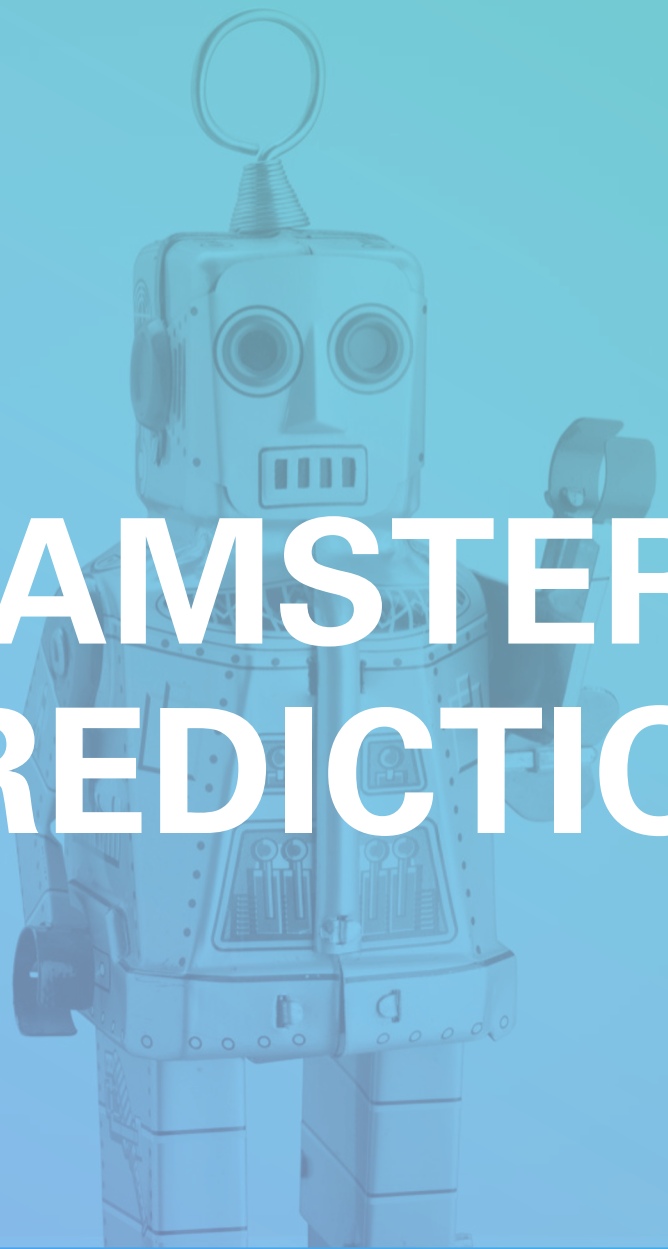


AIRBNB AMSTERDAM PRICE PREDICTION

Ö.Faruk GÖKBAK
AI41 - 3782174

The goal of the project?

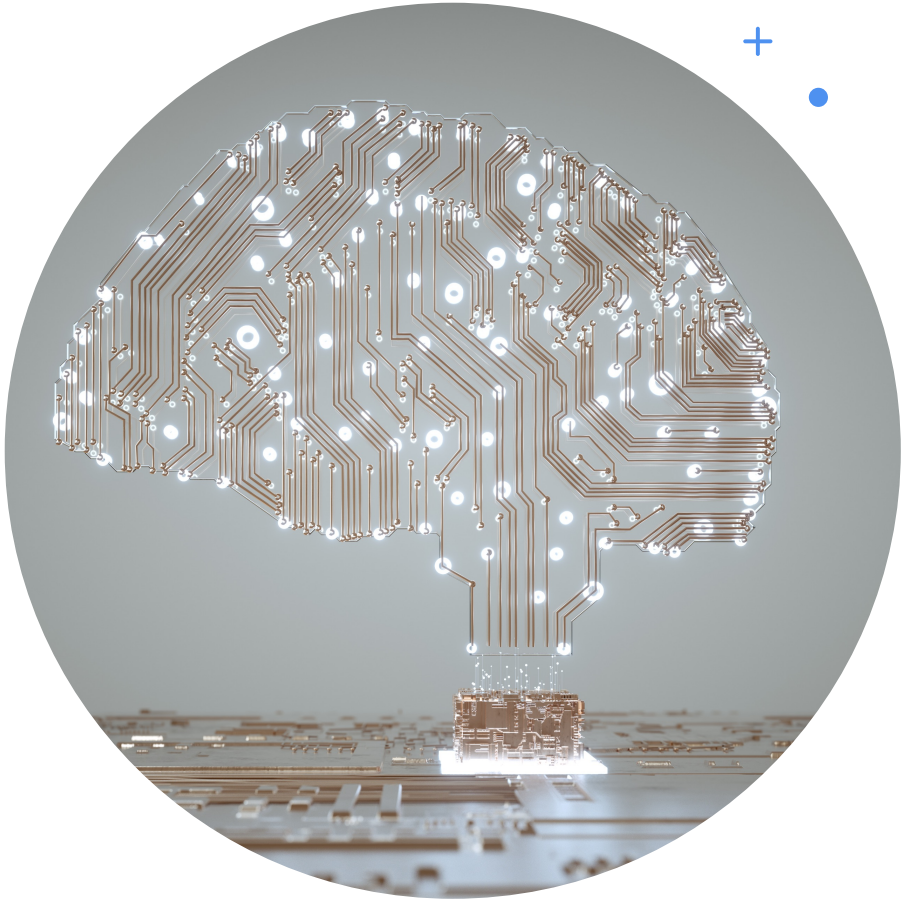
- Assisting Airbnb hosts in Amsterdam to set appropriate price for their listings

Problem??

Currently there is no convenient way for a new Airbnb host to decide the price of his or her listing. New hosts must often rely on the price of neighbouring listings when deciding on the price of their own listing.

The Solution

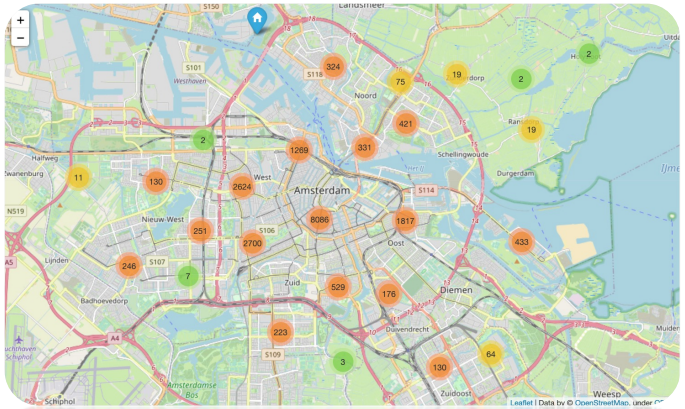
- A Predictive Price Modelling tool whereby a new host can enter all the relevant details and the Machine Learning Model will suggest the Price for the listing.

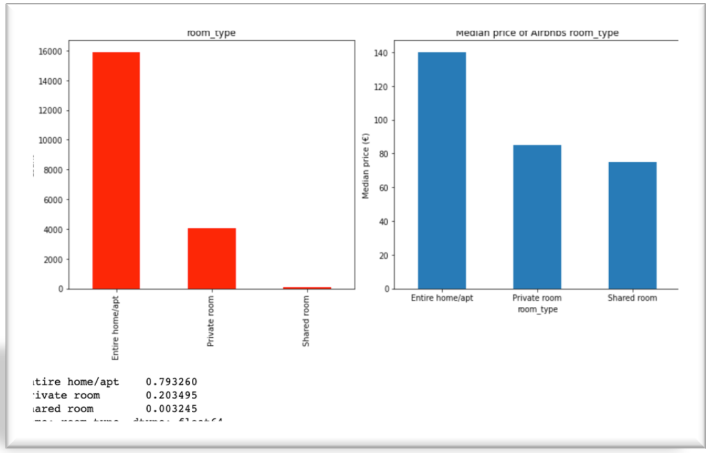


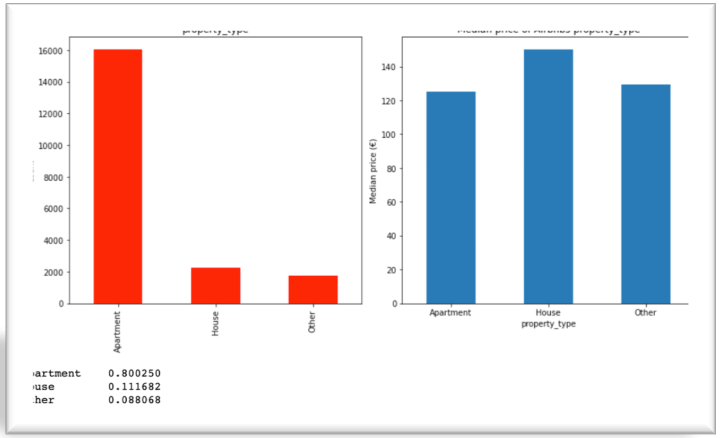
Overview

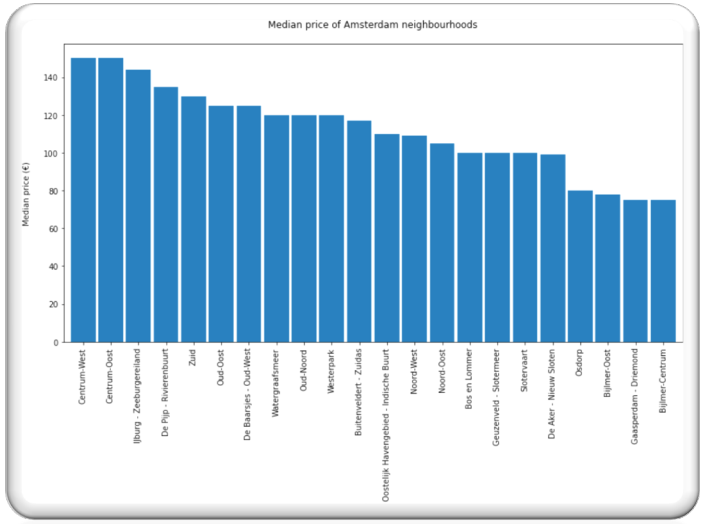
The genuine challange involved the following steps,

1. **Exploratory Data Analysis:** *Explore the various features, their distributions using Histograms and Box-plots*
2. **Pre-processing and Data Cleaning:** *Normalisation, filling missing values, encoding categorical values*
3. **Feature Selection:** *Study the correlation with response variable (Listing Price) and determine which features are most useful in predicting the price.*
4. **Model Fitting and Selection:** *Training different models, tuning hyper-parameters and studying Model performance using Learning Curve.*
5. **Model Serving:** *In order to deploy and serve Model predictions using REST API*



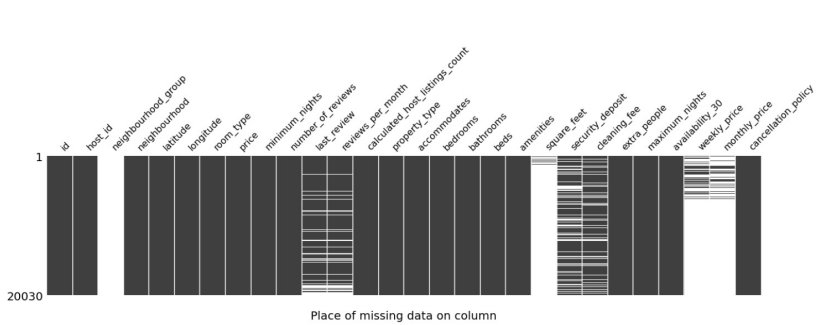




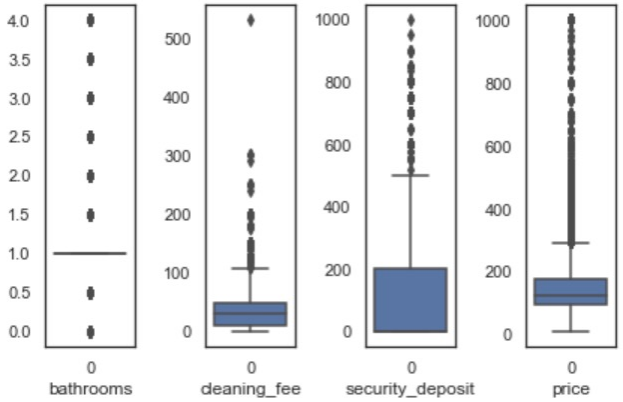


Exploratory Data Analysis

- Inner Amsterdam neighbourhoods have significantly more listings than outer Amsterdam neighbourhoods.
- If a property is rented as entirely the price is getting expensive and the cheapest one are shared rooms.
- House is the most expensive property type in Amsterdam
- Centrum-West and Centrum-Oost are the most expensive area - Centrum is a famously expensive area to live in the Amsterdam.




```
Percentage of the missingness by column
bedrooms              0.039940
bathrooms             0.049925
beds                  0.034948
security_deposit     30.783824
cleaning_fee         18.117823
weekly_price         85.806291
monthly_price        92.206690
```



Pre-processing and Data Cleaning

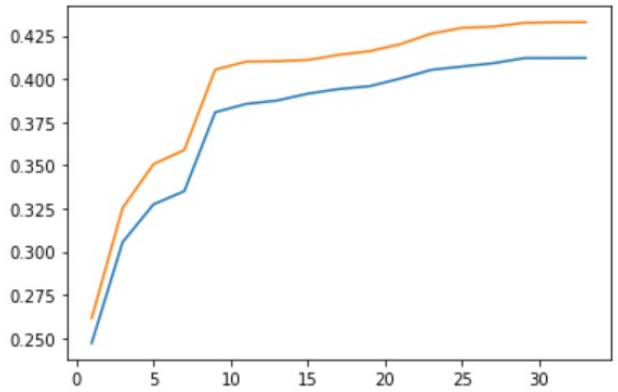
- Columns that contains majority of null entries are not counted. The remaining columns are replaced NULL values with a median value.
- Outliers , unusual values in the dataset , are dropped to prevent bias on predictions.

Feature Selection

- 2 different training dataset was created by using SelectKBest and Correlation heatmap manually.

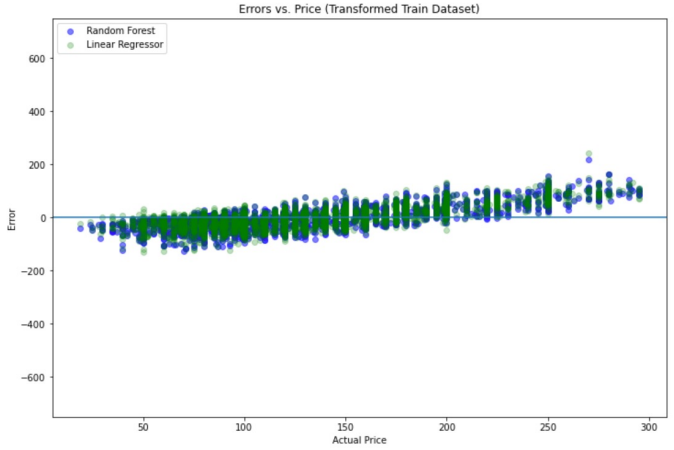
- Number of Accommodates and bathrooms, extra fee(cleaning fee), property type and neighborhood are some of the related features with the listing price and it is also demonstrated by the plots in exploratory data analysis.

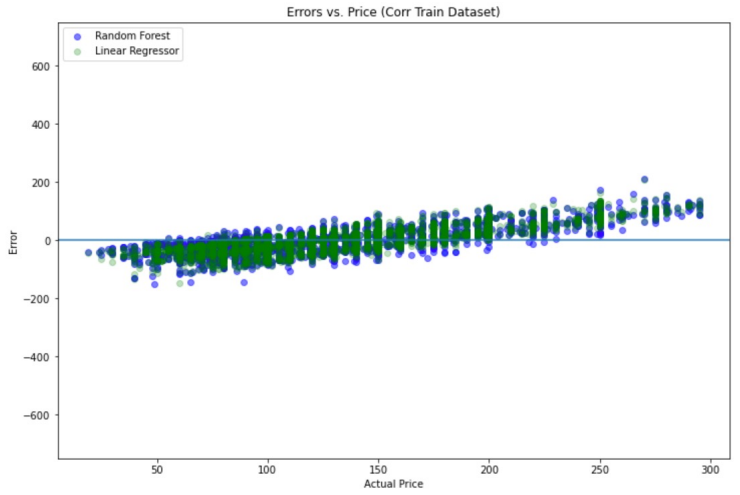
SelectKbest, f_regression

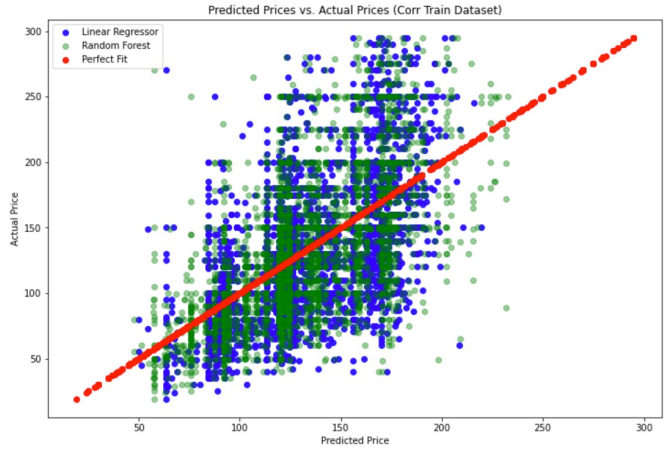


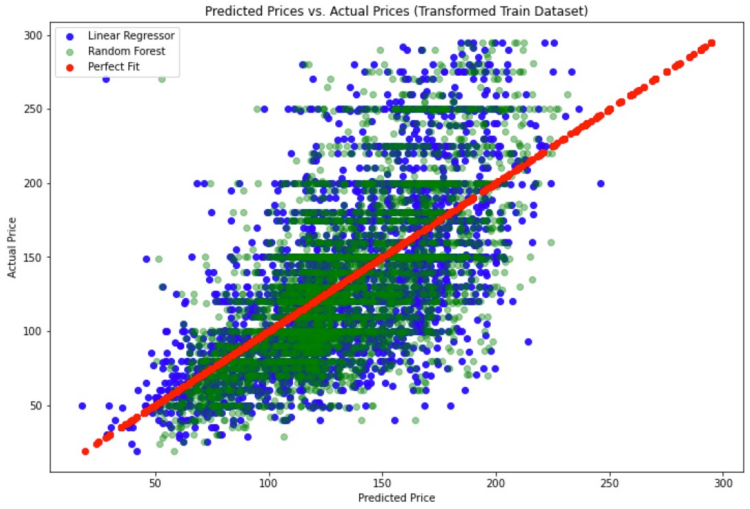
It is obviously seen in the plot above max score is reached around 25 features.

Corr Heatmap









Model Fitting and Selection

- Using Machine Learning Alghoritms, I applied 5 different models from Scikit-learn library i.e. RandomForestRegressor, Linear Regression, DecisionTreeRegressor, Lasso and SVR on both training datasets with default and tuned hyperparameters. The highest R2 score is 0.44 out of 1.00 with the tuned Random Forest Regressor.

- Even though reducing the features and selecting the best features manually and creating the Corr training dataset, r2 score could not be got higher. It obviously shows that as much as related features are selected despite correlated low with the listing price, also gets better predictions on modelling.

Model Serving

1- Saving the Trained Model

Pickle phyton module would be used to convert a phyton object to a bitstrean and allows it to be stored to the dosk and reloaded at a later time. It also provides a good format to stroe machine learning models provided that their intended applications are built-in python.

2- Developing a Web Service

FLASK would be used as a web service, It is the commonly used lightweight framework for deceloping the web services in Python. After building the web service , Docker would be used to containerize the application. It also works well with the modern CI/CD workflows.

3- Deploying the application

After containerization using the Docker, either Heroku or AWS would be used for deployment. Heroku is flexible and easy to use that offers lots of services and tools to speed up the development and helps avoid starting everything from scratch.

4- Monitoring and Logging

Qualdo would be using for monitoring the applied machine learning model performance. It has some nice, basic features that allow you to observe your models throughout their entire lifecycle. It is also compatible with Azure, Google Cloud Platform or AWS.