New York City Airbnb Open Data Regression Problem

Matteo Bianco
Politecnico di Torino
s300781
s300781@studenti.polito.it

Filippo Grobbo
Politecnico di Torino
s305723
s305723@studenti.polito.it

***Abstract*—In this report we introduce a possible approach to the *New York City Airbnb Open Data* regression problem. In particular, the proposed approach consists in applying label encoding to categorical variables. This choice was made after discovering patterns in data, which can let us classify the categories of some features into an ordered ranking. The proposed approach obtains overall satisfactory results even though correlations among features and the target variable are very weak.**

I. Problem Overview

The proposed competition is a regression problem on the *New York City Airbnb Open Data Dataset*. The aim is building a regression model able to identify the price of an Airbnb facility, given different informations related to the listing.

The dataset is divided into two part:

- a development set with 39116 samples, each one characterised by 15 features and a target variable that is the price.
- an evaluation set consisting of 9779 samples with the same features except for the price that should be predicted.

Features of both parts are distinctive attributes of a typical listing, such as the latitude and longitude of the place, type of room, number of reviews.

Making few statistical analysis we got a better understanding of the problem we were facing. In particular we decided to take a look at the distribution of features and their correlations using an heatmap. At first sight we noticed that correlations among different variables and price are not so high. This made us think that we would not achieve optimal results, as found in the end. To obtain additonal information about importance of features characterised by repeated categories, we decided to analyze distribution of the mean price with respect to the unique values (see Figures 1, 2, 3). This analysis turned out to be the starting point for more robust results.

II. Proposed Approach

A. Data Preprocessing

Looking at the meaning of the features, we decided to perform some preprocessing such as:

- Feature Selection
- Handling of missing values
- Label Encoding

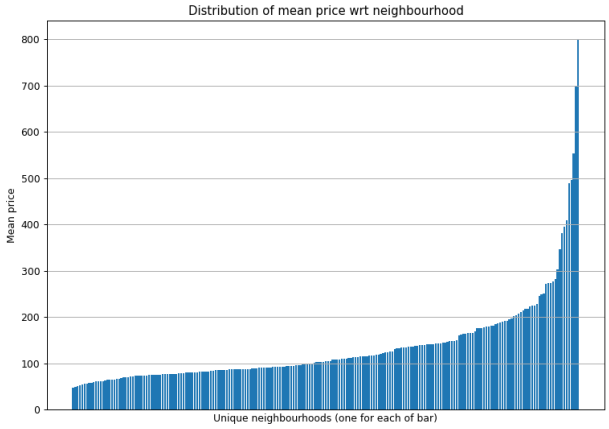

Fig. 1. Features importance of unique values of *neighbourhood* attribute

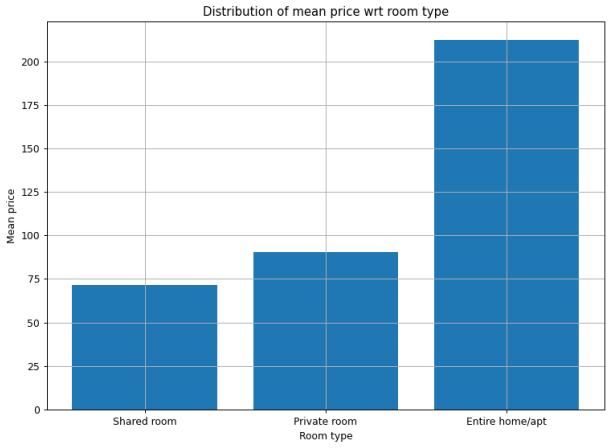

Fig. 2. Features importance of unique values of *room_type* attribute

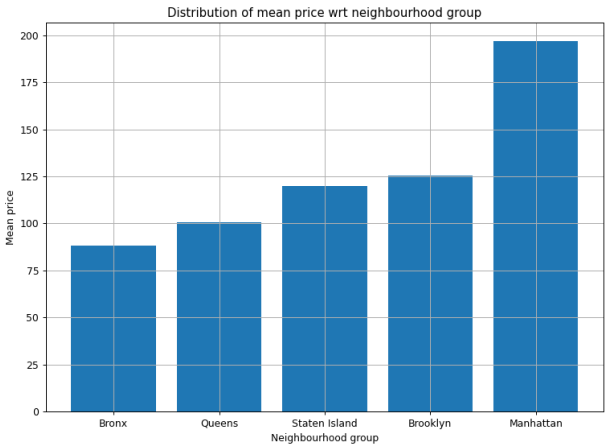

Fig. 3. Features importance of unique values of *neighbourhood_group* attribute

The reduction in the number of features was choosen to simplify the model. Initially this step was guided by our intuition and we discarded features associated to name of the house, host name, data of the last review, id of the house, host id and neighbourhood. Essentially these values were in one-to-one relation with each sample so we thought that there should not be neither correlation nor price dependence. Actually this was partially true because, for example, the distribution of mean price with respect to neighbourhood has a clear association, depending on the neighbourhood value (see Figure 1). The same can be deduced for the attributes *room_type* and *neighbourhood_group* (see Figure 2, 3). For these reasons and in order not to increase too much the dimension, we decided to perform a Label Encoding instead of One Hot Encoding.

Finally, since there were observations with 0 reviews but with a corresponding NaN in the attribute *reviews_per_month* we decided to fill these missing values with zeroes, as one would intuitively think.

B. Model Selection

The following algorithms have been tested:

- Polinomial Lasso of order 2: at first sight we decided to start from a polynomial regression with regularization technique. We did this because from our preliminary analysis we do not think that there could be a linear relationship between features and price. In addition to this, to avoid overfitting, we choose Lasso regression in order to assign 0 to some coefficients and decrease the complexity of the model.
- Random Forest Regressor: continuing along the path of avoiding overfitting we performed Random Forest Regressor with default parameters. In a 80/20 hold-out setting the performace clearly exceeded the one of Polinomial Lasso as it could be seen in Table I. However, we did not know whether the results were due to a lucky split between train and test set. Therefore, we applied a 5-fold Cross Validation to both algorithms and observed again that average r2 score between all 5 test sets was better in the Random Forest case (see Table II). At the end we choose to focus on this model.

TABLE I
REGRESSORS HOLD-OUT COMPARING

Regressor	r2 score
Polinomial order 2 Lasso	0.1124
Random Forest Regressor	0.2127

TABLE II
REGRESSORS 5-FOLD CROSS VALIDATION COMPARING

Regressor	average r2 score
Polinomial order 2 Lasso	0.124
Random Forest Regressor	0.148

C. Hyperparameters tuning

We decided to fix hyperparameter *max_features* to the value *sqrt* and we ran a grid search with 5-fold cross validation using all the possible combinations defined by the values in Table III:

TABLE III
HYPERPARAMETERS GRID SEARCH

Parameter	Values
n_estimators	{90, 130}
criterion	{squared_error, friedman_mse}
max_samples	{0.8, 0.9}

III. RESULTS

A. Hyperparameters

The best configuration was found with {*n_estimators* = 130, *criterion* = *squared_error*, *max_samples* = 0.8} and a corresponding r2 score ≈ 0.2170.

We trained this model on all the development data and than used it to predict values of the evaluation set. We obtained a public score MSE[1] = 30949.931.

B. Heatmap

Plot in Figure 4 repesents the geographical distribution of prices. Since we know that development and evaluation set have the same distribution of data, we would expect prices predicted for the evaluation set to be distributed in the same way. We can state that our regression has worked pretty well, since areas where yellow (or red) color predominates are quite the same in both plots.

[1]Mean Squared Error

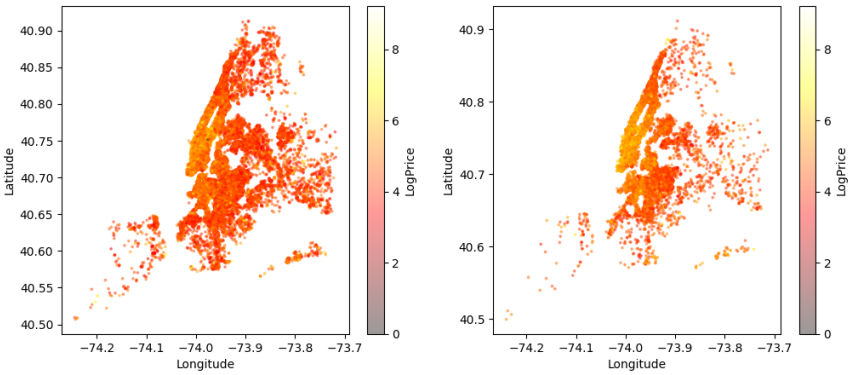

Fig. 4. Heatmap of prices in the Development (left) and Evaluation (right) set.

IV. Discussion

Considering features importance in our best model, we discovered something counterintuitive. As we expected, a great contribute ($\approx 37,1\%$ of feature importance) was given by geographical features (*neighbourhood_group*, *longitude*, *latitude*, *neighbourhood*). At the same time though, also feature *id* was considered very important by the model ($\approx 12.2\%$ in feature importance), even if we initially decided to discard it, since we thought it could not be significant. This fact may be showing that our model could be improved by not discarding any feature of the data.

The key intuition that made us achieve a strong result was the encoding of the features *neighbourhood*, *room_type* and *neighbourhood_group*. Initially we performed a *One Hot Encoding* not to introduce a bias in the learning algorithm, but our reaults were very poor. Proceeding in the analysis we thought that with a *Label Encoding* we could classify the geographical areas and room types by their "luxury" in order to give our algorithm more informations about the data. This turned out to be a powerful intuition which improved the performance of our model.

The following are some other aspects that we think should be studied in more detail, since they may further improve the results:

- Try using an SVR[2] model instead of a Random Forest Regressor since the feature space is not very big (12 dimensions) and Support Vector Machines are usually a very strong type of learner
- Perform a more accurate grid search of the hyperparameters

Anyway, the results obtained are already very good, especially considering the low values of correlation (both Pearson and Spearman) that data have with prices, as we found in the first explorative analysis.

[2]Support Vector Regression