# Capstone Project

Data Science Nanodegree

Carlo Giordana

August 16th, 2023

# Definition

## Project Overview

For Airbnb hosts a critical factor for the success of their business is having high scores associated to their listings. However, for beginners it is not easy to realize which aspects are the most influential when working to boost thier ratings in order to invest their time and money more efficiently. Moreover, when enlisting a new post on the platform they could be also interested in understanding if the description of they provide for the accommodation enhances their appeal and if it leads to better reviews.

In this project I performed a statistical analysis to determine which are the more important features of a listing that are associated to high review scores. Then I created a web application that, according to the results of the analysis, allows potential hosts to test if a new post could be catchy after enlisting and to predict the expected review ratings for their accommodation. The application uses a regression model trained on the Airbnb activity record in the Seattle area over a year span.

The data are available in open format at Kaggle.com and the project was inspired after the discussions about these data available on the platform.

## Problem Statement

The goal is to create a web application to predict the expected review rating of an Airbnb given certain factors imputed by the user. These factors are determined by a preliminary analysis, based on both heuristic considerations and data exploration.

The tasks involved are the following:

1. Download and preprocess the Seattle Airbnb open data.
2. Determine the most influential features in the dataset to predict the review rating scores by studying the correlation coefficients of a linear model.
3. Train a regressor that can predict the expected rating scores for a new listing.
4. Develop a front-end component to collect users' inputs and display predictions based on the trained regressor.
5. Make the application run on a browser.

The application is expected to be useful for hosts in evaluating if their listings are going to be successful.

## Metrics

*Coefficient of Determination* (denoted as $R^2$) and *Mean Squared Error* (MSE) are the most common metrics used when studying the performances of a regression model. They provide a measure of how well observed outcomes are replicated by the model.

In particular:

- *$R^2$ score* is the proportion of the variation in the dependent variable that is predictable from the independent variables
- *MSE* is the average squared difference between the estimated values and the actual ones

# Analysis

## Data Exploration

The Seattle Airbnb open data includes the following activity record:

- Listings, including full descriptions and average review score
- Reviews, including unique id for each reviewer and detailed comments
- Calendar, including listing id and the price and availability for that day

The *Listings* dataset refer to the over 3800 accommodations that were enlisted to Airbnb for the Seattle in 2016. For each listing more than 90 distinct features are available, including:

- Textual descriptions of the accommodation
- Attributes of the host
- Information about address and location of the accommodation
- Characteristics of the property
- Prices and other fees
- Information about the availability
- Attributes related to reviews and rating scores
- Classification according booking and cancellation policies
- Ids, URLs and other technical attributes related to the render of the listing on a browser and to the scaping of the data

Among these columns some have missing values for a large proportion of records. In some cases (such as for the *square_feet* field) the presence of these missing data is due to an incomplete filling of the listing, while in other situations (such as for the *monthly_price* field) the values are missing due to inapplicability. Moreover, other fields assume a unique value over all the dataset or for the vast majority of the records.

Other issues are related to the format of the data for some fields. In particular monetary amounts, percentages and Boolean values are always expressed as text stings presenting *$*, *%* and *t/f* characters that prevent a quantitative analysis before some refinements.

## Data Visualization

Since the focus of the project is to study the factors that influences the review scores rating, some preliminary data representations were realized to better understand the relation between ratings and other features of the listings.

First were computed average ratings by neighborhood, resulting in low but significant differences from the overall city average (figure 1).

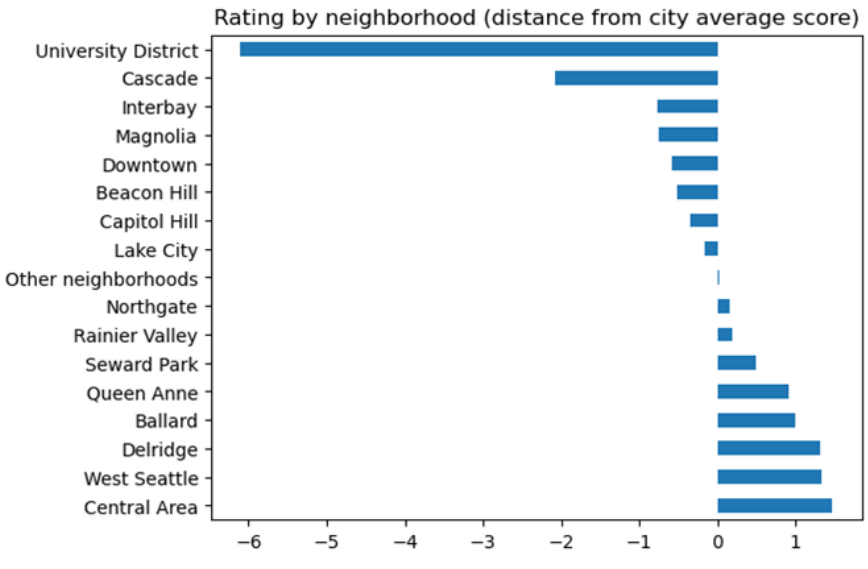


*Figure 1*

*Among the 17 neighborhoods, the scores of 16 of them appear not to deviate much from the average, being 94.5 out of 100. The only neighborhood that makes an exception is "University District" having a score lower than 90.*

Then the listings were split in price and number of available days tiers and the average ratings were computed for each of these tiers. As depicted in figure 2, there was no evidence of an influence of prices, while the total day of availability could play a role in determining the final rating.

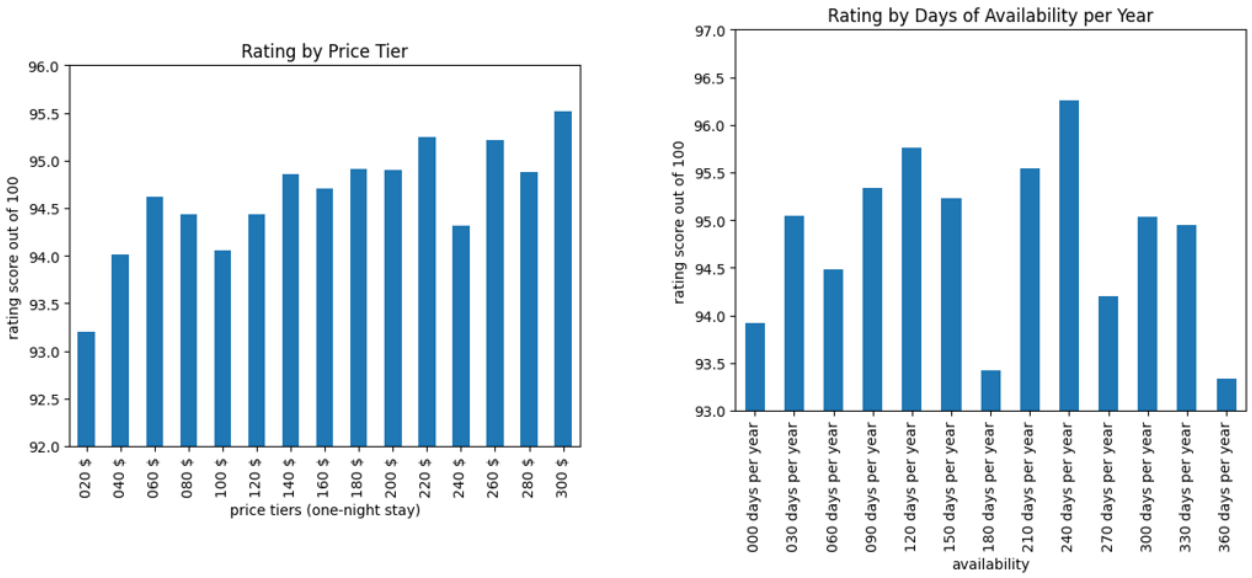


*Figure 2*

a) *Even if there is as slight growth of ratings corresponding to an increase of prices, accommodations associated to all price tiers have similar average rating.*

b) *There is not evident pattern of relation between availability and ratings emerging from the representation. The only noticeable deviation is a lower average rating for accommodations available all year around, that incidentally are the more numerous among all tiers.*

More analysis was performed in order to answer some related business questions. You can find the details about this data exploration at [this post](#) this post on Medium.com.

# Methodology

## Data Preprocessing

The preprocessing done in the "Process Data" notebook consists of three main stages:

1. data cleansing
2. selection of the most influential features
3. data preparation for the prediction model

The data cleansing is conducted by the following steps:

1. records with average rating missing are removed
2. a subset data frame with columns of interest for feature selection is defined
3. quantitative columns are cast to numeric type when possible
4. categorical columns are encoded
5. binary columns with almost same values for all records are removed
6. licit null values are imputed
7. collinear columns are removed

The feature selection is performed by fitting a linear regression model and studying its correlation coefficients. The components which present an absolute value less than 0.1 are excluded from the model, while the other factors are selected to be added to the list of the predictor for the final regression model.

The last stage of data preprocessing consists in the computation of the final input dataset for the regression model by merging the text feature with the quantitative predictors selected in the former step.

## Implementation

The implementation process can be split into two main stages:

1. The regressor training stage
2. The application development stage

During the first stage, the regressor was trained on the preprocessed training data. This was done in a Jupyter notebook (titled *Train Regressor*) and can be further divided into the following steps:

1. Load the preprocessed dataset from database
2. Split the dataset in training and test sets
3. Write a tokenization function to process the text data
4. Build a machine learning pipeline structured as follows
    a. the text feature is transformed by `CountVectorizer` and `TfidfTransformer`
    b. the numeric features are transformed by `StandardScaler`
    c. the transformed features are fitted through `DecisionTreeRegressor`
5. Train the pipeline by fitting on the training set
6. Test the predicting performances of the model
7. Save and freeze the trained model

The application development stage can be split into the following steps:

1. copy the instruction in the Process Data and Train Regressor notebooks in the respective python scripts
2. implement the html templates and the *run.py* script that should be executed for launching the application

## Refinement

To enhance the prediction performances of the pipeline, the training of the model was repeated five times for each combination of parameters. Moreover, 108 different combinations of parameters were tested in order to obtain the best tuning choice minimizing the prediction errors. This job was performed via grid search.

# Results

## Model Evaluation and Validation

During development, a validation set was used to evaluate the model. The final architecture and hyperparameters were chosen because they performed the best among the tried combinations.

In particular, as shown by the grid search training log:

- Linear Regression
- Elastic Net
- Decision Tree Regressor

As depicted in the following table the best choice is the Elastic Net regressor since its validation scores are significantly higher than the other models.

| **Model** | **R2 score** | **MSE** |
|---|---|---|
| **Linear Regression** | -3.37 | 187.34 |
| **Elastic Net** | 0.35 | 5.21 |
| **Decision Tree Regressor** | -0.56 | 8.42 |

## Justification

Unfortunately, the validation results are quite poor, resulting in a R2 score of 33% and mean square error of 3.19. Therefore, the prediction performances of the algorithm are expected to be low and a further analysis to enhance performances is overdue.

In summary, the application is useful mainly in suggesting to hosts which are the most influential aspects to increment their ratings but does not execute an excellent job when predicting the expected rating for a new post to be enlisted. See the section Improvement for my suggestions about the further improvement steps that are necessary.

# Conclusion

## Reflection

The process used for this project can be summarized using the following steps:

1. An initial problem and relevant, public datasets were found
2. The data was downloaded and preprocessed
3. A subset of the most relevant feature in the dataset was computed
4. A regression model was defined and trained on the cleaned data
5. The parameters of the model were tuned to obtain better performances
6. A front-end component to collect users' inputs and to display prediction results was implemented

I found step 2 very time consuming due to the large number of available features and to the "dirtiness" of the data.

As for the most interesting aspects of the project, I'm very glad that I found the Seattle Airbnb datasets that is very complete and has a lot of useful information that I am sure I am going to explore further in future projects.

I also appreciated the opportunity to get to use Flask framework, since it widens my portfolio relative to front-end development tools and I found it very helpful for a data science project like this.

## Improvement

There are many improvements that could be realized for the application in order to become a really useful tool for potential hosts.

First some tooltips and more explanatory labels could be added to the front-end component. A page reporting the general scope of the application could be useful too.

Second, since the prediction performances of the model are quite poor, it could be further refined by exploring more regression techniques and using a larger spectrum for tuning parameters.

Third and more important, the application would be useful in real World only if the data used for the training were drastically widened by for instance:

- Including data from other cities (each user could be able to get better predictions if data are focused on its own city)
- Including data about availability that were touched marginally by the project.
- Include the text of the reviews to add a better context for the ratings and to analyze which are the key factors for success directly from the words of guests.