New York City Airbnb Rental Price Prediction Model

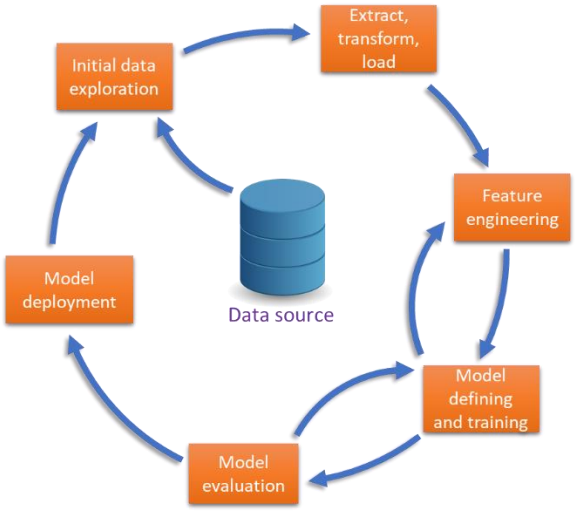

Figure 1. Model Architecture

Data Source and Use Case

- **Dataset**

The dataset consists of Airbnb listings and metrics in New York City (NYC) in 2019. Information about the hosts, their geographical locations, reviews and availability of almost 50,000 listings are provided in the data file. Also provided are the accommodation type, minimum number of night and the price per night. This public dataset is available at: https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data

- **Use case**

Airbnb is a widely known platform that brings homeowners offering affordable, short-term accommodations and trip goers looking for accommodations together. Either for a homeowner planning to start an Airbnb rental business or an individual looking for a place to stay, rental price is the number one factor. In this project, we will develop and training a machine learning model that predicts the price of Airbnb rentals in NYC.

Initial Data Exploration

Data exploration was done using Pandas APIs. Pandas APIs are fast, flexible, and easy to use and are designed specifically for data exploration, analysis, and manipulation. Data exploration step potentially helps in deciding what features in the data are relevant to the task at hand.

Extract Transform Load

- **Feature extraction**

In this step, feature extraction was performed using Pandas. Our feature selection choices here were partly based on intuition and partly on correlation with the rental price. For instance, features such as *id, host id,* and *host_name* were removed because, intuitively speaking, they should not have any impact on the rental price. *last_review* was removed due to too many missing data (one-third of the entire column), while *review_per_month* was removed because it's practically the same as *number_of_reviews*.

- **Outlier removal**

To improve prediction accuracy, price outliers in the training dataset were identified and removed. After this step, there was a dramatic improvement in the model performance.

Feature Engineering

- **One-hot encoding**

Non-numerical categorical features were transformed using the pandas DataFrame.get_dummies API. There are 3 non-numerical categorical features in the dataset that are critical to price prediction. These features were one-hot encoded to be included in the training set.

- **Feature creation**

In addition, in the attempt to improve model performance, we created and added another feature, *crime_per_capita,* using NYC's population and crime datasets obtained from external sources. However, there was no notable change in model performance after this step.

Model Defining and Training

We trained and tested scikit-learn models as well as a deep neural network model in this project.

- **Scikit-learn based models**

These were chosen because our training dataset is relatively small, roughly 45000 rows and 235 columns, suitable for Scikit-learn models. We defined and tested 2 models and picked the one with the best performance. First, we tried a linear regression model because it is simple and fast. And then, we tested an Extreme Gradient Boost (XGBoost) regressor, which is based on ensemble learning. XGBoost was picked because our training data are mostly categorical (228 out of 235 features) consisting only of 0s and 1s.

- **Deep learning model**

In addition to the Scikit-learn based models we also trained a deep learning model for performance comparison. The choice of the number of nodes as well as the depth of the deep learning model is purely empirical because using more than 3 layers did not improve the performance, neither did using more nodes. Other available optimizers such n*adam, adamax* and *RMSprop* also work as well as the *adam* optimizer used. One node is chosen in the output layer for regression and *relu* activation is used because **price** has 0 lower bound and no upper bound.

In general, these two techniques gave similar model performance based on the R-squared and mean absolute error evaluations. Consequently, of all the models trained and tested, the XGBoost regression model was picked because not only was it better in performance and more reliable than the linear regression model, training and evaluation was faster than the deep learning model.

Model Evaluation

- Since the task here is a typical case of regression, we used mean absolute error for model training evaluation and R-squared score for model prediction evaluation.
- When compared, our price predictions correlate strongly with actual listing prices and predictions are made with up to 60% variability of the input metrics accounted for, based on the cross-validation evaluation.

Model Deployment

Our model was deployed as a data product on an interactive geographical map plot

Consequence

Although, there is room for improvement given more data, nonetheless, our Airbnb rentals price predictions are quite reliable. This model in its current form could aid decision making for people looking to start the Airbnb hosting business in New York City as well as those who want a place to stay.