Finding the best Airbnb for travellers

CMPT 353

Dipansh - 301329048
Shan Ying (Felicity) Yang - 301305759

Abstract

The project aims to find the best AirBnbs for travellers to stay in Vancouver. The scope of the project is to suggest Airbnb listings to users based on the nearby outdoor amenities. We used the provided "`amenities-vancouver.json.gz`" and Airbnb's listings dataset to acquire the amenities around each AirBnb listing. We also explored if the price of listings correlates to listings' total nearby amenities using supervised machine learning techniques.

Introduction:

The problem we found for travellers is that not many travelling websites or applications would provide the option to tell them the different outdoor amenities around their place of stay. Most of the hotel and Airbnb listings only provide the indoor amenities, but not the shops nor restaurants around their stay. Therefore, we thought of resolving this issue by finding the outdoor amenities around the Airbnb listing and providing an amenities rating.
Since we want to develop a more user-centred program, we also ask users to input their preference on Airbnb's price range, number of bedrooms, and number of accommodates.

Furthermore, we performed three regression techniques to analyze if there exists a stronger correlation between Airbnb listing's price and nearby amenities. As a result, we will produce two CSV files, one containing the complete filtered listing information and the other containing the result from the machine learning models.

Data Analysis and Cleaning:

We used the provided Vancouver amenities dataset and the detailed AirBnb's Vancouver Listings (listings dataset) which was gathered from Airbnb's website.

Amenity dataset consisted of 'lat', 'lan', 'timestamp', 'amenity', 'name' and 'tags'. First, we removed unnecessary columns (timestamp, tags) and then checked for NaN values. We further analyzed the amenity dataset using the .value_counts method on the amenity column and found all the different amenities. We filtered the dataset based on the amenities that we would like nearby if we were a traveller and only kept the rows with the needed amenities information.

The Listings Dataset had 74 total columns. We found all the column names using the pandas ".column" method, and we removed all the unnecessary columns and kept these columns:

```
['id', 'listing_url', 'name', 'description', 'picture_url', 'latitude', 'longitude',
'property_type', 'accommodates', 'bedrooms', 'beds', 'amenities', ‘price']
```

The listings dataset was first filtered based on the user requirements of the number of accommodates, the number of bedrooms needed and the budget. This input is read by a .txt file of the following format:

```
accommodates: 5
bedrooms: 3
price range: 0-100
exact: False
```

If the user sets exact to True then the program will only show listings with the exact value of accommodates and bedrooms, if the exact is set to False then the program will treat the accommodates and bedrooms as the minimum value. In the above example, the program will show the listings which support more than or equal to 5 accommodates and have 3 or more bedrooms.

Feature Engineering:

Since we want to know the specified amenities around all Airbnb listings, we created an algorithm that calculates the number of different amenities within 1km of the listing's coordinates. Then, we calculated the amenity score for each listing based on the number of different types of amenities and their frequency.

- The num_amenities function returns a dictionary of the number of amenities in a 1km radius of each Airbnb listing. This function is simplified with the help of the haversine_distance function, taking the listing data's coordinates and the filtered amenities DataFrame as parameters.

- Using the haversine_distance function, we accept the filtered amenities DataFrame and the latitude and longitude data from the filtered AirBnb's listing DataFrame. The amenities' latitudes and longitudes are also extracted from the DataFrame inside the haversine_distance function. All of the latitudes and longitudes are converted into radians.

- The following haversine formula is applied to find the distance between the two given points:

$$d = 2r \arcsin\left(\sqrt{\operatorname{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\operatorname{hav}(\lambda_2 - \lambda_1)}\right)$$
$$= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

- The haversine distance is then multiplied by the earth's radius in kilometres (6371km) to return the actual circle distance between the two points on earth.
  The amenities_score function is relatively simple after we have appended the amenities dictionary to each Airbnb listing. We iterate through the filtered listing DataFrame and increment the score based on the number of different types of amenities and their frequency.

The Resulting listings dataframe was then sorted based on the amenity_score.

Machine Learning:

We further analyzed our data with regression techniques from supervised machine learning. The original Airbnb listing data was filtered with the function clean_data_ML specifically for this step. We dropped unnecessary columns and kept only the columns which we thought could affect the price such as columns with information about the hosts, listings and the reviews.

We corrected the wrong data types (i.e., converting from strings to floats), and used LabelEncoder to transform categorical data. After data filtering for the regression algorithm was completed, we obtained the training and testing data using the train_test_split function in the Sklearn model, which splits the data arrays into two subsets. The X data is the filtered listing data without the price column, and the y data is the price column of the listing data. The models are applied to the listing data without the amenitiy_score initially. Then the models are applied to the listing data with the amenity_score column added to it. Upon successfully splitting the data arrays, we fitted the models after calling the regressor functions from the Sklearn model in the following format:

- knn = KNeighborsRegressor(n_neighbors=50)
- rf = RandomForestRegressor(100, max_depth=40)
- gb = GradientBoostingRegressor()

The coefficient of determination (R^2) for each model is obtained for analysis.

Findings:

The following table displays the computed coefficient of determination for the original and the filtered Listing Datasets using the three different regression models:

Table 1. Sample Output of Coefficient of Determination for Listing Datasets

Regressors	**Original Listing Dataset (without Amenity Scores)**	**Filtered Listing Dataset (with Amenity Scores)**
K-Nearest Neighbors	0.05235971270926676	0.08927844411212049
Random Forest	0.6421580362281065	0.6691652824194065
Gradient Boosting	0.659758407071633	0.6788327640120857

The coefficient of determination represents the square of the correlation between the predicted and actual prices. A higher coefficient implies a better fit for the regression models. The above table sample output shows that the filtered listing data has a higher coefficient of determination (R^2) than the original listing dataset from all models. From observation, we concluded that nearby outdoor amenities are more correlated to price prediction. However, other factors, such as fancy or luxury interiors, could impact the price of a listing.

Output:

We output the two CSV files: the "ML_Price_Prediction.csv" that represent results of the machine learning model scores, and the "AirBnb_search_results.csv," with the sorted Airbnb listings.

Limitations and Conclusion

One limitation we had in the project is that we did not have enough data to support the claim in our findings of machine learning models. During our analysis of the correlation between outdoor amenities and Airbnb listing prices, we did not have data for the other features that may have influenced the listing price.

We have tried multiple approaches to find a meaningful relationship between outdoor amenities and Airbnb listings in Vancouver. Eventually, we narrowed it down to the final version, where we examine the relationship and provide suggestions to users according to the outdoor amenity scores. If we had more time, we would further clean and select the appropriate data for analyzing the relationship between listing price and listing's outdoor amenities. We would also find more data with factors influencing the prices of Airbnb. We would also further polish the display of the listings results and plot it on a street view map.

Project Experience Summary

Dipansh:

- Performed Data analysis and Data cleaning on both datasets.
- Performed feature engineering to find the number of amenities in 1km radius of each listing.
- Prepared data for machine learning models.
- Performed parameter tuning to improve the performance of the machine learning models.
- Analysed the results of the machine learning models.
- Documented the conclusions and contributed to the project report.

Shan Ying (Felicity) Yang:

- Proposed project outline and milestones
- Refined the research question for the project
- Cleaned datasets
- Prepared input and output data
- Analyzed findings from machine learning models
- Conducted project report covering all requirements and analysis