New York City Airbnb listing price prediction

Vladimir Andrianov
student
Taltech
Tallinn, Estonia
vlandr@ttu.ee

***Abstract* — Making accurate predictions is the first thing naturally emerging into the peoples mind when the machine learning or data science is mentioned. Machine learning is the best fit for the predictive analytics and construction of the statistical models. Airbnb is a wide-spread housing rental platform with 272 million bookings made in 2019 [1]. Airbnb is investing many resources into data science and supports this field by driving internal knowledge into the public domain, they share data sets and release open-source platforms for the public use. The data set used for this research has been acquired from the publicly available data set on Kaggle made from open Airbnb data.**

I. INTRODUCTION

Data set used for the research contains the data for Airbnb New York City offered rental housing options. This data set includes rich set of attributes such as price, number of reviews and the housing object neighbourhood. From the rich set of attributes arises a natural question, are there any direct correlations or non-linear dependences in the data, can the price be predicted based on these and can it be done effectively? Price prediction model for the housing objects for the rental housing platform might be as useful as risk-prediction models and price calculators for the insurance companies. Airbnb receives money when people rent a property through their platform, having the best deal in least possible time range and highest prance is in interest of both customer and the company. The price prediction model would allow company to give approximate price for the property even before it was published, so the company can provide basic aid for the unexperienced property owners willing to rent out their property by giving them some reasonable price ranges and hints on what is people looking for and what should be improved. Having balanced offers would result in achieving a better result on the market, meaning more profit for customer and company.

In order to use data effectively it must be gathered, processed, sorted and stored in a computer readable format. The NYC data set has a well-defined data label, definition of the data cells and attributes are nicely pre-processed, which is very good for the classification and machine learning models such as linear regression.

To analyze the data and make predictive models and find dependencies and correlations the proper data environment should be used. The "Anaconda", the open-source data science toolkit has been chosen for the research, as this is open-source toolkit with built-in R/Python support and the toolkit comes with the built-in package manager and "Jupyter", the "environment for working with notebook containing code".

II. RELATED WORKS

A. *Predicting the rental value of houses in household surveys in Tanzania, Uganda and Malawi: Evaluations of hedonic pricing and machine learning approaches*

Housing value can be viewed as a major rate of the welfare status of household and wealth in general, therefore an accurate estimation is important to obtain the value of housing for the householding surveys. Rental prices are a basis for the housing value prediction, accordingly to the research normally an Ordinary Least Squares (OLS) method is used for the predictions while there are better ways available for this "The literature shows that Machine Learning (ML) methods, shown to uncover generalizable patterns based on a given data, have better predictive power over OLS applied in other valuation exercises" [2]. The different machine learning methods were used, such as Boosting, Bagging, Forest, Ridge and LASSO were shown the best for the house values prediction for all countries and all the years, while the Tree regression underperformed compared to OLS models on the same data [2]. Machine learning models provide a better alternative to standard methods, while these might require better computing power and big data sets.

Hedonic pricing approach and Bias-variance tradeoff were selected for review and the future comparison with the machine learning models with Ridge regression, least absolute shrinkage and selection operator regression, Tree regression, Bagging, Random forests and Boosting methods. The data used for the research was also explored and the summary table of data has been provided. After the approaches explained and compared, the models were built to get the experimental results for in-sample prediction and out-of-sample prediction and table of results on table 1 was provided and It was concluded that Boosting method is the most accurate while OLS regression provides the most consistent and reliable predictions, and the current work can be extended by including more data and should include evaluations and comparisons.

	Uganda		Tanzania		Malawi		Overall performance score
	2010	2012	2014	2016	2014	2016	
OLS*	1.00	1.00	1.00	1.00	1.00	1.00	-
Ridge	0.94	0.96	0.99	0.97	0.95	1.01	83%
LASSO†	0.89	0.92	1.00	0.88	0.95	1.01	83%
Tree	0.84	0.83	1.11	1.60	1.11	1.05	33%
Bagging	0.78	0.80	0.98	1.28	0.88	0.91	83%
Forest	0.82	0.88	1.00	0.81	0.84	0.88	83%
Boosting	0.88	0.85	0.96	0.96	0.87	0.91	100%

*OLS = Ordinary Least Squares
† LASSO = Least Absolute Shrinkage and Selection Operator.
https://doi.org/10.1371/journal.pone.0244953.t011

Table 1. Out-of-sample prediction performances based on standardized mean squared errors of predicting housing rental values by country and by year without accounting for spatial autocorrelation. [2]

III. DATASET

Dataset is part of the data set available in the public domain published by Airbnb. The New York City Airbnb Open Data dataset is providing a rich set of attributes and remarkably well-defined values with description of every column and meaningful column names and usability measured as 10 of 10 on the Kaggle [3].

There are 48895 rows of data available in the comma-separated values file and data is split into 16 columns, available attributes are: id, name, host_id, host_name, neighbourhood_group, neighbourhood, latitude, longitude, room_type, price, minimum_nights, number_of_reviews, last_review, calculated_host_listings_count, availability_365, reviews_per_month. Some columns in the dataset represent categorical data, these are the neighbourhood_group and room_type columns as shown on "Figure 1".

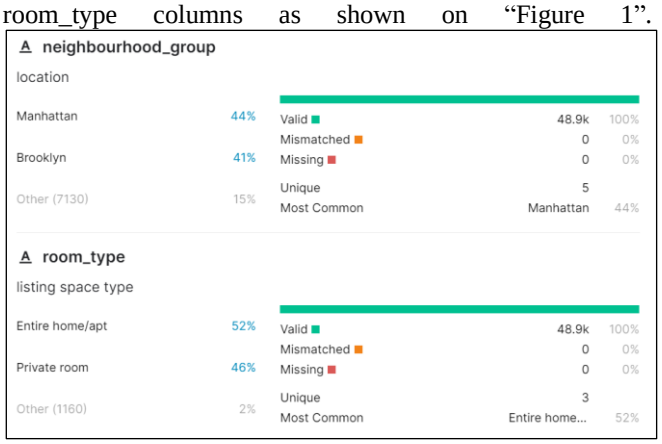

Figure 1. Overview of neighbourhood_group and room_type attributes data [3].

48895 data entries with 16 columns are a reasonable size to analyze data, split data into train and test datasets and apply different models. The price attribute distribution instantly shows there are clearly some outliers in price data "Figure 2", so the data requires further analysis and processing. Mean value of 153 clearly demonstrates that outliers can be dropped from the dataset before constructing a model based on price data.

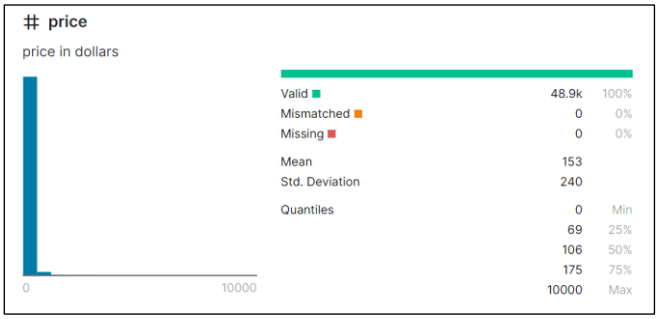

Figure 2. Price distribution [3].

The "id" column with 48896 values should be dropped too as this is a listing id which is a service data used to distinguish the listings, thus there should be no data correlating with this, and if there is some dependence this should not be counted as this doesn't represent any real-world value.

There are missing values in column "reviews_per_month" saying that there was no feedback yet provided "Figure 3". There are also some missing values in "host_name" column, this field can be omitted and "host_id" can be used instead.

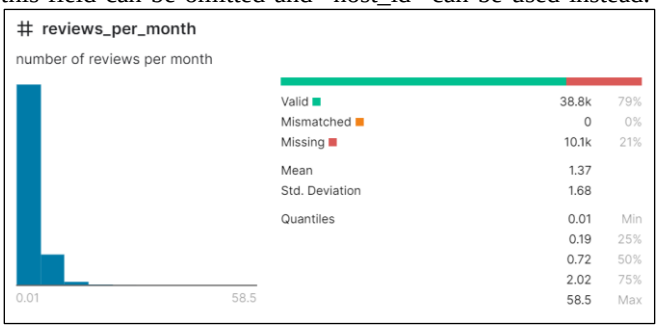

Figure 3. Missing values in reviews_per_month column.

IV. DATA ANALYSIS

The price chart discussed in the Dataset part clearly illustrated that there are outliers in the data. In order to further analyze the distribution, the outliers should be dropped. Total 0.61% of values are outliers with value over 1000 "Figure 4".

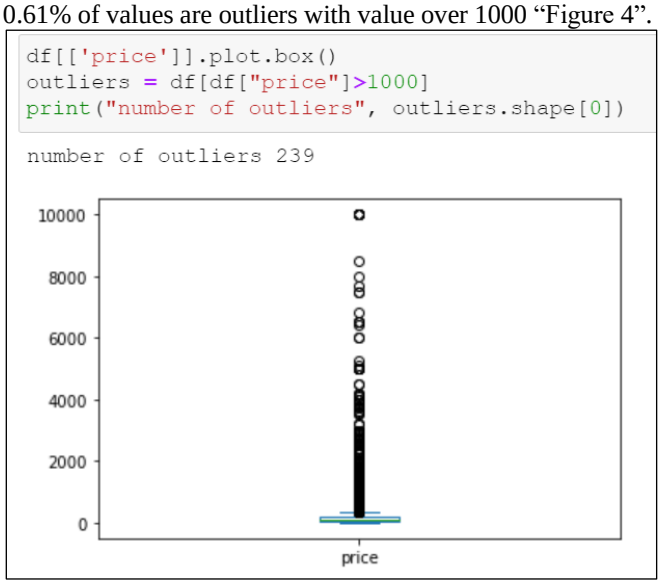

Figure 4. Price outliers.

The price distribution chart without price outliers (price > 1000) clearly demonstrates that most of the prices are located within the mean value in range from 100 to 200 "Figure 5".

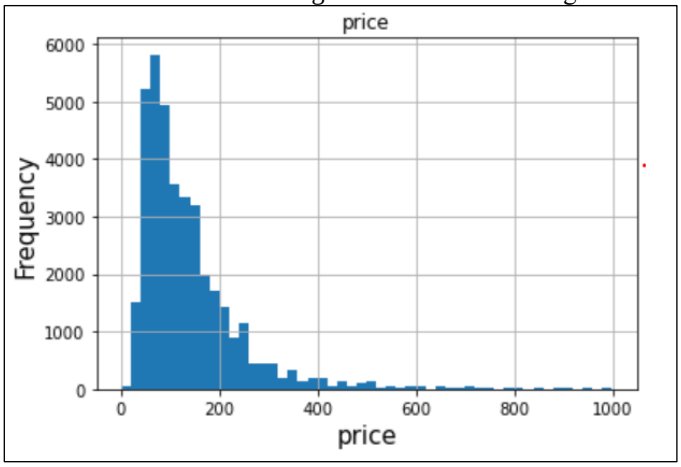

Figure 5. Prices with outliers dropped.

Non-functional columns as id, name, last_review, host_id were removed from the dataset. Any entries containing price outliers (value>1000, mean 153), missing reviews per month, number of reviews outliers (value>200, mean 23.3), reviews per month (value >10, mean 1.37) or minimum nights value outliers (value>365, mean 7.03) were dropped from the dataset decreasing the dataset to 38015 entries “Figure 6”.

```
df=df[df['price'] < 1000]
df=df[df['minimum_nights']<=365]
df=df[df['number_of_reviews']<=200]
df=df[df['reviews_per_month']<=10] # Drops missing too.
df.drop(['id','name','host_name','last_review', 'host_id'],
axis=1, inplace=True)
print(df.shape)
```

```
(38015, 11)
```

Figure 6. Processed dataset.

Geometrical listing data such as longitude and latitude were used to construct heatmap to visualize the listings distribution on the New York City map “Figure 7” (0.2: 'blue', 0.4: 'purple', 0.6: 'orange', 1.0: 'red'). The heatmap shows that longitude and latitude are set correctly too as there are no points located on the water surface and most of the listings are located in the Manhattan and Brooklyn neighborhoods.

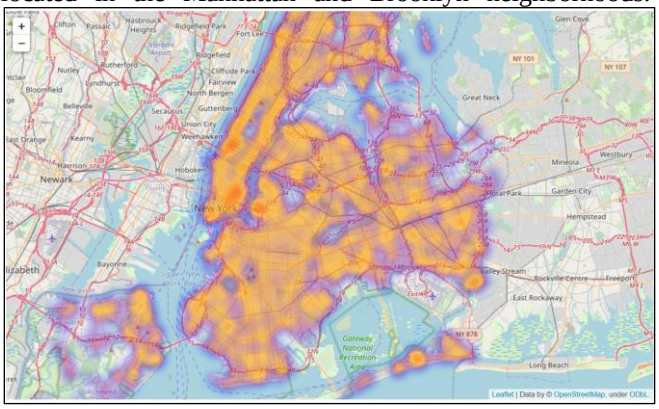

Figure 7. New York City listings heatmap.

The Correlation matrix was built to inspect dependencies between the data and price. The correlation plot matrix built for all leftover attributes, the matrix was split into two matrixes, both include price to inspect the correlations with it. To build the correlation matrixes the non-numerical categorical values were transformed into encoded into numerical indexes. Correlation matrix was sliced into two matrixes “Figure 8” and “Figure 9” for the better visibility.

Figure 8. First correlation matrix.

Figure 9. Second correlation matrix.

Correlation matrixes “Figure 8” and “Figure 9” show very weak positive price correlation with neighbourhood_group, neighbourhood, minimum_nights, availability_365 and calculated_host_listings_count. Combined correlation matrix of most price correlated attributes shows that availability_365, calculated_host_listings_count and minimum_nights are interrelated, neighbourhood_group and neighbourhood are interrelated too “Figure 10”. The strongest correlation is with neighbourhood.

Figure 10. Attributes interrelationships.

The strongest correlation from correlation matrix, the price dependency on neighborhood has been proven on the real data, most expensive listings are mostly located in Manhattan “Figure 11”.

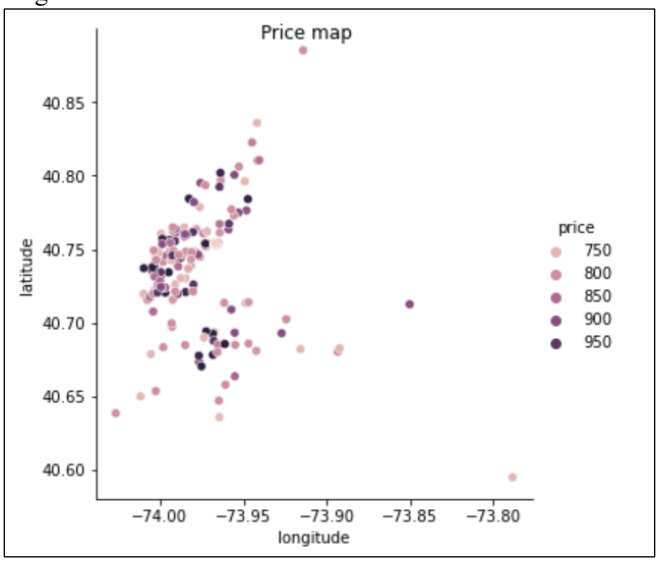

Figure 11. Price map.

V. CLASSIFICATION

Once the data has been analyzed and sorted, the classification and machine learning can be applied to the dataset. From data analysis it's clear that there are categorical attributes as room type, this is a categorical value which has only 3 variants: "Entire home/apt", "Private room", "Shared room". Room type is a good candidate for the classification problems. Price map "Figure 11" shows that there is a price and geolocation dependence. Price map shows that there is a dependence between "price", "room_type" and "neigbourhood_group" features and the listings can be classified by these attributes.

room_type neighbourhood_group	Entire home/apt	Private room	Shared room
Bronx	121.207237	57.207364	56.930233
Brooklyn	165.154173	70.866260	45.484429
Manhattan	213.771829	104.422634	78.293447
Queens	137.936183	65.696169	45.081081
Staten Island	125.738255	57.484076	57.600000

Figure 12. Airbnb listings classes.

The table above "Figure 12" proves existence of the dependency between price, room type and neighborhood here, price and neighborhood were used to classify the room type. Theory says the bigger apartment is and the closer it to the center, the more expensive it is, and cheapest rooms are shared. Data has been split into "price" and "neighbourhood_group" array to "X", "room_type" to Y datasets, data was cut into 70% to 30% volume chunks for train and test datasets. The naive bayes, decision tree and support vector machine methods were applied to the data. The classifier test cross-validation function was introduced. The k-fold cross-validation with 10 folds is used to measure the performance of the models, k=10 becoming a 10-fold cross validation is a commonly used in the field of machine learning [4].

A. *Naive bayes classifiers*

Gaussian naive bayes is useful when working with continuous features in decimal form, while multinomial naive should be used for the features which represent for example, the number of occurrences of a term or its relative frequency and Bernoulli naive bayes can assume only two values, for example, true or false. Gaussian and multinomial bayes were chosen for the train data consisting of "price" and "neighbourhood_group" to classify the "room_type" feature. The Gaussian and multinomial classifiers resulted into 72.3% and 72.0% accuracy on 10-folds cross validation "Figure 13". Naive bayes gives a good result on the "room_type" class.

```
Accuracy GaussianNB bayes 0.7231849720264355
Time for 10 fold CV on bayes is: 0.054024696350097656
Accuracy MultinomialNB bayes 0.7205197927758903
Time for 10 fold CV on bayes is: 0.04199576377868652
```

Figure 13. Naive bayes 10-folds cross-validation results.

B. *Decision tree*

The Decision tree has two functions to measure the quality of a split, Gini is intended for continuous attributes and Entropy is for attributes that occur in classes, based on data the Entropy has been selected for the decision tree measurement function. A decision tree provides a better accuracy of 8% more, resulting in 80.8% "Figure 14" compared to the naive bayes classifiers.

```
Accuracy decision trees 0.8083388548022695
Time for 5 fold CV on decision trees is: 0.07102847099304199
```

Figure 14. Decision tree 10-folds cross-validation results.

The "shared_room" class is extremely sparse, at low values for the depth of the tree (<5) and high values for minimum samples leaf (>50) there are no nodes with the result of "shared_room" in the resulting tree at all. The optimal parameters were chosen, at which the tree is not very complex and there is a node with "shared_room" class, the minimal tree with "shared_room" class has 31 nodes ($2n-1=31$, $n=5$) with shared room in left corner "Figure 15".

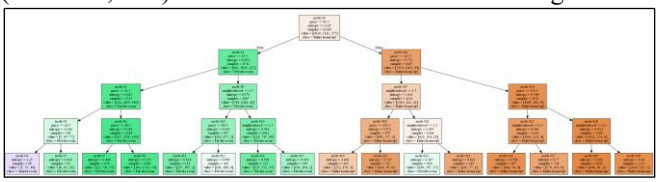

Figure 15. Decision tree nodes.

C. *Support vector machine (SVM)*

SVM accepts kernel function as a parameter, there are 4 kernel functions available: linear, polynomial, sigmoid, rbf. Kernel functions are specifying the core, this affects accuracy and execution speed. All the specified kernels were used, there are no results for the polynomial kernel as the calculations took too much time or were stuck. The experiment results "Figure 16" show that linear and rbf kernels provide the best accuracy of 80%, while linear kernel took 17 times more calculation time, and sigmoid was as fast as rbf kernel, but it provided poor result of 46% accuracy.

```
Accuracy sigmoid SVM 0.46
Time for 5 fold CV on sigmoid SVM is: 85.94s
Accuracy rbf SVM 0.8
Time for 5 fold CV on rbf SVM is: 74.77s
Accuracy linear SVM 0.8
Time for 5 fold CV on linear SVM is: 1312s
Accuracy sigmoid SVM no data
Time for 5 fold CV on sigmoid SVM is: no data
```

Figure 16. Kernels 10-folds cross-validation results comparison.

D. *Results and recommendations*

Classification method should be chosen accordingly to the available data distribution, size, format. Naive bayes works poorly for the abnormally distributed data, but it works fast. The decision tree is good for tasks where clarity and interpretability are important. SVM is the heaviest to compute, while it provides same accuracy as decision tree thus it should be applied only if necessary "Figure 17".

value	decision trees	Naive Bayes	SVM
acc	0.808450	0.710125	0.803872
time	0.072001	0.040026	160.056998

Figure 17. Classification methods results spreadsheet.

VI. Regression

For the regression the price feature is chosen. Price prediction is the initial problem of the research. The R^2 coefficient will be used to measure the model performance. Different regression models are applied to the dataset, these regression models are: linear, ridge, lasso, elastic-net and polynomial. Data has been analyzed and sorted in the previous chapter and is used for the further regression. The training and test sets were updated to predict "price" feature based on the "room_type", "neighbourhood_group", "latitude", "longitude" features.

A. *Linear regression*

Linear regression was applied to the existing data set and it resulted in very poor results of R^2=28% "Figure 18".

```
Mean Absolute Error: 54.05353334605293
Mean Squared Error: 8228.12922431871
Root Mean Squared Error: 90.7090360676306
relative error 0.3987589140787031
R^2 0.2794420828008861
time:  0.0020003318786621094
```

Figure 18. Linear regression results.

From the data it's clear that the prediction range is too wide as most of the values are distributed within the 0-400 range, while there are a few outliers within the 400-1000 range which affect the entire model "Figure 19".

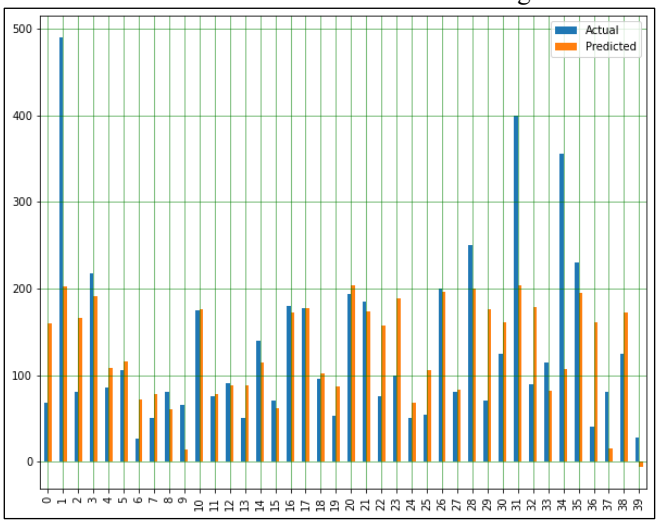

Figure 19. Outliers in data.

The drop of data outliers resulted in a significant improvement, improving the linear regression results and boosting R^2 from 28% up to 40%, reducing mean absolute error from 54 to 41 "Figure 20"

```
Mean Absolute Error: 41.0533891337769
Mean Squared Error: 3232.0780056666217
Root Mean Squared Error: 56.85136766751194
relative error 0.30285540753518014
R^2 0.40268532344464336
time:  0.0030007362365722656
```

Figure 20 Linear regression results in smaller range.

and improving the distribution chart as it doesn't have data outliers anymore "Figure 21".

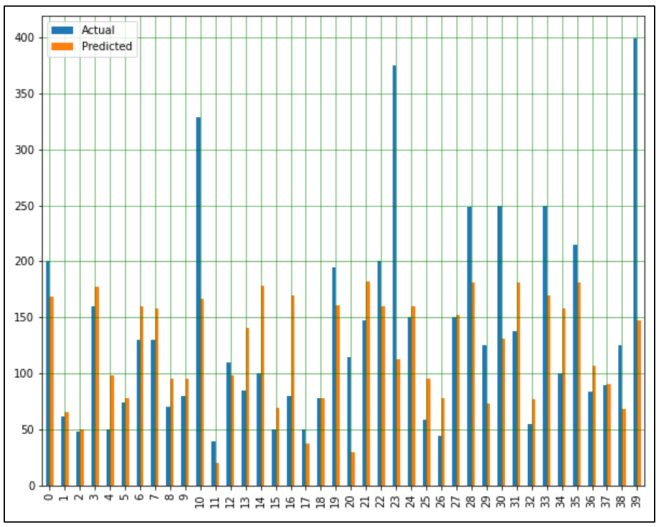

Figure 21. Distribution without outliers.

B. *Ridge regression*

Ridge regression has provided results similar to linear regression "Figure 22", in theory that should help when data suffers from multicollinearity which is not a case here.

```
Mean Absolute Error: 41.05208913242377
Mean Squared Error: 3232.4714553482204
Root Mean Squared Error: 56.854827898325574
relative error 0.30285540753518014
R^2 0.4026126106979544
time:  0.0030007362365722656
```

Figure 22. Ridge regression results.

C. *Lasso regression*

Lasso regression provides similar results to ridge and linear regressions with only difference in a bigger mean absolute error "Figure 23" meaning it made it worse.

```
Mean Absolute Error: 43.5168772546846
Mean Squared Error: 3502.803319775741
Root Mean Squared Error: 59.184485465160044
relative error 0.30285540753518014
R^2 0.35265305220956744
time:  0.0030007362365722656
```

Figure 23. Ridge regression results.

D. *Elastic-net regression*

Elastic-net similarily to lasso regressioon made the results significantly worse „Figure 24" with default parameters.

```
Mean Absolute Error: 49.87572487356842
Mean Squared Error: 4267.152975811097
Root Mean Squared Error: 65.32344889709282
relative error 0.30285540753518014
R^2 0.21139493072564886
time:  0.0030007362365722656
```

Figure 24. Elastic-net regression results with default alpha.

Elastic-net regression takes multiple parameters, while the most significant one is the alpha, that is a constant that multiplies the penalty terms. The default alpha value is 1 while passing 0.001 as alpha value has improved the model

by lifting the R^2 from 21% to 40% „Figure 25“ which is same as linear, lasso and ridge regressions provide.

```
Mean Absolute Error: 41.115367994940414
Mean Squared Error: 3241.711054804337
Root Mean Squared Error: 56.93602598359265
relative error 0.30285540753518014
R^2 0.40090505650187525
time:  0.0030007362365722656
```

Figure 25. Elastic-net regression results with 0.001 alpha.

E. Polynomial regression

The linear, ridge, lasso and elastic-net regressions resulted in average 40% R^2 and 30% relative error, which are not very good results, probably because of the of the non-linear dependence in the data. The polynomial regression works better on the data with non-linear relationship, it also takes degree as an input parameter, which stands for the maximal degree of the polynomial features. Polynomial regression with default settings provides a similar result to linear, ridge and elastic-net regressions “Figure 26”.

```
Mean Absolute Error: 41.05338913376738
Mean Squared Error: 3232.0780056666636
Root Mean Squared Error: 56.85136766751231
relative error 0.30285540753518014
R^2 0.4026853234446356
time:  0.0030007362365722656
```

Figure 26. Elastic-net regression results with default degree.

Experimental optimal value for the polynomial regression is 5 degrees, which significantly improves the regression providing the best results of all regression tested on this data. The R^2 has lifted up to 48% which is a very decent result compared to 40% provided with the linear regression.

```
Mean Absolute Error: 37.249304025055096
Mean Squared Error: 2833.7018813676873
Root Mean Squared Error: 53.232526535640666
relative error 0.30285540753518014
R^2 0.47630851738235147
time:  0.0030007362365722656
```

Figure 27. Elastic-net regression results with 5 degrees.

D. Results and recommendations

The linear, lasso, ridge and elastic-net regressions were introduced and applied to the data to get experimental results. Most these provide average result of 40% R^2 on default settings while lasso shows result below the average. Polynomial regression shows the best result of these all, but without custom parameters it provided average 40% R^2 result. From the results spreadsheet it’s also visible that the polynomial regression is ten times more heavy than linear, ridge, lasso and elastic_net regressions “Figure 28”.

	value	Linear regr	Ridge	Lasso	Elastic_net	Polynomial
0	R^2	0.402685	0.402613	0.352653	0.400905	0.476309
1	time	0.003001	0.001999	0.003000	0.003000	0.030000

Figure 28. Regression results.

Regressions might have shown a better result if more features would have been added to the training dataset. The model can be evaluated with more features and more customized parameters. Modifying the data and prediction range affects the aimed group prediction significantly.

VII. References

[1] D. Curry, “28 Amazing Airbnb Statistics You Should Know Before Booking”, Capital Counselor, May 20, 2021. Accessed on: Oct. 10, 2021. [Online]. Available: https://capitalcounselor.com/airbnb-statistics/

[2] W. T. Embaye, Y. A. Zereyesus, B. Chen, “Predicting the rental value of houses in household surveys in Tanzania, Uganda and Malawi: Evaluations of hedonic pricing and machine learning approaches”, Plos One, Feb. 11, 2021. Accessed on: Oct. 10, 2021. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0244953

[3] Dgomonov, “New York City Airbnb Open Data”, Kaggle, Jul. 18, 2019. Accessed on: Oct. 10, 2021. [Online]. Available: https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data

[4] J. Brownlee, “A Gentle Introduction to k-fold Cross-Validation”, Machine Learning Mastery, May 23, 2018. Accessed on: Oct. 25, 2021. [Online]. Available: https://machinelearningmastery.com/k-fold-cross-validation