Feature Recommendation for Hotel Popularity

Lokin Sai Makkena
Department of Computer Science
University of Colorado Boulder
loma6340@colorado.edu

Nithin Veer Reddy
Department of Computer Science
University of Colorado Boulder
nika9944@colorado.edu

Sankaranarayanan
Department of Computer Science
University of Colorado Boulder
sana6469@colorado.edu

Abstract

The popularity of a hotel is not only based on the brand but also the user reviews and testimonials in current online world. To select or to book a hotel online, users would verify the reviews and assess the reputation based on the reviews and testimonials received by the hotel so far. We aim to bring out the key features which not only maintains the hotel reputation and popularity but also identify those key features and areas which lack appreciation or needs an improvement.

1. Introduction

As reported in one recent survey, a human would tend to make a decision based on reviews/feedback in an unknown/ambiguity domain of choices. This is highly applicable in a competitive travel industry. Apart from the brand, the popularity of hotel or resorts is based on user reviews across multiple platforms. Online reviews are important because they have become a reference point for buyers across the globe. So, it is important that maintaining the popularity and improving the areas where it lacks the appreciation.

We made an attempt using machine learning techniques to find out the features or key aspects which can either help to maintain the popularity of a hotel or suggest the features which needs an improvement where it lacks the appreciation. We used TripAdvisor reviews as a feed to our machine learning approaches.

2. Project Pipeline:

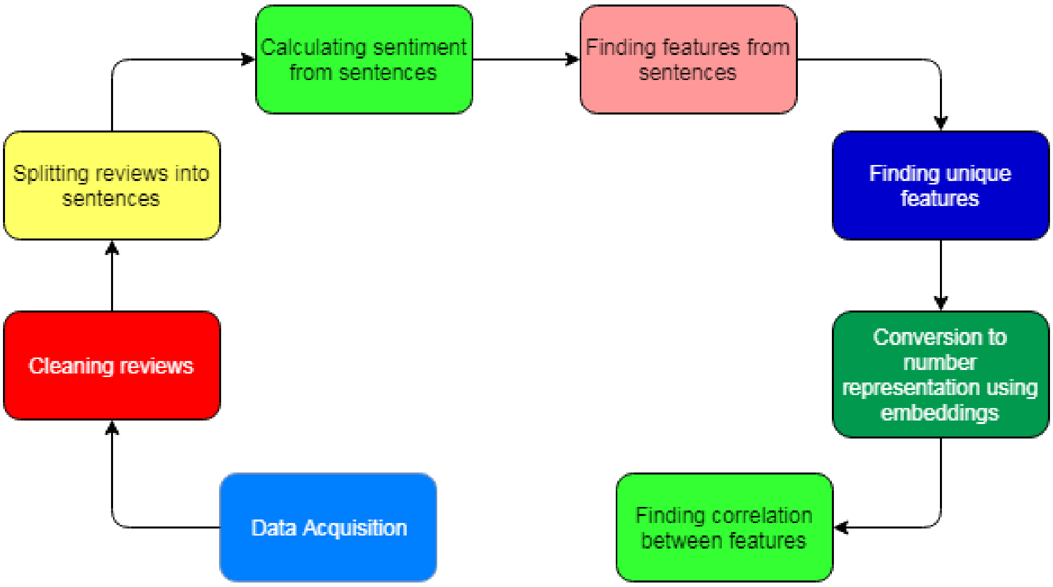

Data Acquisition
Cleaning reviews
Splitting reviews into sentences
Calculating sentiment from sentences
Finding features from sentences
Finding unique features
Conversion to number representation using embeddings
Finding correlation between features

Figure 1: Project Pipeline Flow Chart

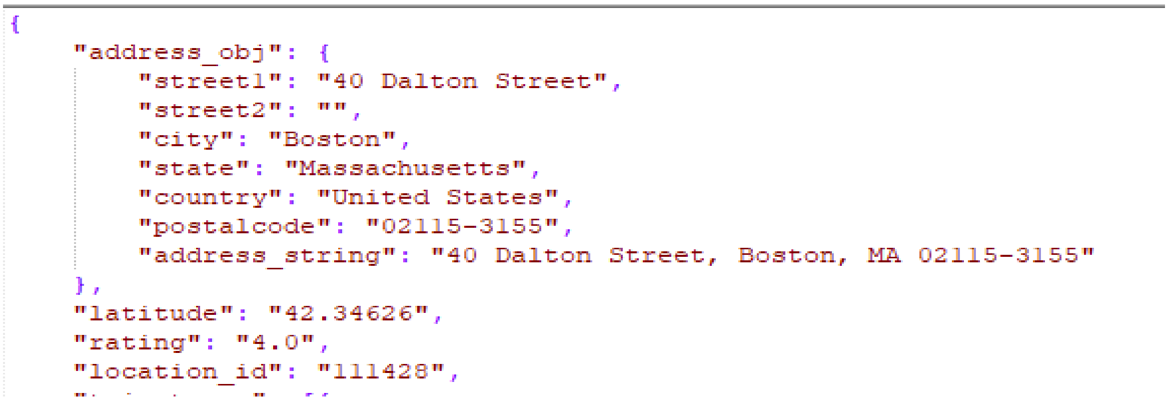
3. Data Extraction:

TripAdvisor is one of the popular and biggest aggregators for user given hotel reviews and rating. We used TripAdvisor in collecting the data. Data collection is divided into 2 parts.

- **Fetch the meta data using tripadvisor API:**

URL : http://api.tripadvisor.com/api/partner/2.0/location/<api_key>

Sample Response:

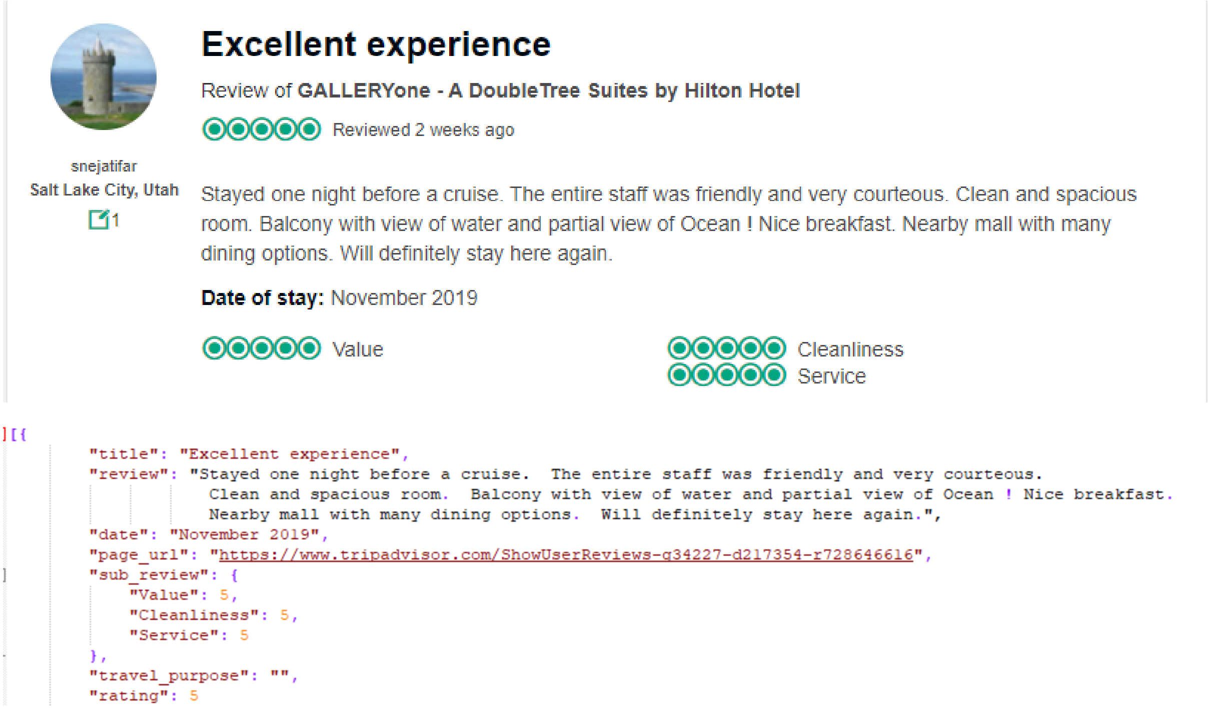


```
{
    "address_obj": {
        "street1": "40 Dalton Street",
        "street2": "",
        "city": "Boston",
        "state": "Massachusetts",
        "country": "United States",
        "postalcode": "02115-3155",
        "address_string": "40 Dalton Street, Boston, MA 02115-3155"
    },
    "latitude": "42.34626",
    "rating": "4.0",
    "location_id": "111428",
```

Figure 2: Sample Response of TripAdvisor API

- **Scrapping the complete reviews from the webpages and saving the data in json format.**

We used beautiful soup – python library for scrapping web pages.
Below is an example of actual review and scrapped content in a json format



Excellent experience

Review of **GALLERYone - A DoubleTree Suites by Hilton Hotel**

●●●●● Reviewed 2 weeks ago

snejatifar
Salt Lake City, Utah
1

Stayed one night before a cruise. The entire staff was friendly and very courteous. Clean and spacious room. Balcony with view of water and partial view of Ocean ! Nice breakfast. Nearby mall with many dining options. Will definitely stay here again.

Date of stay: November 2019

●●●●● Value	●●●●● Cleanliness
	●●●●● Service

```
][{
        "title": "Excellent experience",
        "review": "Stayed one night before a cruise.  The entire staff was friendly and very courteous.
                Clean and spacious room.  Balcony with view of water and partial view of Ocean ! Nice breakfast.
                Nearby mall with many dining options.  Will definitely stay here again.",
        "date": "November 2019",
        "page_url": "https://www.tripadvisor.com/ShowUserReviews-g34227-d217354-r728646616",
        "sub_review": {
            "Value": 5,
            "Cleanliness": 5,
            "Service": 5
        },
        "travel_purpose": "",
        "rating": 5
```

Figure 3 : Sample Webpage and Corresponding Json Response

Data extraction was a tedious task as most of the time client IP would be blocked due to over usage. For this we have used IP spoofing while scrapping the webpages.

We have collected scrapped data for over 48K hotels. All the scrapping script and sample scrapping content are uploaded into git.

Once the data is extracted, it is indexed on to elasticsearch for all future use of data querying and filtering purposes.

4. Feature extraction:

The first step in building the data for machine learning algorithms is to extract the features. We have our data for each hotel id indexed according to months on elasticsearch which makes it easier to extract features for a particular hotel.
Listed below are the sequence of steps followed in the pipeline:

4.1 Text Cleaning:

This step involves stripping out punctuation except full stop. After this we normalize everything to lowercase and remove numbers if any.

INPUT:

hurry back we visited here from UK after a hectic stay at Universal Orlando. The hotel is very well located on sand key with a beautiful beach a few minutes walk away. Although the hotel is dated, it still offers a great holiday full of the best activities and things you want to do on vacation.

OUTPUT:
hurry back we visited here from uk after a hectic stay at universal orlando. the hotel is very well located on sand key with a beautiful beach a few minutes walk away. although the hotel is dated it still offers a great holiday full of the best activities and things you want to do on vacation.

4.2 Breaking into Sentences:

To identify features in a better way, we split the reviews into a list of sentences. This way, it is easier to identify features which can then be built across sentences and across reviews.
The example that has been used for text cleaning in the above sub division has been broken into a list of sentences as given below.

INPUT:

hurry back we visited here from uk after a hectic stay at universal orlando. the hotel is very well located on sand key with a beautiful beach a few minutes walk away. although the hotel is dated it still offers a great holiday full of the best activities and things you want to do on vacation.

OUTPUT:

['hurry back we visited here from uk after a hectic stay at universal orlando', 'the hotel is very well located on sand key with a beautiful beach a few minutes walk away', 'although the hotel is dated it still offers a great holiday full of the best activities and things you want to do on vacation']

4.3 Finding features:

Our intuition here is that most of the features that we are trying to identify from the review are nouns (mostly common nouns). But all common nouns are not features. <*Modify accordingly*>If we find an adjective associated with the noun, then we are counting that noun as a feature. Proceeding further, we split the sentence into a list of words and assign a Part of Speech (PoS) tag to each word. Now we check for possible nouns and associated adjectives in each sentence. The adjective is going to contain information about whether the feature identified is a good or a bad feature.

INPUT:

great beach clearwater beach the best in usa tennis courts good pool busy but ok good restaurants friendly staff great gym and location location location there are so many things to do in the area that we were so happy with the amount of choices. beach jet skiing attractions etc. some really great local restaurants too like island grill and columbia. it's a fab tourist area.

For example, in the first sentence, the words are given Part of Speech Tags. Great- JJ(Adjective), beach-NN(Noun). As there is an adjective 'Great' near to the word beach, the word beach is identified as a feature. Also in the phrase 'friendly staff', 'friendly' is identified as an adjective to the word 'staff' which is a noun. So here 'staff' is a feature.

OUTPUT:

Example sentence features identified for the above review

['beach', 'clearwater', 'tennis', 'pool', 'staff', 'location', 'location', 'location', 'area']

Example Unique features identified

['staff', 'shop', 'pool', 'area', 'staff', 'buffet', 'dinner', 'beach', 'stay', 'remainder', 'stay', 'water', 'bottle', 'sheet', 'bed', 'comforter', 'price']

4.4 Phrase based feature identification:

We also encountered a few edge cases during our implementation of this method. If there is a conjunction (Eg: *and*) linking two different phrases in the same sentence, then the algorithm of identifying features does not work perfectly because there is a chance that there could be a feature in each phrase. So, we split the sentence into phrases to check for features in each phrase which thereby improved the feature extraction part. After this, we need to make sure the same features are not extracted repeatedly. An extra check is to make sure we are collecting unique features only.

What we have done until now is to identify features in a particular review text for a particular month for a particular hotel. Now we need to expand this to all reviews, all months and all hotels individually. Before proceeding further, we also need to ensure that the features collected across the months are the same. So, we will need to pick up only the common features for a hotel across all months of data.

4.5 Building an alternate representation using embedding:

The final step in this feature extraction process is to build a representation using Word2Vec embedding. We used Word2Vec over TF-IDF as Word2Vec retains the semantic meaning of different words in the text. Also, here, the size of the embedding vector is not large. We used 100 dimensions for embeddings. So, we can tackle the problem of sparse vectors. Also, the context information is not lost.

As shown below, features have a 100 dimension word embedding on the 'z' dimension. The 'x' axis represents the features. The 'y' axis represents the months for which data for the hotel is collected.

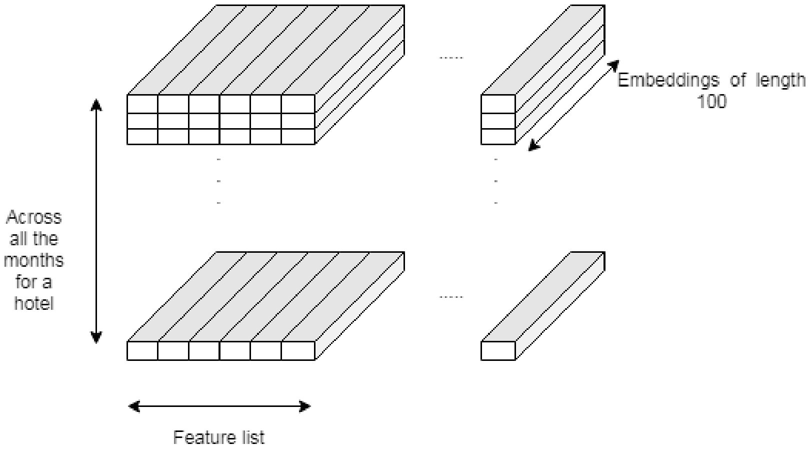


Figure 4 : Word2Vec embedding representation in three dimensional space

For the Word2Vec model, we supply the unique features and also set min_count = 1. After building the model, we check whether the feature was in the vocabulary used by Word2Vec. If it was present, we then directly use the Word2Vec model's value for the particular feature. We keep accumulating the features across months and then index the features collected by hotel id.

We have showed a small subset of the entire model for a particular month.

Table 1: A Subset of Word2Vec embeddings for features

luxury	beach	stay	hotel
-0.00275	-0.003718	-0.00376	0.001941
0.000219	0.002085	0.002161	0.003435
-0.00453	-0.002839	0.000024	0.001617
0.004064	-0.004901	-0.0039	-0.00155
0.004967	-0.003916	-0.00201	-0.00141
-0.00172	0.002839	0.002175	-0.00318
0.000337	0.003708	0.004093	0.004378
0.004625	-0.004225	0.004521	0.004066
-0.00365	-0.0009	0.001215	-0.00174
0.000793	-0.000027	0.003821	0.002496

There is however one small problem here. As we are extracting from text from a large collection of reviews, the number of features is huge. Data Representation will become an issue here. So, we decided to do feature selection to select the best features.

5. Feature Selection:

The concept of feature selection plays a vital role when the number of features in the dataset is large. Any machine learning model will use each feature in the model for training on the data but putting all the features will not give good results since there exist noise in the data. Only a subset of features will be important for the machine learning model to produce better results. For our project, we have used two feature selection methods to pick features that are crucial to determine hotel popularity.

The methods which were used for the feature selection process are:

- Pearson Correlation Coefficient
- Principal Component Analysis (PCA)

5.1 Approach for Pearson Correlation Coefficient:

The Pearson coefficient measures the strength of the association between any two variables. We have calculated the coefficients between all the features directly from the features without implementing principal component analysis. The figure xx shows the person correlation values calculated directly between the features without using principal component analysis.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

$n - total\ number\ of\ pairs$

$x - feature\ x\ values$

$y - feature\ y\ values$

Figure 5 : Pearson Correlation Coefficient Formula

Table 2: Correlation values between features of a hotel without principal component analysis

	staff	**beach**	**key**	**island**	**pool**	**sand**	**resort**	**hotel**
staff	1.000000	0.437403	0.221133	0.505270	0.329338	0.229678	0.140754	0.883246
beach	0.437403	1.000000	0.844406	0.812337	0.627045	0.752106	0.733941	0.440206
key	0.221133	0.844406	1.000000	0.888649	0.663913	0.958090	0.943196	0.421862
island	0.505270	0.812337	0.888649	1.000000	0.784944	0.806900	0.800733	0.713418
pool	0.329338	0.627045	0.663913	0.784944	1.000000	0.543596	0.526276	0.447131
sand	0.229678	0.752106	0.958090	0.806900	0.543596	1.000000	0.943240	0.400521
resort	0.140754	0.733941	0.943196	0.800733	0.526276	0.943240	1.000000	0.318116
hotel	0.883246	0.440206	0.421862	0.713418	0.447131	0.400521	0.318116	1.000000

5.2 Approach for principal component analysis:

This unsupervised learning technique was used to reduce the feature dimension space into lower dimension space. From each principal component, we are choosing a feature which is important based on values in the eigenvector. The values in eigenvector are called the loading scores or coefficients which signifies the variable which is strongly influencing the component. The higher the loading score is the higher feature importance. Later, the Pearson correlation is calculated on all the features from the principal components. The following will explain the approach of selecting important features as follows:

5.2.1 Data for principal component analysis:

Common features across all months of each hotel are segregated to improve the chances of identifying important features.

Table 3: Common features across the months for a hotel

S.No.	Common features per hotel
1	beach
2	hotel
3	island
4	key
5	pool
6	Resort
7	sand
8	staff

5.2.2 Standard Scalar Transformation:

Standard scalar is applied to common features for each hotel to transform its distribution with mean 0 and standard deviation 1. Each value of the feature in the data is subtracted from the mean and divided by the standard deviation. The table 4 shows the standard scalar output of one column. Rest all the feature columns per hotel for transformed in this way

$$Z = \frac{X - \mu}{\sigma}$$

$$\begin{gathered} z \text{ is the standardized value} \\ X \text{ is the feature value from feature embedding} \\ \mu \text{ is the mean of the featue embedding} \\ \sigma \text{ is the standard deviation of the feature embedding} \end{gathered}$$

Figure 6: Standard Scalar Formula Applied On Each Feature

Table 4: Example of standardized values for column “key”

For “key” column	Standardized values
April	0.29963117
August	2.51654144
February	-0.33832862
January	-0.68477461
July	-0.71897602
June	0.77896115
March	-0.44930097
May	-1.18804266
October	0.23388166
September	-0.44959254

5.2.3 Implementing principal component analysis:

This method is applied to the of common standardized features per hotel and reduces the dimensions of the data to lower dimensions of principal components. The target variance was 0.99 and the data was reduced to 4 components after PCA.

$$\begin{gathered} pc = w_{staff} * staff + w_{beach} * beach + w_{key} * key + w_{island} * island + \\ w_{pool} * pool + w_{sand} * sand + w_{resort} * resort + w_{hotel} * hotel \end{gathered}$$

Each principal component score is expressed as a linear combination of features with corresponding loading scores acting as weights. The important feature is picked up from each of the 4 components based upon the highest loading score or coefficient from each component (highlighted in bold below). This is shown in table 5.

Table 5: Loading scores per component and corresponding feature names

Components	Loading Scores per Component	corresponding feature based on value
PC1	0.218, 0.377, 0.406, **0.417**, 0.325, 0.385, 0.373, 0.282	island
PC2	**0.663**, 0.055, 0.248, 0.062, 0.027, 0.254, 0.323, 0.568	staff
PC3	0.188, 0.009, 0.08, 0.125, **0.866**, 0.304, 0.273, 0.16	pool
PC4	0.236, **0.835**, 0.034, 0.178, 0.122, 0.175, 0.183, 0.367	beach

5.3 Implementing Pearson correlation coefficient:

This correlation method is computed between the important features of each hotel from the principal component analysis. The correlation matrix shows how strong is the association between each pair of features. The table 5 is the result of the correlation coefficients values between important features from principal component analysis. We can interpret that island is highly correlated to beach and pool.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

$$n - total\ number\ of\ pairs$$
$$x - feature\ x\ values$$
$$y - feature\ y\ values$$

Figure 7 : Pearson Correlation Coefficient Formula

Table 5 : Pearson Correlation Coefficient with PCA

	island	staff	pool	beach
island	1.000000	0.505270	0.784944	0.812337
staff	0.505270	1.000000	0.329338	0.437403
pool	0.784944	0.329338	1.000000	0.627045
beach	0.812337	0.437403	0.627045	1.000000

6. Error Analysis and Evacuation Procedures:

For the hotel id 111428 (data is present in the git reference), we found out that the word ‘kind’ was being selected as a feature. There are two situations in which the word ‘kind’ has been used.

Usage 1:

The room was spacious and always clean but what made this hotel excellent was their staff. They were helpful, kind, and friendly. Always ready with a smile.

Usage 2:

Visit Newbury street to explore unique and trendy boutiques – as well as every other kind of store imaginable – all housed in beautiful brownstone buildings. This confusion is due to different usage of the word 'kind' in two situations.

In the first example 'kind' is used as an adjective to the word 'staff'. In the second example, it is used as a noun because it is describing a store as being in a particular class. As there are two different usages of the word 'kind', this is sometimes misinterpreted by the system as a feature.

As we have an unsupervised data, using some manual efforts we have tagged few hotel reviews and identified the features that play a prominent role in hotel reputation and also those features which contributed for a negative reputation. We have manually assed the feature importance and made a relative comparison with the output of manual work vs output of ML approach. Most of the scenarios the relative comparison was promising while there are certain features that didn't make any sense.

7. Conclusion:

As mentioned, we have made an attempt to figure out the feature importance by above approach. This approach yielded good results for a hotel where the diversity on features are good. For a hotel reviews which lacks diversity of features didn't yield a meaning full result. We could have made some improvements using some prior tagging or labelling the features that would be under consideration. This whole project would have been dealt with multiple possible approaches and above is one of the approaches. This approach is more of experimental and learning strategy by implementing some of the machine learning algorithms described in the class.

8. References:

[1] TripAdvisorApi https://developer-tripadvisor.com/content-api/documentation/

[2] BeautifulSoup https://pypi.org/project/beautifulsoup4/

[3] TripAdvisor Reviews https://www.tripadvisor.com/ShowUserReviews-g41129-d263087-r8615840-Hampton_Inn_Elkton-Elkton_Maryland.html

[4] Principal Component Analysis https://scikit-learn.org/stable/modules/generated/sklearn-decomposition.PCA.html

[5] VaderSentiment: https:pypi.org/project/vaderSentiment/2.1