 10 recommendations.

1. Introduction

Since the time of its launch, Airbnb has been revolutionizing the hotel industry by providing stays in different parts of the world. Due to this, there has been undeniable growth in the lodging industry (Luo 2018). One of the keys to a successful online business lies in e-WOM (Electronic- Word of Mouth). Over the past few years, e-WOM has become a substantial factor that influences the online booking of hotels, and many people find hotel reviews as important as the brand name or the price of the hotel (Belarmino and Koh 2018). In the past years, there has been much research with the aim to leverage the e-WOM to provide every user better options which really is relevant to them. One of the method to acheive this goal is by analyzing the sentiments of User Generated Comments (UGC) to find the context of their opinion. This thesis is designed in a two fold process, first step is to conduct an Aspect based Sentiment Analysis (ABSA) of the users that wrote their reviews/feedback on the place they stayed on Airbnb after successfully extracting the sentiments the second step will be to leverage the sentiments and create a personalized recommendation system.

Thousands of reviews are posted by people on Airbnb, most of the time it does not exhibit one idea about their experience, people post their mixed feelings about the experience during their stay. So, using the traditional sentiment analysis would not be feasible if we want to achieve better recommendations, as Sentiment analysis focuses on the overall sentiment associated with the text. However, it is unable to include other essential information such as the direction of sentiment within the text of a topic (Hoang, Bihorac, and Rouces 2019). The following example will explain the situation better, for instance, a user posted a review: "The place was good, but the price was expensive!". In this sentence, though the overall sentiment is mixed, it clearly presents the place in a positive way but also talks about the price in a negative way. Diving further deeper into the aspects, there are two types of aspects in ABSA, namely, explicit &

implicit. In explicit aspect type, there is a clear mention of aspects that are opinionated, whereas in implicit the aspect opinionated is not clearly mentioned (Kushwaha and Chaudhary 2017). For example, an explicit aspect would be, "The host is awesome!" here it is clearly mentioning the host in a positive aspect, whereas an implicit aspect would be, "The room was quite messy!" here it is mentioning the opinion indirectly. Being able to classify these aspects in the reviews would make the recommendation without losing the hidden context in the review and thus able to give the user a more precise recommendation of a stay on Airbnb. Therefore, it backs up the significance of the thesis topic in accordance with practical relevance.

There have been several kinds of research in the field of Hotel recommendation which predicts the recommendation of hotels based on either, the past reviews of users, location, or combined use of check-in, check-out, geographical, temporal, and latent features and using methods such as random forests, customer-hotel bipartite network, SGD classifier, XG Boost and Naive Bayes (eg: Cao et al. (2020); Luo (2018)). However, this paper will only focus on reviews for Aspect Based Sentiment Analysis (ABSA) and the listings provided in the dataset.

This paper intends to use customers' emotional expressions as the basis of their experience to predict and recommend the best Airbnb accommodation using a hybrid model. Before, developing the hybrid model, this paper will first focus on Collaborative filtering (CF) and Content-based (CB) filtering recommendation systems to analyze the performance of model based on features extracted. This thesis thus, contributes in both theoretical and scientific aspects. The theoretical aspect of contribution is that, it will help people understand the working mechanism of the Machine learning algorithms that is being used by several companies in order to provide better user experience, the scientific contribution will be that most of the past researches for recommendation systems on Airbnb used methods such as Deep learning, Decision tree, as discussed earlier but this paper will try to implement Content-based and Collaborative filtering on the Airbnb dataset with the aim to discover more insights about user behaviour.

Therefore this paper would like to address two research questions. Since the second research question could be diversified therefore, it is broken down into two sub questions with the be more research specific. This thesis would try to investigate for the Research questions and the sub-question and try to answer them, the research question is as follows:

RQ 1. When reviewing the stays on Airbnb which topic or set of topics are found to be of higher significance to the reviewer based on ABSA on the past reviews?

RQ 2. Based on the users' most common choice of topic(s) for stays, what are the individual recommendation models that could be used to generate a personalized recommendation for each user on Airbnb?

SQ.2.1 How do the individual recommendation model perform on the Airbnb data?

SQ.2.2 Which is the most optimal model to use for the recommendation system?

2. Related Works

2.1 Aspect-Based Sentiment Analysis

The basic idea behind developing an Airbnb room/hotel recommendation system is To provide the best option to the user out of several choices so that the user gets to choose something which matches his/her preference. This does not only save the extra time of searching for the best option available, but also helps users to explore new places to stay, which they would not know about. An ABSA method can be used to analyze large

amount of unstructured text to extract fine-grained information that is not possible to get just from the user ratings that are available on different platforms (Pontiki et al. 2016) .

In the past, different methods have been used to analyze the sentiments of the reviews or comments posted by users. Serrano et al. (2021) used sentiment analysis for recommending green Airbnb users that posted reviews for Eco-friendly stays. They split the task in two steps, the first step was to identify and filter out the prominent themes that represents the green stays that were posted by the users. To achieve this they the Leximancer software to extract a conceptual map with latent semantic dimensions that identifies and displays the terms' hidden relationships with the appropriate labeled themes. The second step then was to perform an analysis of the emotions, which was achieved by finding out the dominant emotions in the review comments through text mining. Finally they implemented preference analysis using LDA, which is a popularly utilized method that helps in the identifying the latent aspects in a large dataset of review comments.

Mowlaei, Saniee Abadeh, and Keshavarz (2020) used Aspect based sentiment analysis using Adaptive Aspect based lexicons. In this method, they proposed Aspect-based Frequency-based Sentiment Analysis, an extension of their own method in which an adaptive lexicon was developed to update the aspects of the sentiments in the text without human intervention. The result was that, in dynamic lexicon generation, the context of the training dataset affected the performance in target datasets; in other words, the similarity of the respective contexts of training and target datasets improved the performance of dynamic lexicon generation.

Ekawati and Khodra (2017) used six step method to extract the aspect terms and calculate the polarities of the result. The first step was to pre-process the text and remove any inconsistencies from the texts. The next step was to extract the aspect terms, and to achieve this a CRF model was used. The resulting labels were then used to extract the features in the form of POS tags and headwords generated by using SyntaxNet. The third step involved categorizing the extracted aspect terms, this was a two fold process. The first process was to add pre-prcoessed sentences before the aspected terms and then they used multi-label binary relevance classifier with MaxEnt algorithm for categorizing the aspects. As a result, the classifier labels each sentence in the corpus as True if it has certain categories; otherwise, it labels the sentence as False. For the fourth step of sentiment classification, they used the same approach as they previously used for aspect classification, in sentence classification they created labels each category as positive and negative and in the next step for the assigning the sentiment polarities they used classifier for food and place category. The next step was to generate an opinion structure and they used CBOW model for this purpose. The aim of using the CBOW model was to find similarity between extracted aspect with seed words for each category. Category that had maximum similarity score was paired with the extracted aspect. The last step was to calculate polarities for each extracted aspects in the step five, for this purpose they used a formula which is as follows:

$$Rating = \left(\frac{P}{P+N} * 4\right) + 1$$

Where, P is the number of positive aspects and N is the number of negative aspects in a particular User generated review.

2.2 Recommendation System

Developing a better recommendation system is a basic problem for many internet companies where new content is published everyday, there exist no generalized solution for recommendation as the results needs to be personalised according to people's behavior while surfing and utilising the content on the web (Grbovic and Cheng 2018).

Kaya (2020) tried to approach the recommendation system as a link prediction problem by mapping the stays in hotels to a bipartite customer-hotel interaction network. In this method, the network structure was used to capture detailed information on the relationship between customers and hotels and then predict by utilizing the network structure through supervised learning. Moreover, in this method, the main focus was on the users' location and making suggestions on how users in a certain geographical region chose hotels.

Grbovic and Cheng (2018) proposed two distinct approaches, i.e. listing embeddings for short-term real-time personalization and user-type and listing type embeddings for long term personalization, respectively for the task of recommending and ranking a listing in search at Airbnb. Since, particularly in the case of Airbnb several new listings are updated on the website. This leads to the cold-start problem, to tackle this problem they proposed a method in which it was assumed that the new listings have similar preferences as other listings based on three features, geographically closeness of the new listings with the listings already present in the train-dataset, same room type, and same rent price. To achieve this, K-nearest neighbours method was introduced as this successfully grouped the new listings with less historical preferences data with the listings having similar features.

Next, they calculated the mean vector using 3 embeddings of identified listings to form the new listing embedding. Using this technique they were able to cover more than 98 percent of new listings. For the purpose of search ranking and recommendation different factors such as user clicks, logged in user sessions, host rejections were used. For more personalized results their proposed search rank model used cosine similarity to find the most relevant listing which incorporated the use of skip-gram models for listing_type and user_type. Thus, the model produced the recommendations based on type of embeddings.

Haldar et al. (2019) used deep learning for search ranking of listings at Airbnb.The approach used feature engineering and feature distribution. In the feature engineering process for the NNs, they focused on Feature normalization and Feature distribution. Feature normalization was opted to use because the loss saturated in the middle of training and additional steps and had no effect. Thus, to avoid this problem it was ensured that all features are restricted to a small range of values, with the bulk of the distribution in the -1, 1 interval and the median was mapped to 0. There were two types of transformations performed:

- If feature distribution resembled a normal distribution,

$$(feature\_val - \mu)/\sigma$$

  where, μ is the feature mean and σ is the standard deviation of the features generated.

- If the feature distribution looked closer to power law distribution,

$$log(\frac{1 + feature\_val}{1 + median})$$

The next step was to make the distribution of the features smooth, as this helped in spotting the bugs when the train dataset was huge and also the model was generalizable. To scale up the performance of the model hyperparameters tuning was done, this included the tuning of learning rate, different batch-size, and use of dropout layer in the final NN model. The final model chosen Deep Neural Network, the team reached to this conclusion after comparing the performances between Simple NN, LambdaRank NN, Decision Tree, and Deep NN as Deep NN outperformed the other models.

Tang and Sangani (2015) in their research used an interesting approach to recommend Airbnb stays to the user based on the price and the neighbourhood. They used the dataset for San Fransisco which is available on Insider Airbnb website. Since the dataset available is large, so in order to compensate for high computation power, the listings that belonged to neighborhoods containing fewer than 70 listings were excluded. This resulted to have 6764 listings and 27 neighbourhoods thus, this resulted in reducing the training time significantly. They used listing information, bag of words, text sentiments, and to extract visual features from the images OpenCV was used. The features were fed to the Support Vector Machine of the sklearn SVM package it was observed that the model performed with highest accuracy of 42 percent using the bag of words text features.

After carefully reading the past researches that were performed on the Airbnb recommendation system, this paper is aiming to create a more personalized recommendation system for the users that would be able to make the recommendations based on the aspects of the comments related to the listings and its popularity. Since, most of the previous researches focused using K-NN, Neural networks, Decision trees to classify and recommend. Therefore, through this paper it is attempted to make recommendations based on User Generated Comments (UGC) and use popular recommendation methods such as Content-based and Collaborative Filtering recommendations. The Method/Models section of this research paper discusses about the approach in detailed manner.

3. Experimental Setup

3.1 Dataset Description

The dataset used for the aspect-based sentiment analysis and building the recommendation system is downloaded from Insider Airbnb. It is a non-commercial website that scrapes and stores the data from Airbnb enabling users to download and visualize how different Airbnb apartments are used around the different cities of the world. Among data for different cities and countries available, the Amsterdam dataset is used for the analysis and for building the recommendation system for this thesis. The scraped data is stored in CSV format, a CSV file is a delimited text file that uses commas to separate the values stored. Each row in a CSV format contains a complete record for that row.

The dataset used for the analysis and building up the recommendation model is named reviews.csv on the website. This file used in this thesis was uploaded on February 8, 2021, it holds the information of all Airbnb listings in Amsterdam city from March 30, 2009, up to January 21, 2021. The downloaded CSV file contains 6 columns

namely, listing_id, id, date, reviewer_id, reviewer_name, comments. The column listing_id stores the unique id for each Airbnb stay in integer format, the listing_ids are unique for each stay. The column id contains a unique identification number for each row which can be used to retrieve particular information about a listing or a reviewer and the date column contains the date when the review was posted on the Airbnb platform. The column reviewer_id contains a unique identification number associated with each reviewer whose names are stored in the name column. The comments column contains the reviews dropped by a user after completing their period of stay in that particular stay type. **Figure 1** shows the raw data downloaded from the website.

	listing_id	id	date	reviewer_id	reviewer_name	comments
0	2818	1191	2009-03-30	10952	Lam	Daniel is really cool. The place was nice and ...
1	2818	1771	2009-04-24	12798	Alice	Daniel is the most amazing host! His place is ...
2	2818	1989	2009-05-03	11869	Natalja	We had such a great time in Amsterdam. Daniel ...
3	2818	2797	2009-05-18	14064	Enrique	Very professional operation. Room is very clea...
4	2818	3151	2009-05-25	17977	Sherwin	Daniel is highly recommended. He provided all...
...	...	...	...	...	...	...
441269	47467526	725605314	2021-01-20	241723501	Judith	Me and my husband stayed at Bell's Boutique ho...
441270	47467526	726768392	2021-01-25	128667624	Georgina	Had a very pleasant stay. The chaps were so lo...
441271	47854439	727618857	2021-01-30	226585219	Madison	Awesome, cozy spot with friendly staff made m...
441272	47882460	727955918	2021-01-31	386238081	Jordan	Bon accueil, bon emplacement etc..
441273	47882460	727963645	2021-01-31	386400953	Enhar	Schönes Zimmer und gastfreundlich

441274 rows × 6 columns

Figure 1
Raw dataset before performing EDA and Text pre-processing

Figure 2 shows how the reviews are posted after a reviewer post a review about a stay. This snapshot contains reviews, date of posting the review, first name of the reviewer, and ratings provided on different criteria. The first review in the picture is posted by me, after finishing my trip to Turkey in November 2020. Reading my comments gives an accurate idea of what aspect-based sentiment analysis, as the review states, *"... had quick responses to my queries. The only problem I had with the place was terrible wifi. Other than that everything was really great."* This comment by me shows that there were few aspects of the stay which were not satisfying for me and it would be helpful criteria for the next person who chooses this place to stay.

Figure 2
Review section of the Airbnb app viewed from iPhone IOS.

3.2 Data Pre-processing

The entire Pre-processing, data visualization, and model building is done using python programming language on Google Colaboratory. The raw data contains 441274 rows, the data was first analyzed to find out if there were any missing or NULL values in the dataset, 271 rows from comment columns were NULL and were removed in the data cleaning process. In the next phase of the data cleaning process, the column date, name, and id were dropped and stored in a new dataframe named cleaned_df. Due to limitations, and less knowledge of other languages only the comments written in the English language are retained and stored in a new pandas dataframe named cleaned_df. To filter out English language from the comments column fasttext model by (Joulin et al. 2016) was used, a pre-trained model lid.176.bin was downloaded. A new column named lang was created and each comment that were written in the English language

were represented by 'en' in the corresponding column. Non-English comments were then filtered out using the apply function on the column of the dataset.

3.3 Exploratory Data Analysis

An EDA on the dataset was performed to analyze as well as visualize the data, its structure, and its distribution.

3.3.1 Readability. The most important part of aspect-based sentiment analysis is to have sentences in a structured format, the better the grammatical structure of the sentences are, the better the result would be when calculating the sentiment intensity of each review posted. To observe the readability of the sentence, the Flesch readability score using the flesch_reading_ease function from the textstat library is used. The score generated by this function lies in the range between 0-100. Where the higher score represents the ease in reading a particular sentence. The formula used for calculating the score in the function is as follows:

$$SCORE = 206.835 - 1.015\left(\frac{Number\_of\_words}{Total\_sentences}\right) - 84.6\left(\frac{Syllables}{Total\_words}\right)$$

Figure 3 shows the Felsch score distribution for the comments posted by users in the dataset. It was observed that majority of the text were awarded with a good readability score. Thus, it was deduced that majority of English speakers were able to write their comments in a readable manner.

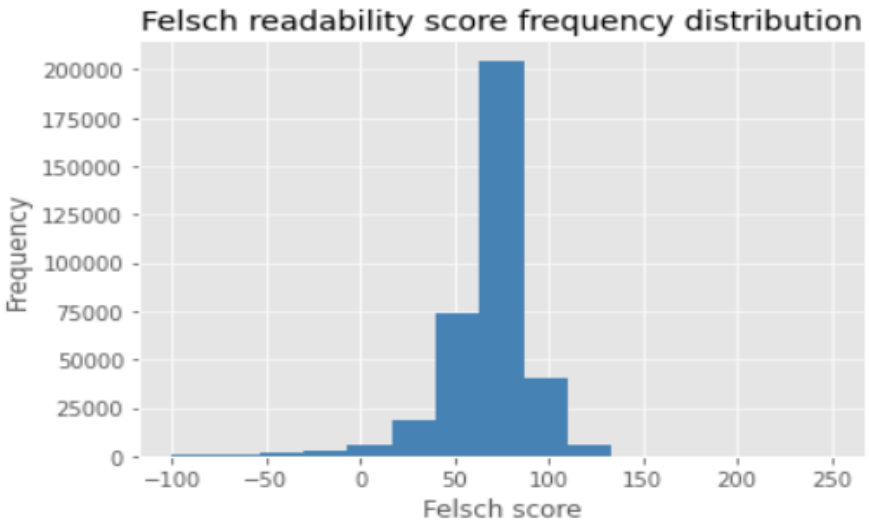

Figure 3
Felsch readbility score distribution

3.3.2 Length of comments. To extract the aspect from a comment or a review, it is important to have comments of appropriate length. Since, there are many people who posts comments which are of either one word or grammatically incorrect, therefore it

becomes important to filter-out the comments which are of shorter length. **Figure 4** shows the length of the comments posted by the users in the dataset.

	listing_id	reviewer_id	comments	comments_length	lang
0	2818	10952	Daniel is really cool. The place was nice and ...	250	en
1	2818	12798	Daniel is the most amazing host! His place is ...	334	en
2	2818	11869	We had such a great time in Amsterdam. Daniel ...	399	en
3	2818	14064	Very professional operation. Room is very clea...	203	en
4	2818	17977	Daniel is highly recommended. He provided all...	277	en
5	2818	20192	Daniel was a great host! He made everything so...	474	en
6	2818	23055	Daniele is an amazing host! He provided everyt...	312	en
7	2818	26343	You can´t have a nicer start in Amsterdam. Dan...	482	en
8	2818	40999	Daniel was a fantastic host. His place is calm...	205	en
9	2818	38623	Daniel was great. He couldn.t do enough for us...	281	en
10	2818	48138	Daniel has been more then a great host: it was...	529	en
11	2818	55661	Daniel's apartment and room was spotless. Dani...	419	en
12	2818	33284	Daniel was an exeptional host!! We only had a ...	517	en
13	2818	82918	No amount of praise for Daniel would be corny ...	666	en

Figure 4
Length of each comment or review posted by users.

3.3.3 Frequency of occurrence of each listing. The aim of this thesis is to develop a recommendation system, therefore one of the important task was to get the information about the number of time a listing occurred in the dataset.Extracting the information about frequency of occurrence gave the information about the popular listings in terms of booking time by different users (in this case, user comments are not included to decide the popularity of a listing). **Table 1** shows the frequency of occurrence of few listings in the entire dataset.

Table 1
Listing id and their corresponding frequency of occurrence in the dataset.

listing_id	Frequency
2818	278
20618	339
25428	5
27886	219
278871	336

3.3.4 Frequency of reviewers using Airbnb to book a stay. A better understanding of the data to develop the recommendation model would be the information of the number of times a person books a stay, as it is necessary to have insights about the user booking history to develop a personalized recommendation system. Therefore, the frequency of a user booking Airbnb helps to provide a better understanding of the data. **Table 2** shows the frequency of booking done by several users.

Table 2
reviewer id and number of times a listing was booked by them in the dataset.

reviewer_id	Number of times booked a listing
12574897	22
11318929	18
10146524	13
7647712	12
12900104	9

3.3.5 Wordcloud. In order to extract the aspect terms, it is important to know the words which is most frequently used by users. This step was a crucial step in visualizing the data as, it provided the insights to the top 100 words used by users in their comments. **Figure 5** shows the most frequent words used by different users in the form of word-cloud.The larger the font of a word, the more frequent it was used by different reviewer in their comments for a review posted.

Figure 5
Word cloud displaying the top 100 most common words used in reviews.

3.4 Feature Engineering

In the process of feature engineering, 9 unique features are extracted from the comments, in order to obtain the aspect-based sentiments of each comment. The 9 unique features extracted throughout the process of feature extraction are expanded_words. This method converts the contracted words in English to their regular form, such as **{couldn't = could not; didn't = did not; wouldn't = would not}**. To achieve this, python library contractions was imported and then function fix() was used, after the words were expanded, they were converted into sentences. The next step then, was to perform tokenization on expanded sentences and remove any special characters or punctuation present, for instance **{"The bedroom, kitchen, bathroom was dirty!" = [The], [bedroom], [kitchen], [bathroom], [was], [dirty]}** after performing this the tokens were converted to the lower case. After the tokenization was performed, the next step performed was to remove the punctuation signs **(, . ? ! / ' " ")** from the tokenized sentences. To remove the punctuation signs the string library was used.

To extract the aspect terms from the sentences, the feature engineering process did not stop at an early stage. After, the removal of the punctuation, the next step that was carried out was, stopwords removal. Sentences are formed by joining different words such as {and, the, when, or}. Such terms acts as noise while computing the sentiment and also it becomes difficult for the machine learning model to extract the aspect terms properly, therefore, to achieve this, nltk.corpus, a pre-trained pipeline from the stanford library was used. For instance, a sentence **"The hotel was quite expensive"**, when stopwords are removed using the nltk.corpus pipeline the output becomes like **"hotel quite expensive"**. Thus, it makes easier for the ML model to find the the Part of speech used in the sentences which is discussed briefly in the forthcoming paragraph.

In order to extract the aspect terms that would grammatically stand correct, it is first needed to understand the Part-of-speech used in the sentences. To achieve this, nltk.pos_tag was used. In order to perform a refined feature engineering to get better aspect terms, lemmatization was performed on the POS tagged words so that the words that were not present in their root form could be converted. For instance, words such as **{'cooked', 'cooking', 'cooks'}** are not in their root form, so applying lemmatization would convert them to the root form, which is **{'cook'}**. NLTK package was used to perform lemmatization, and Wordnetlemmatizer() was used to carry out the task. This was the final step, from this step the actual task of checking the dependency and extracting the relationships between two words was carried out. **Table 3** shows the different POS tags that can be visualized using the nltk.pos_tag function.

The next step, then carried out in the process was to visualize the dependency between the words. In order to visualize, spacy library was used, as shown in the **Figure 6**. The last and important step in the feature engineering process was to extract the aspect terms. The pair of terms that were tagged with Nouns (NN), Adjectives (JJ), Comparative Adjectives (JJR), CommonNouns (NNS), Adverbs (RB) were extracted.

In order to obtain the sentiment polarities of the UGCs, a lexicon-based and rule-based sentiment analysis tool introduced by (Hutto and Gilbert 2014). The output of the intensity is categorized into 4 groups namely: negative, positive, neutral, and compound, the compound score represents the sum of all the lexicon ratings standardized to [-1,1].

Table 3
Universal Part-of-Speech (POS) Tagset

Tag	Meaning	Example
ADJ	Adjective	new, good, high, special, big
ADV	Adverb	really, already, still, early, now
NN	Noun	year, home, costs, time, The Netherlands
PRON	Pronoun	he, their, her, its, my, I
Verb	Verb	is, say, told, given, playing, would
NUM	Number	twenty-four, fourth, 1997, 14:24
DET	Determiner	the, a, some, most, every, no
CONJ	Conjunction	and, or, but, if, while, although

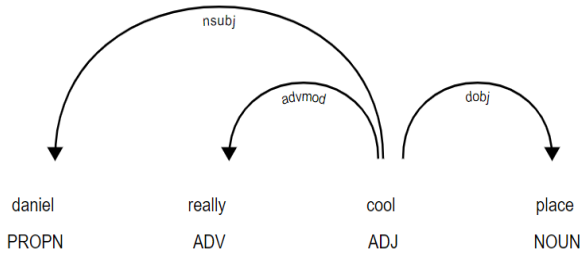

Figure 6
Dependency visualisation on a part of the review text using Displacy function. It shows that the **daniel** is being reffered to as really cool, as the arrow from cool goes back to *really* and *daniel*. Also, the reviewer mentioned the place where he/she must have stayed, and it was also mentioned as a *cool* place. Therefore, an arrow from *cool* is linked to the word *place*.

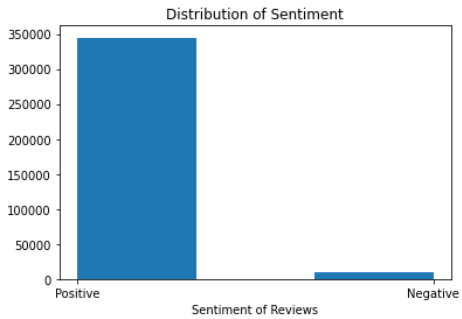

Figure 7
Positive and negative sentiment score distribution after calculating the sentiment scores for each reviewers.

4. Method / Models

Extracting the aspect terms and calculating their polarity was the step 1 task. After successfully completing step 1 task, the next task in this paper was to design a recommendation model. The forthcoming paragraphs briefly describe the steps that were taken into the account while developing the respective models.

4.1 Pre-model development phase

After visualising the distribution of the sentiments, the data was first split into train and test set, weights were assigned to the positive and the negative comments this was done because Agresti (2003) mentioned that a weight is given to a data point to assign it a lighter, or heavier, importance in a group. Therefore, it was aimed that negative comments were to be assigned less importance than the positive comments. The formula used to calculate the weights is as follows:

$$w_{nc} = \left(\frac{\%\ of\ known\ number\ of\ negative\ comments}{\%\ of\ negative\ comments\ in\ dataset} \right)$$

$$w_{pc} = \left(\frac{\%\ of\ known\ number\ of\ positive\ comments}{\%\ of\ positive\ comments\ in\ dataset} \right)$$

the weight for negative comment (w_{nc}), obtained using the formula mentioned was 0.99 which was rounded off to 1 and for the positive comments (w_{pc}), the weight obtained was 99.9 and it was rounded off to 100. This was done so that the ratings could be returned as a whole integer for the ease in developing the algorithm further and the recommendation systems uses normalised weights instead. Since the data was highly imbalanced, therefore, only listings which are booked at least 3 times are selected. This helped to extract the users with historical choices available. The filtered data was stored in pandas dataframe named all_bookings. This data was then split into train and test set using sklearn train test split library function (Pedregosa et al. 2011). The data was randomly split in the ratio of 80% as train set and remaining 20% as test set.

As it was known beforehand that the original dataset was highly imbalanced, therefore in order to make the models generalised the train set minority classes of the train set was upsampled and made equal to the length of the majority class. When the data was upsampled and the imbalanced was handled, then the train set was passed to the model. To normalize the compound score of the polarities, the following formula was used:

$$label\_weight = log_2(1 + y)$$

where, y is the compound score for each review posted by the reviewer, and the resulting $label\_weight$ is the normalized compound score for each reviewer. Normalizing the scores was done to convert the negative compound score to a common positive scale range without distorting the difference in the range of the values. The forthcoming paragraphs describes the main models that were approached and fine tuned for making recommendations. The 4 models developed is as follows:

4.2 Baseline Approach

Popularity Model: A popular method used in building a recommendation system is the popularity model. The fundamental approach of this model is to recommend to a user an item that has not been previously used by that user. Therefore, the result of this model is more generalized and from the perspective of everyone. Thus, this model was chosen as a baseline model to compare the performance of the main model. **Table 4** shows the top 5 listings provided by the developed popularity model.

Table 4
Top 5 popular listings in the dataset.

listing_id	label_weight
68290	93.214961
68873	93.214961
2429334	73.240326
1601408	73.240316
12908561	73.140789

4.3 Content-based model

Content-based recommendation systems aim to suggest things that are close to those that a consumer has previously enjoyed. Indeed, a content-based recommender's basic process entails matching the attributes of a user profile, which stores preferences and interests, with the attributes of a content object (item), in order to suggest new interesting items to the user (Lops, de Gemmis, and Semeraro 2011). **Figure 8** shows the basic working structure for a Content-based recommendation system. To perform the content-based filtering method, a two step process was designed. The first step that was to create a tf-idf matrix for the User Generated Comments, the second step used cosine similarity to return the recommendations. The forthcoming subsections discusses about the methodology in detail.

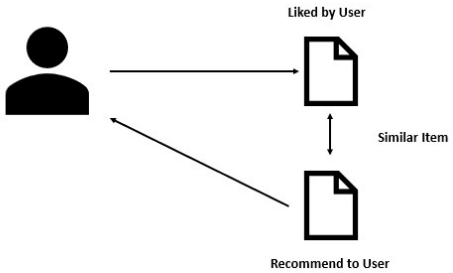

Figure 8
Item similar to user's previous choice is recommended to the user.

4.3.1 Term-Frequency*Inverse-Document Frequency (TF*IDF). The TF*IDF methodology originated from the IDF method proposed (Jones 2004), which was based on the heuristic intuition that a query word that appears in a large number of documents is not an useful discriminator and should be given less weight than one that appears in a small number of documents (Zhang, Yoshida, and Tang 2011). In this thesis TF-IDF weighting was used to gain insight into what makes individual users unique. Equation below is the classical formula of TF*IDF used for term weighting.

$$w_{ij} = tf_{ij} \times log(N/df_i)$$

where w_{ij} is the weight for term i in document j, N is the number of documents in the collection, tf_{ij} is the term frequency of term i in document j and dfi is the document frequency of term i in the collection.

TF-IDF values were calculated for all unique terms (1-grams) and the combinations of 2 sequential terms (2-grams) from the corpus using the above weighting equation and stored in an $n \times k$ matrix where n represents a row of users and each k represents a column of 1 or 2-gram. To create a personalized recommendations for each reviewers in the training set, their profiles were created and named as **reviewer_profile**. This profile of each reviewer was created by using the relevance score computed from tf-idf method for the posted reviews against the stays for which the user has not posted any comments for the listings within the training set, as shown in **Figure 9**. The value in each position

	token	relevance
0	space	0.316799
1	lovely	0.243399
2	furnished	0.231159
3	quiet	0.227424
4	stay	0.215086
5	would	0.202372
6	nicely	0.192062
7	day	0.183086
8	communication	0.181691
9	automate	0.176351
10	post	0.174003
11	cancel	0.173667
12	reservation	0.172388
13	check	0.167198
14	kind	0.151332

Figure 9
Top-15 Relevance score for different features extracted for reviewer with reviewer_id 52998263 using tfi-idf method.

represents how relevant is a token (unigram or bigram) for the reviewer. The relevance score is computed against the reviews of the airbnb stays for which user has not posted any reviews. The forthcoming subsection discusses about the cosine similarity and how it was used in developing the Content-based recommendation system in this thesis.

4.3.2 Cosine Similarity. A similarity measure reflects the degree of closeness between 2 articles using a single numeric value (Huang et al.). Cosine similarity was chosen because, it is easy to calculate and interpret and is a popular method used in the calculating the similarity score (Dhillon and Modha 2001). Cosine similarity returns a value between 0 and 1, where 2 documents with a similarity value of 1 or closer to 1 are regarded as identical, and a value of 0 or close to 0 implies no similarity between the documents (Huang et al.). The formula used to calculate the cosine-similarity is as follows:

$$cosine(x, y) = \sum_{i=1}^{n} (x_i, y_i) / \sum_{i=1}^{n} (x_i^2) * \sum_{i=1}^{n} (y_i^2)$$

where, x is user-profiles in the vector form and y is the feature vector corresponding to every Airbnb stay which was not reviewed by the user, generated using the tf*idf methodology. Finally, the top-5 and top-10 similar stays were recommended for a particular user based on the cosine-similarity that was generated. However, a disadvantage of Content-based recommendation system is facing cold start problem, this occurs when there is no historical data available about a user's preferences (Lops, de Gemmis, and Semeraro 2011). Recommendation system developed for hotel industries faces such problem, as it is very rare for a person to book a same place again while visiting a place (Grbovic and Cheng 2018). The forthcoming sub-section briefly discusses the model developed to tackle the problem.

4.4 Collaborative filtering model

To overcome the cold-start problem, Collaborative filtering model was designed as there were many users with no history of booking. The basic working algorithm for this model is that to compare a customer's preferences to those of other customers before making recommendations. It's a personalized algorithm, thus, providing recommendations for each user based on their historical choices (Chaudhary and Anupama 2020). **Figure 10** shows the working of Collaborative filtering model.

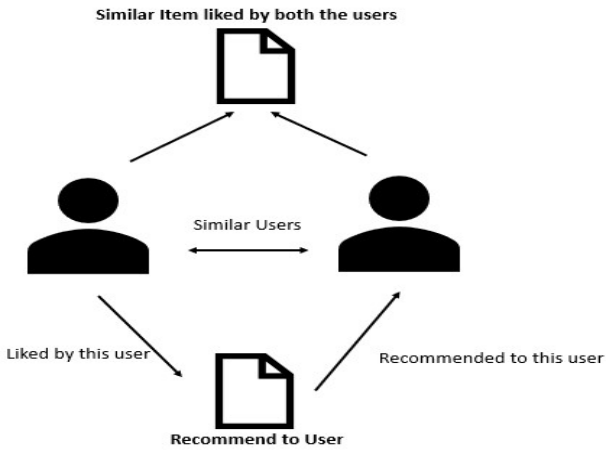

Figure 10
Item similar to user's previous choice is recommended to the user having mutual choices.

There are two types of Collaborative filtering methods available, **Figure 11** shows the types of methods available with their short description. Before the model was

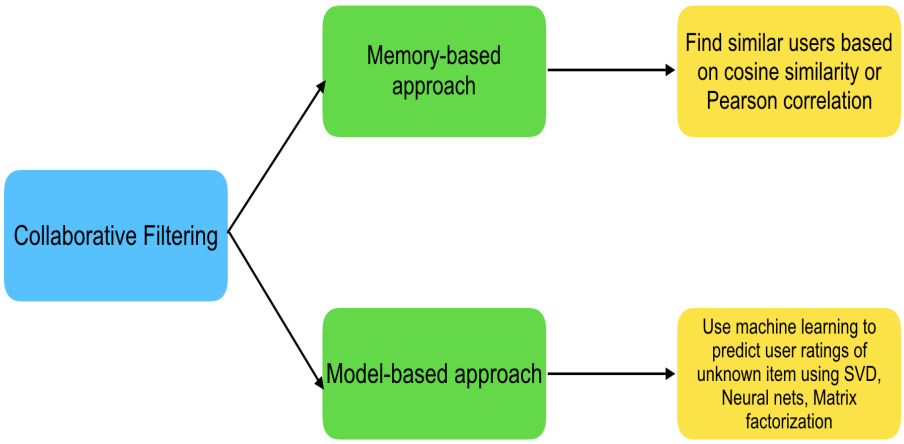

Figure 11
Short description for the prominent types of Collaborative Filtering method. Model-based approach was used in this thesis.

developed, LDA method was utilised on the reviewers' comments. The forthcoming subsection describes the implementation of the LDA method and its relevance for the thesis.

4.4.1 LDA:. Blei, Ng, and Jordan (2001) introduced the LDA method, in their research paper they defined LDA as a generative probabilistic model of a corpus, where corpus was defined as a collection of more than one document. The basic mechanism for a working LDA model is that documents are represented as random mixtures over latent topics, where each topic is characterized by distribution over words. Blei, Ng, and Jordan (2001) further stated an equation that illustrates the working method of LDA

to obtain the probability of the corpus. For given parameters α, β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w the probability of a corpus D (collection of M documents) is obtained by:

$$p(D|\alpha,\beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha)(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d)p(w_{dn}|z_{dn},\beta)d\theta_d)$$

The parameters α and β used in the equation are corpus-level parameters, they are assumed to be sampled once in the process of generating a corpus. The variables θ_d are document-level variables, sampled once per document. Finally, the variables z_{dn} and w_{dn} are word-level variables and are sampled once for each word in each document. Hence, the probability of a corpus is obtained by taking dot product of marginal probabilities of single documents.

Often, LDA model is mistaken with the Simple Drichlet-multinomial clustering model, thus it is important to distinguish between both the models. A classical clustering model typically involves a two-level model in which a Dirichlet is sampled once for a corpus, a multinomial clustering variable is selected once for each document in the corpus, and a set of words are selected for the document conditional on the cluster variable. As with many clustering models, such a model restricts a document to being associated with a single topic. On the counterpart, LDA, involves three levels (shown in **Figure 12**), and particularly the topic node is sampled repeatedly within the document. Therefore, with the LDA model, documents can be associated with multiple topics instead of one. The forthcoming paragraph explains how the topics were formed.

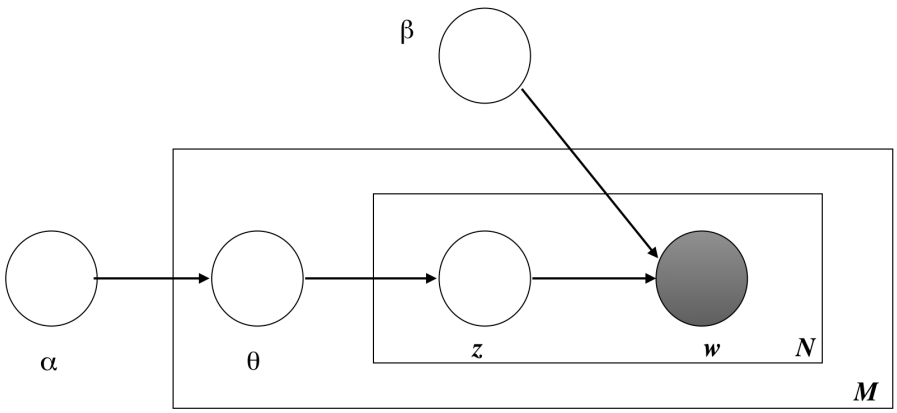

Figure 12
Graphical model representation of LDA. The boxes are "plates" representing replicates. The outer plate (M) represents documents, while the inner plate (N) represents the repeated choice of topics and words within a document.

To build the LDA topic model using LdaModel(), a corpus and a dictionary was created. Gensim module was used to produce dictionary, it creates a unique id (**word_id**) for each word in the document. The produced corpus is in the form of mapping of (word_id, word_frequency). For example, (0, 1) implies, word id 0 occurs once in the first document. Then a dictionary was created on the user generated reviews, using the module corpora.dictionary. In addition to the corpus and dictionary, the number of topics were set to 10 in the ldamodel(), as the aim was to cover the top-10 topics that

reviewers were interested in. Apart from that, alpha and eta, the hyperparameters that affect sparsity of the topics were both set to defaults to 1. The number of topics was decided by computing the coherence score from one topics to ten topics, as illustrated in **Figure 13** Mifrah and Benlahmar (2020), in their research mentioned that c_v measure

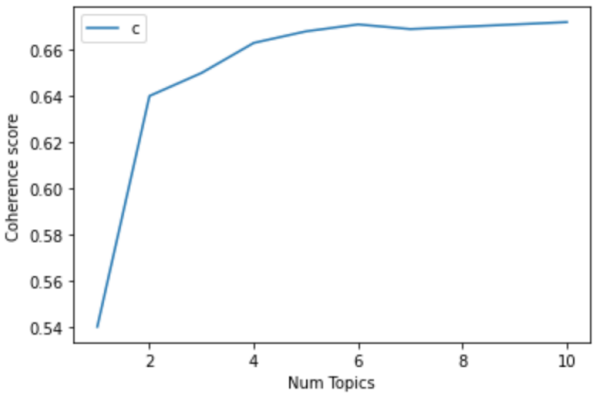

Figure 13
Graphical representation of choosing the optimal number of LDA topics.

is a prevalent method to calculate the coherence of topics. It is based on a sliding window, one-set segmentation of the top words and a measure that uses normalized point-wise mutual information (NPMI) and the cosine similarity. Therefore, this method was used to calculate the coherence score as shown in the **Figure 13**. To achieve this *compute_coherence_values()* method was used, and the dictionary and corpus that was created to generate the topics, were passed as the parameters to calculate the coherence scores.

Finally, the lda model generates the output in the form of words and their respective weights as a contribution for a particular topic. **Figure 14** shows the topics generated along with the words dedicated for each topics. To make the output look cleaner, top-10 words were chosen to be displayed. For instance, in Topic 0 displayed in **Figure 14** it is evident that word ***'stay'*** holds ***0.049%*** weight, word ***'great'*** holds ***0.046%*** weight, word ***'apartment'*** holds ***0.032%*** weight, and so on. It is evident that the **Topic 0** as displayed in the **Figure 14** is collection of words such as **stay, great, apartment,size,..** and so on, therefore the **Topic 0** was renamed as **Topic Stay**. In the similar fashion other topics were renamed as well. By forming the topics in the fashion described, it allowed to understand the most common and important discussion that the users did about the Airbnb Listings, thus the prior information about users' preferred topic of discussion was leveraged to develop the recommendation model, also it allowed to study the different words users used to describe their experience and explain the whereabouts of their stay in a brief manner.

LDA method was incorporated in this research so that a clear understanding could be obtained about the choices, and aspects of a reviewer for a particular stay at Airbnb by deducing the reviews into a group of topics. This helped to understand aspects that reviewers found as a positive or negative for a stay. There was two motives behind the creation of the topics using the LDA model. First being, a medium to visualize the topics using LDAVis, and finding the optimal topics and exploring the choices of different users for Airbnb listings, and second motive being using these topics as the feature matrix in LDA-SVD model to generate recommendation using Model-based CF method.

```
[(0,
  '0.049*"stay" + 0.046*"great" + 0.044*"place" + 0.032*"apartment" + 0.031*"nice" + 0.030*"host" + 0.029*"location" + 0.025*"clean" + 0.024*"recommend" + 0.024*"good"'),
 (1,
  '0.211*"center" + 0.085*"bike" + 0.080*"far" + 0.044*"town" + 0.042*"exactly" + 0.034*"rent" + 0.033*"ride" + 0.027*"thoughtful" + 0.021*"gorgeous" + 0.020*"second"'),
 (2,
  '0.086*"get" + 0.037*"staff" + 0.032*"provide" + 0.029*"stop" + 0.028*"amenity" + 0.027*"boat" + 0.026*"bus" + 0.022*"breakfast" + 0.020*"access" + 0.020*"food"'),
 (3,
  '0.155*"day" + 0.089*"spacious" + 0.069*"arrival" + 0.050*"reservation" + 0.049*"cancel" + 0.045*"leave" + 0.038*"arrive" + 0.024*"let" + 0.022*"late" + 0.021*"totally"'),
 (4,
  '0.069*"room" + 0.029*"bed" + 0.020*"little" + 0.019*"night" + 0.017*"people" + 0.017*"could" + 0.016*"small" + 0.016*"bathroom" + 0.015*"big" + 0.015*"cozy"'),
 (5,
  '0.463*"perfect" + 0.060*"friend" + 0.050*"family" + 0.043*"ever" + 0.041*"park" + 0.036*"happy" + 0.034*"owner" + 0.030*"worth" + 0.029*"nicely" + 0.018*"machine"'),
 (6,
  '0.086*"cosy" + 0.083*"sure" + 0.079*"responsive" + 0.073*"hospitality" + 0.068*"weekend" + 0.060*"accommodation" + 0.054*"awesome" + 0.035*"bring" + 0.034*"market" + 0.032*"care"'),
 (7,
  '0.040*"respond" + 0.033*"value" + 0.032*"quickly" + 0.032*"meet" + 0.031*"still" + 0.028*"money" + 0.024*"corner" + 0.024*"message" + 0.024*"service" + 0.021*"hope"'),
 (8,
  '0.177*"flat" + 0.066*"stair" + 0.041*"appartment" + 0.036*"steep" + 0.035*"appartement" + 0.035*"terrace" + 0.031*"tidy" + 0.031*"keep" + 0.027*"list" + 0.025*"facility"'),
 (9,
  '0.285*"super" + 0.056*"cute" + 0.054*"part" + 0.035*"guy" + 0.033*"flexible" + 0.027*"due" + 0.025*"welcoming" + 0.022*"suggestion" + 0.020*"surround" + 0.017*"young"')]
```

Figure 14
Topics generated using ldamodel(). Numbers before each words shows their weight of contribution for that particular topic.

The results of using the LDA method is included in the Result section **Figure 16, and Figure 17**, later in the discussion section the result is described.

After developing the LDA model and extracting the topics, the LDA feature matrix was developed, this feature matrix was then utilised in the SVD model to predict the user-ratings. Since, in the dataset, the user-ratings was not available therefore, the compound score generated for each reviews in the Aspect-based sentiment analysis process. Furthermore, the forthcoming sub-section describes the use of LDA-SVD together for generating the recommendations using Collaborative Filtering model.

4.4.2 Model-Based Filtering:. These models use various machine learning such as Singular value decomposition, Markov decision process, clustering models, Bayesian networks, Dirichlet allocation, and data mining algorithms to predict what rating a user would offer to things he has not used or rated yet in this form of the filtering process. To improve the accuracy and efficiency of the model, these algorithms need dimensionality reduction to minimize the number of dimensions of its features. (Chaudhary and Anupama 2020). Therefore, for this thesis the second model developed used the Singular Value Decomposition method as the Matrix Factorization method to predict the user-ratings for the listings that were not booked earlier. The model developed incorporated a two fold step in order to predict the ratings by reviewers for listings. The following point describes the steps taken:

- **Decomposition of the feature matrix generated from LDA modeling using SVD (LDA-SVD Model):** The approach used in this thesis was to associate the label_weight which was obtained from the compound score of overall review for each reviewer with the topics that best represents the user's opinion for that particular Airbnb listing. Therefore using LDA method it was aimed to represent these documents(Collection of reviews from the training set) as being composed of topics and use that as a base in developing the term by sentence matrix for SVD. Then SVD was implemented to extract the most sentences that best represent these topics by obtaining the most orthogonal representations. After performing this step, the next crucial stage that was performed was to predict the label_weights (named as R in the equation), the mathematical formula is

as follows:

$$R = U.\Sigma.V^T$$

where,

$$U \in IR^{n \times r},\ \Sigma \in IR^{r \times r},\ V \in IR^{r \times m}$$

Inspired by the approach of Koren, Bell, and Volinsky (2009), a sparse matrix was created from training dataset with the values as label_weight, the sparse matrix generated was decomposed in n x r User latent feature matrix, r x r diagonal matrix containing singular values of LDA generated r x m feature matrix and number of factors were set to 15. The main aim was to make the resultant matrix, able to capture most of the variance from the original matrix. Therefore, the feature matrix generated from LDA topic modelling was decomposed using SVD method, as it is evident that SVD helps to capture the most orthogonal representations of topics in the documents (Arora and Ravindran 2008). For this thesis, instead of predicting ratings, it was focused to predict the label_weights that was assigned to each listings in the Pre-model development phase.

- **Predict label weights (R'):** After decomposing the sparse matrix, the second step taken was to generate the predicted label weights by taking the dot product of reviewers, and the listings matrix such that,

$$R' \approx R$$

and, R' matrix was constructed by sorting the values of k x k diagonal matrix by decreasing absolute value and truncating this matrix to first k dimensions (k singular values). **Figure 15** shows the calculation for R'(Predicted label weights) and the formula used to calculate R' is as follows:

$$R' = \tilde{U}_{n \times k}.\tilde{\Sigma}_{k \times k}.\tilde{V}^T_{k \times m}$$

.

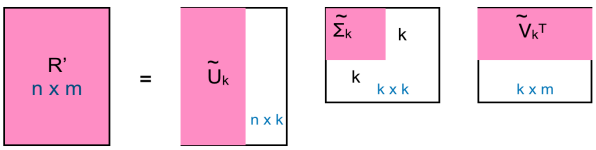

Figure 15
Construction of the predicted label weights by reducing the k dimensions of the diagonal matrix.

Once, the label weights were predicted, the next step taken was to normalize the matrix. The normalization was done to reduce the cost of errors. The calculation of

normalization performed is as follows:

$$normalized\ \ R' = \frac{R' - min(R')}{R' - max(R')}$$

After this step, the final step taken was, to pass the normalized predicted label weights to the collaborative filtering model and generate recommendations in terms of Airbnb listings for reviewers and return the top-5 and top-10 results based on the predicted normalized ratings.

4.5 Hybrid Model:

Hybrid filtering incorporates various recommendation approaches to improve system optimization and prevent some of the drawbacks and issues that comes with pure recommendation systems. Hybrid approaches are based on the assumption that a mixture of algorithms can provide more reliable and efficient suggestions than a single algorithm, since the drawbacks of one algorithm can be solved by another algorithm (F.O. Isinkaye 2015). With the aim of achieving better performance, weighted hybridization method was incorporated. The forthcoming sub-section describes the weighted hybridization methodology used in the model development process.

4.5.1 Weighted Hybridization:. Weighted hybridization integrates the outcomes of various recommender systems to produce a recommendation list or forecast by using a linear formula to integrate the scores from each of the techniques in use. A weighted hybrid model has the advantage of using all of the recommender system's strengths in a transparent manner throughout the recommendation process (F.O. Isinkaye 2015).

Thus, the hybridized model for this thesis was developed in 3 fold steps. The following points describes each step taken briefly:

1. **Ensemble weight initialization for the Content-based and Collaborative filtering methods:**
   As a pre-requisite of weighted hybridization model, ensemble weights were assigned to the two models. The weights chosen for each model was based on the individual performances of both the models (shown in **Table 5**). Therefore, for the Content-based recommendation model a weight of 20.0 was assigned and for the Collaborative-filtering model a weight of was 500.0 assigned. This was because the Collaborative filtering method outperformed the Content-based method (as shown in **Table 5**). The ensemble weights were the hyperparameters, therefore the weight for each model was chosen randomly and the final weight combination that was retained provided better result.
2. **Merging the recommendations of the two models:**
   After, initializing the weights, the next step performed was to merge the recommendations generated by each model and obtain a single unified dataframe of recommendations. To merge the dataframe, outer join was performed and the join was done on the reviewer_id column. The outer join merges the rows from the left and the right dataframe with NaNs where there are no matched join variables (Lynn 2021). Since, the outer join

also returned NaNs, all those entries which were NaNs were filled with 0.0 in order to ignore execution errors.

3. **Calculating the Hybrid recommendation strength:**
The final step in developing the hybrid model was to calculate a hybrid recommendation strength, following formula was applied for the strength calculation:

$$Ensemble\_CB\_strength = CB\_strength \times ensemble\_weight\_CB \quad (1)$$

$$Ensemble\_CF\_strength = CF\_strength \times ensemble\_weight\_CF \quad (2)$$

Where, CB_strength is the weighted average of the produced recommendation ratings (in the case of this thesis, the recommendation ratings was the compound score generated using the Aspect-based sentiment analysis), similarly the CF_Strength was computed by calculating the weighted average of the normalized ratings score that was generated by the CF method as described earlier. The ensemble weights were used to give importance to better performing model among CB model and CF model and to achieve this, model-averaging was performed. Model-averaging is used when dealing with parameter uncertainty, the main idea of using it is when the model is trying to make predictive models some models works right for the prediction point while some models tend to overestimate or underestimate. By averaging over all the models, it is focused to even out the overestimation and underestimation (Steel 2017). To accomplish this task, RMSE (root mean squared error) was used as the model fitness. Then it was averaged using the following formula as stated below:

$$ensemble\_weight_i = \frac{\sum_i \frac{y_i}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2}}$$

in the formula mentioned, it was aimed to weight the prediction y for model i (where i is Content-based and Collaborative-filtering model) with the inverse of the squared RMSE (σ^2) for model i. After calculating the ensemble weights for each model and calculating the Ensemble strength for both CB and CF model, a Hybrid weight was calculated as described in the equation below:
Substituting Eq.(1) and Eq.(2) we get;

$$Hybrid\_weight = Ensemble\_CB\_strength + Ensemble\_CB\_strength$$

After, the calculation of the associated hybrid strength for each listings, the dataframe was arranged in ascending order, to get the highest strength on the top. Then the final step carried out was to return the recommendations with top-5 and top-10 hybrid weights.

5. Evaluation:

5.1 Workflow of Evaluation Model:

Evaluation is important for machine learning Models, because it allows to compare objectively different algorithms. Therefore, the dataset was split into two parts - training (80%) and testing dataset (20%). The model was trained on a majority of training dataset , the recommendations were predicted for the reviewers in the testing dataset. **Figure 14** shows the workflow of an evaluation metrics. The data from train set was first used to build the recommendation system, after properly tuning the recommendation system, the test set was used to generate the recommendations. These recommendations were passed to the evaluation metrics to obtain the performance of each models. In this thesis standard classification accuracy metrics namely precision, recall and f-measure was used. These values were measured per user by retrieving K recommendations (where K was 5 and 10) and calculate their average values across users, corresponding to each K. Recommendations were generated based on the training split of Airbnb data and quality of prediction was assessed by comparing predictions with the test split of the Airbnb data.

In Recommender Systems, there are a set metrics commonly used for evaluation. For this thesis it was chosen to work with Top-N accuracy metrics, which evaluates the accuracy of the top recommendations provided to a user, comparing to the items the user has actually interacted in test set. This evaluation method works as follows:

- For each user:
  - For each item the user has interacted in test set, sample 250 other items (Airbnb listings) the user has never interacted.
  - The recommender models were then used to produce a ranked list of recommended items, from a set composed one interacted item and the 250 non-interacted ("non-interacted Airbnb listings") items.
  - Compute the Top-N accuracy metrics for this user and interacted item from the recommendations ranked list
- Top-N accuracy metrics was then aggregated, in order to get an overall score.

The 3 popular evaluation metrics used in this thesis:

Recall: Herlocker et al. (2004) defined recall as the probability that a relevent item will be selected. The formula used for recall@k:

$$recall@k = \left(\frac{\#\_of\_recommended\_items\_that\_are\_relevant\_@k}{total\#\_of\_selected\_items\_in\_the\_test\_set} \right)$$

Precision: (Herlocker et al. 2004) defined precision as ratio of number of relevant items selected to the number of selected items. This metric was chosen because, it represents the probability that a selected item is relevant The formula used for precision@k:

$$precision@k = \left(\frac{\#\_of\_recommended\_items\_that\_are\_relevant\_@k}{\#\_of\_recommended\_items\_at\_k} \right)$$

f-1 score: Hand, Christen, and Kirielle (2021), defined f-score as harmonic mean of precision and recall measure. It was used as it provides a single score that summarizes the precision and recall. The formula used for f-1_score@k(More weight to Precision is given):

$$f-1\_score@k = 1 + 0.5^2 * \left(\frac{precision@k * recall@k}{(0.5^2 * precision@k) + recall@k} \right)$$

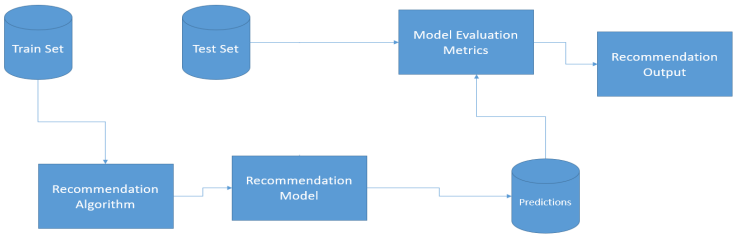

Figure 16
Workflow of the evaluation model. The data from train set is first trained on the recommendation algorithm. Then a recommendation model is made on which first the train data is fed and tested for tuning the model correctly. Finally, the predictions generated and the test set is passed to the evaluation metrics to generate the recommendations on test set.

6. Results

After performing the LDA topic modeling on the train set, following results were obtained as shown in **Figure 17**. It was observed that reviewers paid attention to 10 prevalent topics, such as Transport, Host, Amenities, Stay, Aesthetics of the stay, Hospitality, Reservation, Responsiveness, Room, and general topic that included reviews regarding family, overall experience and different ideas. The words for each topics are grouped according to their weighted contribution for that particular topic. In the **Figure 17** top-10 words are shown, according to their weighted contribution. **Figure 18** on the left shows the distribution of the top-10 topics, the size of the bubble represents the dominance of the topic. **Figure 18** on the right shows the frequency distribution of a word used in a topic. As shown in the right side of the Figure, it is evident that for topic 3 the words that was used the most was **get, staff, provide, amenity** and other words, but these words were used frequently. The topics obtained via the LDA model is discussed in detail in the forthcoming section.

Once the topics were modeled, recommendation models were created and well tuned, the model when passed to the evaluation metrics, the hybrid model outperformed the baseline model and the individual Content-based, and Collaborative filtering model the highest values for recall, precision, and f-1 score. **Table5** shows the performance of the models after calculating evaluation metric scores.

```
['Topic Stay:',
 'stay great place apartment nice host location clean recommend good',
 'Topic Transport:',
 'center bike far town exactly rent ride thoughtful gorgeous second',
 'Topic Amenities/Accessibility:',
 'get staff provide stop amenity boat bus breakfast access food',
 'Topic Reservation:',
 'day spacious arrival reservation cancel leave arrive let late totally',
 'Topic Room:',
 'room bed little night people could small bathroom big cozy',
 'Topic General:',
 'perfect friend family ever park happy owner worth nicely machine',
 'Topic Hospitality:',
 'cosy sure responsive hospitality weekend accommodation awesome bring market care',
 'Topic Responsivness:',
 'respond value quickly meet still money corner message service hope',
 'Topic Asthetics:',
 'flat stair appartment steep appartement terrace tidy keep list facility',
 'Topic Host:',
 'super cute part guy flexible due welcoming suggestion surround young']
```

Figure 17
Top-10 topics generated by the LDA model. Each topic is represented by the group of the words according to their weighted contribution in defining a topic. Top-10 words only are displayed for each topics for neat representation.

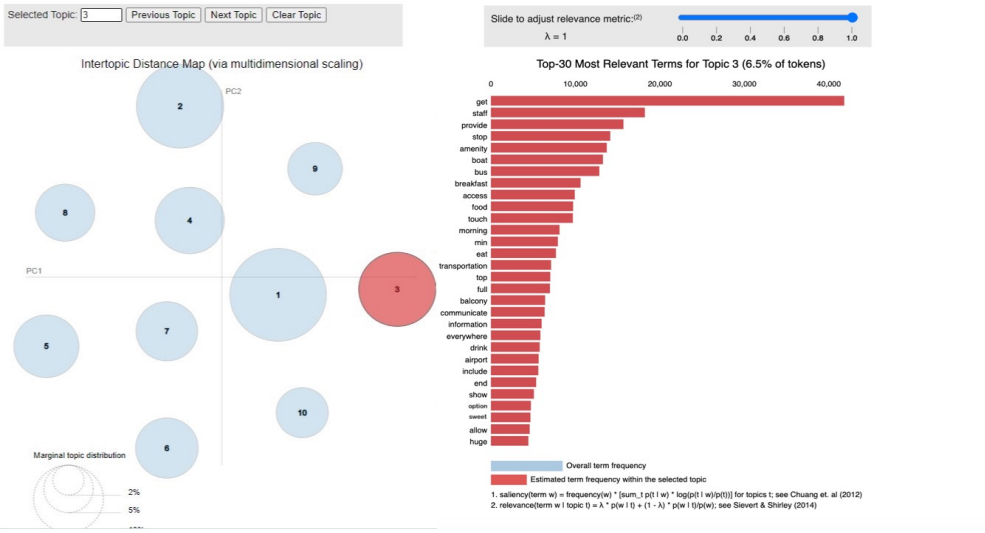

Figure 18
The layout of LDAvis which is generated using the LDAvis module, with the global topic view on the left, and the term barcharts (with Topic 3 selected) on the right. Linked selections allow users to reveal aspects of the topic-term relationships compactly.

The **Figure 19** shows the evaluation metric scores for each developed models in the form of barchart,

Table 5
Performance of the recommendation models developed based on Recall, Precision and F-1 score as the evaluation metrics.

Model Name	Recall@5	Recall@10	Precision@5	Precision@10	F-1@5	F-1@10
Popularity	0.233	0.350	0.244	0.367	0.242	0.363
Content-based	0.143	0.183	0.150	0.185	0.149	0.180
Collaborative	0.394	0.396	0.413	0.415	0.409	0.411
Hybrid	**0.444**	**0.464**	**0.466**	**0.486**	**0.461**	**0.482**

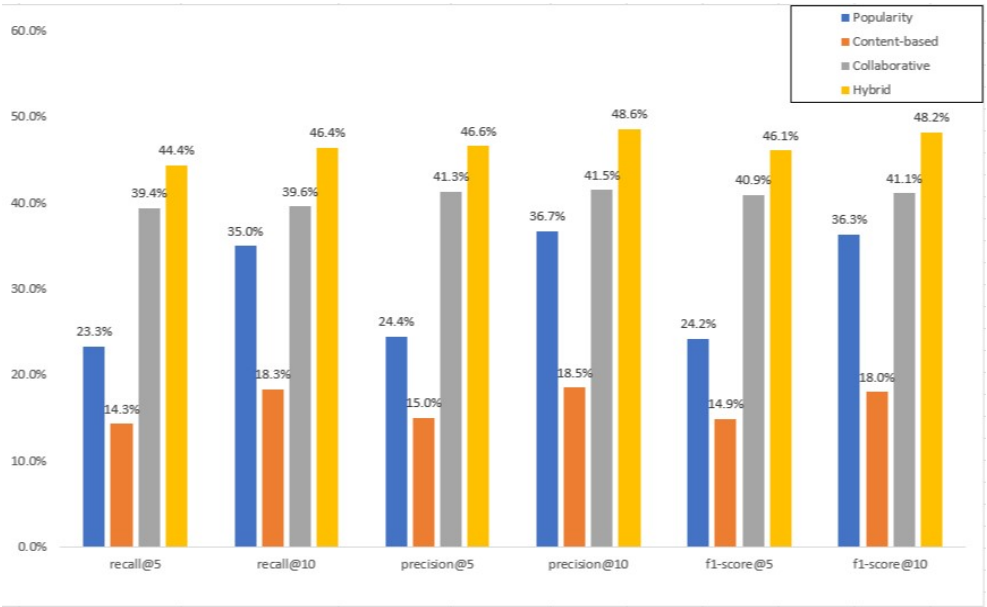

Figure 19
Evaluation metric score for Recall, Precision and F-1 score for Popularity, Content-based, Collaborative Filtering, and Hybrid Model.

7. Discussion

The main aim of this thesis was to develop a recommendation system that performs Aspect based sentiment analysis on the reviews posted by the users and then uses the sentiments of the reviews to develop a recommendation system that would generate a personalized recommendation for different users. The forthcoming paragraph answers the initial research questions mention in the Introduction section of this paper.

RQ 1. *When reviewing the stays on Airbnb which topic or set of topics are found to be of higher significance to the reviewer based on ABSA on the past reviews?*

After performing the Aspect-based sentiment analysis on the dataset, the top-10 topics that were generated is shown in **Figure 17** The clustering of the topics mostly centered around on the topic of distance, followed by amenity, cleanliness and size or the area of the room. One of the interesting findings was that location was one of the most frequently mentioned term regarding locality description. However, it was

observed that the terms, such as place, great, host, room was the most discussed topic while describing a stay. Therefore, the of the use of these words were the central theme of description of the an Airbnb stay. **Figure 18** shows the visualization of the term frequency for the 10 topics generated. The left part of the **Figure 18** shows the topics in the form of bubbles with different sizes. The size of the bubble is proportionate to the dominance of a particular topic and the right part of the **Figure 18** shows the terms that were used the most in a particular topic. It is necessary to note that the results of LDA can be interpreted differently by different people as the LDA model only outputs a probability mix of topics with the distribution of the term frequencies for the respected topics. Therefore, it requires human intervention to discover distinct topics. In order to develop deeper understanding of the topics, aspect-based sentiment analysis can be tried using a larger set of data, such as detecting target entity for every sentence and running a sentiment analysis. Also, it is important to note that because the sample size was rather small, it is subjected to the effect of words of few mouths (Roy, Datta, and Mukherjee 2019). Thus, to finally answer the **RQ 1.** it can be affirmed that the topics that were found to be of higher significance to the reviewer were ten topics, namely, **Stay, Transport, Amenities/Accessibility, Reservation, Room, General, Hospitality, Responsiveness,Aesthetics and Host.**

RQ 2. *Based on the users' most common choice of topic(s) for stays, what are the individual recommendation models that could be used to generate a personalized recommendation for each user on Airbnb?*

In order to generate personalized recommendation to the user based on their mostly used topics of discussion for an Airbnb stay(s), two models were developed Content-based filtering and Collaborative filtering model. Anyhow, the way of utilising the users' most discussed topics were approached differently in both the models, as the Content-based model used tf-idf whereas the Collaborative filtering model used LDA topic model combined with SVD model. The tf-idf model first computes the frequency of a words used in a document and the inverse document frequency of the word across a set of documents (Jones 2004). In this method the topics are represented as documents whereas in the Collaborative filtering method it generates different topics and words that represents the topic(s) in a best way. Also, a hybrid model was developed that used both Content-based and Collaborative filtering models' recommendation with the aim to be able to generate optimal recommendations.

SQ.2.1 *How do the developed recommendation model perform on the Airbnb data?*

The evaluation metric that was used to evaluate the performance of the model was recall, precision and F-1 score. The aim was to evaluate the performance of the model when it recommended top-5 and top-10 from random 250 listings in the test set. The forthcoming paragraph discusses each metric separately.

The hybrid model scored the recall5 of 44.4%, and recall10 of 46.4%, this result can be interpreted as the hybrid model was able to correctly classify 44.4% and 46.5% of unseen random 250 listings from the test set as the top-5 and top-10 recommendation. When compared with the recalls for top-5 and top-10 listings for Content-based, collaborative filtering and the most importantly the popularity model were outperformed by the developed Hybrid Model.

Second metric that was used to evaluate the performance was precision metric, When the final performance score of the hybrid model was compared to the remaining models, it was observed that for the hybrid model was able to produce 46.6% and 48.6% of relevant listings from 250 random unseen listings in the test set for top-5 and top-10 recommendations. The precision produced by the hybrid model again outperformed the precision produced by the baseline, Content-based and Collaborative filtering model for

top-5 and top-10 recommendations. The forthcoming paragraph discusses about the F-1 score for the final models.

It is important to check the accuracy of the model on the test set as the would help to understand the performance of the model on a different dataset as well. When the F1-score of the hybrid model was analyzed, it was observed that according to the result, the hybrid model turned out to be 46.1% and 48.2% accurate on the test set for top-5 and top-10 recommendation. When the F-1 score for top-5 and top-10 recommendation of the hybrid model was compared with the baseline, Content-based and Collaborative model, it was observed that the hybrid outperformed each model. Thus, after the visualization and comparison of the evaluation metrics with the models developed, it can be affirmed that the Hybrid model performed best.

SQ.2.2 *Which is the most optimal model to use for the recommendation system?*

After evaluating the performances of the individual developed models and the hybrid model as discussed in the ***SQ.2.1*** it can be said that it is possible to obtain better results for recommendation with the hybrid model, as compared to the pure Content-based and Collaborative filtering model that were developed.

This study contributes to the existing framework of recommendation systems by introducing a novel approach of using hybrid model combining the Content-based, and collaborative filtering methods. As it has been proved in the previous studies (by for example Nikulin (2014);Rodríguez, Ovalle, and Duque (2015)) that using a hybrid model gives a better performance and is suggested to develop a hybrid model for the recommendation task as the hybrid model successfully compliments the back holdings of the model which performs poor.This, suggestion was used and it worked well for this thesis. This research, however has few limitations as well. First, the dataset was only for the Amsterdam city, therefore, it lacked history of users' previous choices. In order to build better recommendation system, it is advised to use dataset of few more cities neighbouring to Amsterdam (in this case), or in general it is advisable to use dataset that has the information about user booking of more than one city. This would be helpful in not only building a better recommendation model, but also it would be helpful to analyze people's behavior while booking a place in a different parts of a city. This could provide deeper insights to develop fine-grained recommendation system. Secondly, expanding the analysis of other languages by removing the limitation of using a single language for the Aspect-based analysis could help to incorporate wider range of features that could be explored to understand and predict the booking behaviour of people based on more than one location too and developing a more generalized recommendation system to use.

8. Conclusion

With growing technology, the tourism has also grown, and due to this speed of technology growth, tourism has incorporated the use of technology and committed to provide seamless experience to the users right from booking a stay until the check-out from the place. E-commerce businesses have flourished, especially the hotel industry (Luo 2018). According to Belarmino and Koh (2018), electronic publicity or (E-Word of mouth) has played an essential role in this scenario, with E-WOM, text data comes into the picture and therefore, these huge amount of text data can be leveraged to build systems that would not only help the business to grow but also, it would help people in enjoying a seamless trip planning. In this thesis, a recommendation system for Airbnb stays in the Netherlands is built using the Aspect-based sentiment analysis as the ground. The main aim of this thesis was to apply and extract Aspect-based sentiment analysis on

the reviews posted by the users and then build a recommendation system that uses the sentiment intensity obtained from the reviews. Analysing the performance of the model developed it is worthwhile to highlight that having a large dataset always is not helpful in achieving the desired goal. A proper combination of quantity and quality of the data can do the work.

This research was dedicated to perform an Aspect-based sentiment analysis on the user generated comments and leverage the sentiments to develop the recommendation model. Therefore, the optimization was performed mainly on the text data such as choosing topic, understanding the readability features, decomposition of the feature matrix, and Part-of-speech tagging, to understand the relation between the words in order to extract the aspect features.

In the previous studies of recommendation systems using the text data alone as the basis of the recommendation system for Airbnb stays was not explored, through this research it may be helpful to similar studies where the main focus is on the User generated texts with limited information available.

One of the interesting finding was that models performed better while recommending the top-10 listings, as compared to the performance when recommending top-5 listings. Therefore in general, using top-10 recommendation would be better to use as it would provide more accurate recommendations when compared to top-5 recommendations. Moreover, the target model (hybrid model) outperformed the baseline model. While observing the performance of the Content-based model, the major limitation of this research that got highlighted was the lack of number of users with historical data. Therefore, this research paves a way for future research where further directions can be taken in the future to build on the findings:

- Incorporating variety of datasets potentially by using different cities dataset such as Brussels, Luxembourg etc. available on the Airbnb insider website. As this would help to understand the patterns in booking stays and also provide with more textual features.
- Consider using novel ML algorithms such deep learning models to effectively use the word embeddings and significantly improves the recommendations for Content-based recommendation system.
- Compare the findings of the Collaborative filtering recommendation system by using different Matrix Factorization techniques such as Spark ALS Matrix Factorization and training the model on different dataset such as Surprise, mrec in order to develop and find new insights between similarities in choices for customers.
- Incorporating different Hybrid models use, such as Cascade hybridization, Mixed Hybridization methods to develop even stronger recommendation system just by using the textual features.

References

Agresti, Alan. 2003. *Categorical data analysis*, volume 482. John Wiley & Sons.

Arora, Rachit and Balaraman Ravindran. 2008. Latent dirichlet allocation and singular value decomposition based multi-document summarization. pages 713–718.

Belarmino, Amanda Mapel and Yoon Koh. 2018. How E-WOM motivations vary by hotel review website. *International Journal of Contemporary Hospitality Management*, 30(8):2730–2751.

Blei, David, Andrew Ng, and Michael Jordan. 2001. Latent dirichlet allocation. volume 3, pages 601–608.

Cao, Qian, Xiaodi Liu, Zengwen Wang, Shitao Zhang, and Jian Wu. 2020. Recommendation decision-making algorithm for sharing accommodation using probabilistic hesitant fuzzy sets and bipartite network projection. *Complex & Intelligent Systems*, 6(2):431–445.

Chaudhary, Shreayan and C. G. Anupama. 2020. Recommendation system for big data software using popularity model and collaborative filtering. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, Advances in Intelligent Systems and Computing, pages 551–559, Springer, Singapore.

Dhillon, Inderjit S. and Dharmendra S. Modha. 2001. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1/2):143–175.

Ekawati, Devina and Masayu Leylia Khodra. 2017. Aspect-based sentiment analysis for indonesian restaurant reviews. In *2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)*, pages 1–6.

F.O. Isinkaye, B.A. Ojokoh, Y.O. Folajimi. 2015. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261–273.

Grbovic, Mihajlo and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '18, page 311–320, Association for Computing Machinery, New York, NY, USA.

Haldar, Malay, Mustafa Abdool, Prashant Ramanathan, Tao Xu, Shulin Yang, Huizhong Duan, Qing Zhang, Nick Barrow-Williams, Bradley C. Turnbull, Brendan M. Collins, and Thomas Legrand. 2019. Applying deep learning to airbnb search. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1927–1935. ArXiv: 1810.09591.

Hand, David J., Peter Christen, and Nishadi Kirielle. 2021. F*: An interpretable transformation of the f-measure. *Machine Learning*, 110(3):451–456.

Herlocker, Jonathan L., Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53.

Hoang, Mickel, Oskar Alija Bihorac, and Jacobo Rouces. 2019. Aspect-based sentiment analysis using bert. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 187–196, Linköping University Electronic Press, Turku, Finland.

Huang, Anna et al. Similarity measures for text document clustering.

Hutto, C. and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1).

Jones, K. 2004. Idf term weighting and ir research lessons. *J. Documentation*, 60:521–523.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Kaya, Buket. 2020. A hotel recommendation system based on customer location: a link prediction approach. *Multimedia Tools and Applications*, 79(3):1745–1758.

Koren, Yehuda, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Kushwaha, Amit and Shubham Chaudhary. 2017. Review highlights: opinion mining on reviews: a hybrid model for rule selection in aspect extraction. In *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, pages 1–6, ACM, Liverpool United Kingdom.

Lops, Pasquale, Marco de Gemmis, and Giovanni Semeraro. 2011. Content-based Recommender Systems: State of the Art and Trends. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*. Springer US, Boston, MA, pages 73–105.

Luo, Yi. 2018. What airbnb reviews can tell us? an advanced latent aspect rating analysis approach. *Iowa State University Digital Repository*.

Lynn, S. 2021. Merge and Join DataFrames with Pandas in Python.

Mifrah, Sara and EL Habib Benlahmar. 2020. Topic modeling coherence: A comparative study between lda and nmf models using covid'19 corpus. *International Journal of Advanced Trends in Computer Science and Engineering*.

Mowlaei, Mohammad Erfan, Mohammad Saniee Abadeh, and Hamidreza Keshavarz. 2020. Aspect-based sentiment analysis using adaptive aspect-based lexicons. *Expert Systems with Applications*, 148:113234.

Nikulin, Vladimir. 2014. Hybrid recommender system for prediction of the yelp users preferences. In *Advances in Data Mining. Applications and Theoretical Aspects*, pages 85–99, Springer International Publishing, Cham.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pontiki, Maria, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeny Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *International Workshop on Semantic Evaluation*, pages 19 – 30, San Diego, United States.

Rodríguez, Paula A., Demetrio A. Ovalle, and Néstor D. Duque. 2015. A student-centered hybrid recommender system to provide relevant learning objects from repositories. In *Learning and Collaboration Technologies*, pages 291–300, Springer International Publishing, Cham.

Roy, Gobinda, Biplab Datta, and Srabanti Mukherjee. 2019. Role of electronic word-of-mouth content and valence in influencing online purchase behavior. *Journal of Marketing Communications*, 25(6):661–684.

Serrano, Laura, Antonio Ariza-Montes, Martín Nader, Antonio Sianes, and Rob Law. 2021. Exploring preferences and sustainable attitudes of Airbnb green users in the review comments and ratings: a text mining approach. *Journal of Sustainable Tourism*, 29(7):1134–1152.

Steel, Mark. 2017. Model averaging and its use in economics. *Journal of Economic Literature*, 58.

Tang, Emily and Kunal Sangani. 2015. Neighborhood and price prediction for san francisco airbnb listings. *Departments of Computer science, Psychology, economics–Stanford University*.

Zhang, Wen, Taketoshi Yoshida, and Xijin Tang. 2011. A comparative study of tf*idf, lsi and multi-words for text classification. *Expert Syst. Appl.*, 38(3):2758–2765.