Department od Economics, Management and Quantitative Methods
Università degli Studi di Milano
Data science and economics

Statistical Learning Project

Analysis and prediction models for New York City Airbnb

Report

Author:
Andrea Ierardi

Academic year 2019-2020

Contents

1

Abstract

The aim of the project is to analyse, develop prediction models and define data clusters from the "New York City Airbnb Open Data" from a Kaggle competition.
In particular, one part is focused on the development of predictive models to forecast house prices using these Supervised Learning technics:

- Linear Regression
- Decision Tree
- Random Forest
- Ranger Random Forest
- Neural Newtworks

For each of these, a comparison between the Mean Square Error and between all R^2 measure of all methods has been made to highlight which have the best performance. Also, for training all the models, a partition of the dataset in three parts has been applied: entire dataset, filtered by neighboorhood group and filtered for neighboorhood group and room type. In this way, it is possible to check the perfomance giving less or more features in input.

The second part is focused on the cluster and data reduction technics using these Unsupervised Learning technics:

- K-means Algorithm
- Clustering for mixed-type data
- Principal Component Analysis

2

Problem Definition and Algorithm

2.1 Two main Goals

2.1.1 Develop predictive models for price

First objective is the forecast of the prices given some input informations. This could be usefull for a lot of scenarios. For example from a AirBnB user point of view, he/she would like to get the houses more in line with his preference choice; or for a host point of view, where given the position and other information he/she could get information about the possible per day price of his house in New York City.

2.1.2 Define clusters and groups

Second objective is the definition of group or partition between the houses with different characteristics. For a user point of view could be usefull to have information about similar houses before the booking, giving in input some information about their preference. This could be also usefull after the booking for a suggestion analysis giving the information about last booked houses in NewYork or houses in similar cities around the globe.

2.2 Algorithms

2.2.1 Linear Regression

Linear regression is a linear approach to modeling the relationship between a dependent variable and one or more independent variables. Linear regression should be suitable, since there could be a linear relationship between the position and the price. In the city center there will be the expensive houses, while in the outskirts there will be the cheaper ones.

2.2.2 Decison Trees

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making.

2.2.3 Random Forest

Random forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees.

2.2.4 Ranger Random Forest

Ranger is a fast implementation of random forests or recursive partitioning, particularly suited for high dimensional data.

2.2.5 Neural Networks

2.2.6 K-means

2.2.7 Principal Component Analysis

3

Experimental Evaluation

3.1 Methodology

3.1.1 Data Inspection

The dataset is a Kaggle competiotion dataset, called the New York City Airbnb Open Data. It contains 48.000 data points for each different column. The dataset is structured with 16 columns:

- **id**
- **name**: name of the listing
- **host_id**
- **host_name**
- **neighbourhood_group**: location
- **neighbourhood**: area
- **latitude**: coordinates
- **longitude**: coordinates
- **room_type**: space type
- **price**: in dollars
- **minimum_nights**: amount of nights minimum
- **number_of_reviews**: number of reviews
- **last_review**: latest review
- **reviews_per_month**: number of reviews per month
- **calculated_host_listings_count**: amount of listing per host

- **availability_365**: number of days when listing is available for booking

It has been select only 5 of this features since these are the most important for determine the prices and also the different groups of clusters. These feature selected are: latitude, longitude, room type, neighbourhood and the price itself that will be compared with the prediction.

From the Figure 1 is possible to see the distribution of the price. The minimum price is 0 and the maximum price is 10000$. It is not possible to rent a house for free, so it is possible to filter the price with a price higher than 15$ and lower than 500$. It is possible that a luxury house cost a lot per day, but these value can not be consider in the model and can be consider as outliers. The reason is that the third quantile has a value of 175$ which is a far value from 10000$. It is also been checked the null and missing value and have not been found in the dataset apart from the reviews_per_month column. This feature has been not taken in account as possible one to the model training.

From Figure2, it is possible to see the distribution of all houses in New York City and the price. It can be noticed that the prices for the most part are in the range 0-500$ and only a low number of istances have a price greater than 500$. This picture is really informative.

```
     price
Min.   :    0.0
1st Qu.:   69.0
Median :  106.0
Mean   :  152.7
3rd Qu.:  175.0
Max.   :10000.0
```

Figure 1: Price summary

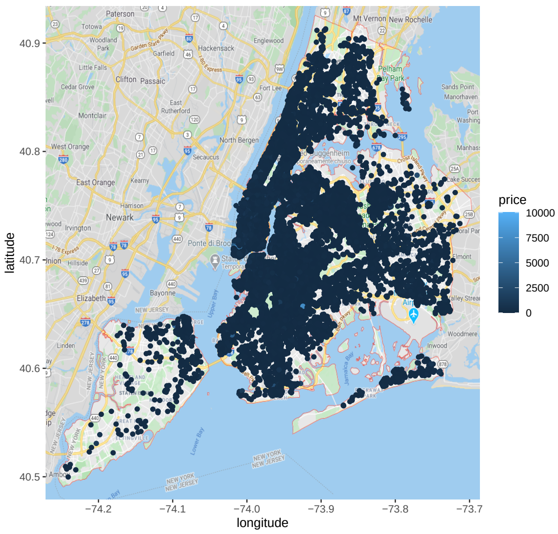

Figure 2: Distribution of all houses in NY colored by prices

3.1.2 Data Cleaning and Pre-processing

The dataset has more or less 48.000 data points for each column, so an important part of this work was the pre-processing since the large amount of data. Moreover, running different methods and algorithms on the entire dataset has an high computation cost. An important choice to made is that the user is able to discriminate which type of house has a particular interest. Then, it can be assumed that the user chooses which is the price range of interest and also the type of room. For the simplicity of the project we will focus on the Manhattan region for the popularity, a range of price from 15$ to 500$ and an entire apartment type of room. The project can be extended easily to the entire dataset, based on the user preference. Each variable choosed in this project have been rescaled to let the model perform and learn better, apart from latitude and longitude since the rescaling should not have a real meaning.
Also categorical variables have been rescaled assigning a numerical value to each category, resulting as factors.
The selected categorical features are: neighbourhood_group and room type.
The selected numerical features are: latitude, longitude and price.

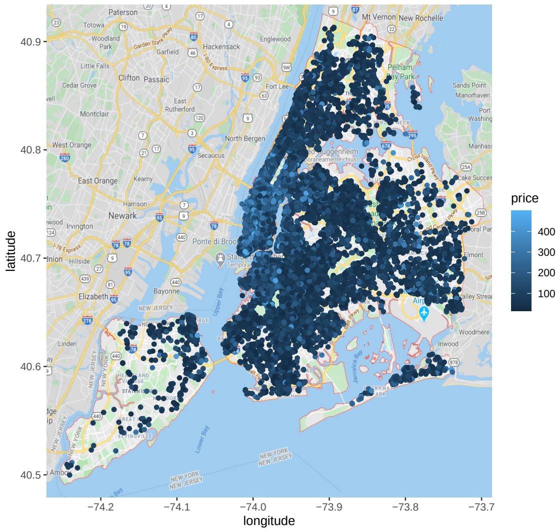

Figure 3: Distribution of all houses in NY of price between 15$ and 500$ per day

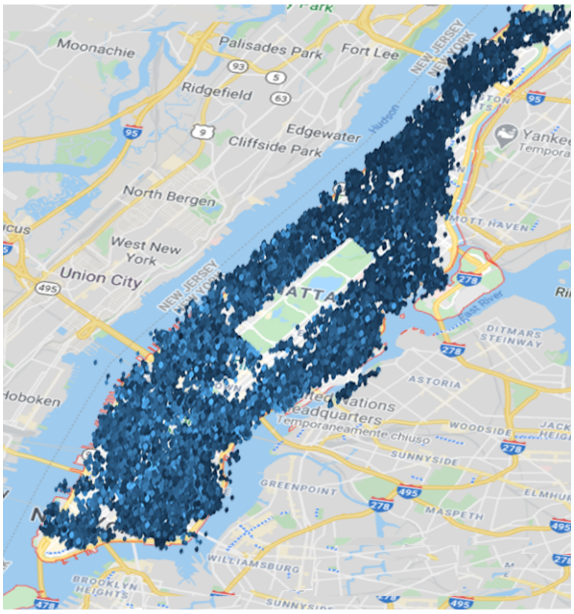

Figure 4: Approximated distribution map of all houses in Manhattan

3.2 Results

3.2.1 Linear Regression Results

For the Linear regression the model give different results based on the variables used.

Linear Regression selecting the Neighboorhood group

For semplicity, the tests have been only taken filtering for Manhattan data points, but changing the neighboorhood the results are similar. Results are acceptable (Figure 5), given a R^2 value of 0.4011 and a Mean Square Error, comparing prediction and test set, of 0.67.

Linear Regression selecting the Neighboorhood group and the type of room

As in the previous case the tests are for Manhattan and for Entire home/Aparment type of room.

The model output (Figure 6) a value of R^2 equals to 0.05953 which is low and a Mean Square Error, comparing prediction and test, of 1.23.

Linear Regression without filters

The model (Figure 7) obtain a R^2 value of 0.4031 which is acceptable and a Mean Square Error, comparing prediction and test set, of 0.59.

```
 === Linear Regression selecting the Neighboorhood group ===
Neighboorhood group = Manhattan

Call:
lm(formula = price ~ latitude + longitude + room_type, data = train_filtered[-1])

Residuals:
    Min      1Q  Median      3Q     Max 
-1.5205 -0.3504 -0.1194  0.1855  4.6287 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -598.32453   22.46859 -26.629  < 2e-16 ***
latitude       5.45751    0.22945  23.785  < 2e-16 ***
longitude     -5.07910    0.24902 -20.396  < 2e-16 ***
room_type2     0.97734    0.01238  78.951  < 2e-16 ***
room_type3    -0.15381    0.04391  -3.503 0.000462 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6595 on 11882 degrees of freedom
Multiple R-squared:  0.4013,	Adjusted R-squared:  0.4011 
F-statistic:  1991 on 4 and 11882 DF,  p-value: < 2.2e-16

MSE:  0.6730552
```

Figure 5: Linear Regression output filtering by Manhattan

```
=== Linear Regression selecting the Neighboorhood group and room_type ===
Neighboorhood group = Manhattan and room_type = Entire home/apt

Call:
lm(formula = price ~ latitude + longitude, data = train_filtered[-1])

Residuals:
    Min      1Q  Median      3Q     Max 
-0.7397 -0.2368 -0.0868  0.1206  4.5392 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -387.2888    19.9701  -19.39   <2e-16 ***
latitude       3.0005     0.2045   14.68   <2e-16 ***
longitude     -3.5771     0.2186  -16.36   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4114 on 6065 degrees of freedom
Multiple R-squared:  0.05984,	Adjusted R-squared:  0.05953 
F-statistic:   193 on 2 and 6065 DF,  p-value: < 2.2e-16

MSE:  1.230804
```

Figure 6: Linear Regression output filtering by Manhattan and Entire home/Apartment

```
=== Linear Regression without filters ===

Call:
lm(formula = price ~ latitude + longitude + room_type + neighbourhood_group,
    data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2092 -0.4469 -0.1364  0.2398  4.5600

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          -2.019e+02  1.405e+01 -14.370  < 2e-16 ***
latitude             -1.634e+00  1.383e-01 -11.814  < 2e-16 ***
longitude            -3.620e+00  1.578e-01 -22.935  < 2e-16 ***
room_type2            1.012e+00  9.515e-03 106.379  < 2e-16 ***
room_type3           -2.637e-01  3.034e-02  -8.692  < 2e-16 ***
neighbourhood_group2  5.067e-01  1.601e-02  31.644  < 2e-16 ***
neighbourhood_group3  2.623e-01  1.964e-02  13.354  < 2e-16 ***
neighbourhood_group4 -9.063e-01  5.844e-02 -15.508  < 2e-16 ***
neighbourhood_group5  3.093e-01  3.853e-02   8.029 1.02e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7732 on 28566 degrees of freedom
Multiple R-squared:  0.4033,	Adjusted R-squared:  0.4031
F-statistic:  2413 on 8 and 28566 DF,  p-value: < 2.2e-16

MSE:  0.5932014
```

Figure 7: Linear Regression output without filters

Linear Regression	MSE	R^2
Entire dataset	0.59	.40
Filter: Mahnattan	0.76	0.355
Filter: Manhattan & Apt	0.44	0.12

3.2.2 Decison Trees Results

For the Decision tree the model give different results based on the variables used.

Decison tree without filters

Decison tree selecting the Neighboorhood group

Decison tree selecting the Neighboorhood group and room type

Decision Tree	MSE	R^2
Entire dataset	0.59	..
Filter: Mahnattan	0.77	..
Filter: Manhattan & Apt	0.40	..

3.2.3 Random Forest Results

There were no problem running Random Forest regression for the price, givin the default parameters. For the parameter tuning there were no possibilities for the entire dataset, due to the large number of data points. For the filtered dataset, instead, were possible to tune the mtry, number of maximum nodes and number of trees.

Random Forest without filters

There were no possibility to run a forecast of the price

Random Forest selecting the Neighboorhood group

Random Forest selecting the Neighboorhood group and room type Using the tuning of the parameter the results are slightly better. The model start with a 23% explained variance to a value of 25%.

Random Forest	MSE	$\%Var$
Entire dataset	0.56	43.9
Filter: Mahnattan	0.70	38.9
Filter: Manhattan & Apt	0.38	21.6

3.2.4 Ranger Random Forest Results

Ranger Random Forest is known to be computationally light with respect to the classic Random Forest. In fact, for the tuning part there were not problem in running it for the entire dataset.

Ranger without filters

Ranger outputs for the entire dataset are consistent. We have a R^2 of 0.47 and OOB error of .

Ranger selecting the Neighboorhood group

The dataset filtered by Neighboorhood ouputs a value of 0.4 R^2 and a OOB error of.

Ranger selecting the Neighboorhood group and room type The dataset filtered by neighboorhood and room type gives as result a R^2 of 0.25.

Ranger RF	MSE	%*Var*
Entire dataset	0.53	46.6
Filter: Mahnattan	0.71	38.8
Filter: Manhattan & Apt	0.38	21.7

3.2.5 Neural Network Results

Neural Networks	MSE	Loss
Entire dataset	0.61	0.53
Filter: Mahnattan	0.84	0.64
Filter: Manhattan & Apt	0.48	0.44

3.2.6 K-means Results

3.2.7 Principal Component Analysis Results

3.3 Discussion

3.3.1 Linear Regression Discussion

Linear regression model gives interesting results for the non-filtered dataset and also for the filtered by neighbourhood group. All variable results to rejected by Null hyphotesis, so the model depends on all the selected variables. Latitude and longitude are correlated with the target, room type is strongly positive correlated with the price and neighbourhood group does not seem to have a great contribution in the prediction of the price.

3.3.2 Decision Tree Discussion

The performance with respect to the other models are not the best, but acceptable. The predicton results are not also very precised for the filtered neighboorhood and room type. Also, the plots of the predicted value are not so consistent since the values are divided in category which correspond to the leaves that are not so strong with respect to the other model predictions.

3.3.3 Random Forest Discussion

Random forest outputs consistent results and performs better than linear regression and decision tree. Parameters tuning does not give big improvement in performance and also are computationally expensive.

3.3.4 Ranger Random Forest Discussion

Results of Ranger are the best with respect to the previous models. Also the tuning part was fast and computationally cheaper than the classic Random Forest model but does not give great improvements in performance.

3.3.5 Neural Network Discussion

3.3.6 K-Means Discussion

3.3.7 Principal Component Analysis Discussion

4 Conclusion

From the result the method with the most higher accuracy is the Random Forest method... while the worst are ....
Moreover, Random Forest method is also the worst in term of computation time for the tuning part since it takes for a configuration with 4 core, more or less 1 hour to tune the parameters.

4.1 Linear Regression

4.2 Decision Tree

Decision tree are one of the most used model in the Machine Learning world since are very familiar to human users and can be easily plotted.

4.3 Random Forest

Random Forest is an ensemble method which use a combination of decision tree to get the prediction.

4.3.1 Ranger Random Forest

Ranger Random Forest is a computationally light model which results are very close the classical Random Forest.

5

Appendix

5.1 Footnote

You can create a footnote like this.[1]

[1] I created a footnote.

References

[1] Giusti, Santochi, *Tecnologia Meccanica e Studi di Fabbricazione*. Casa Editrice Ambrosiana, Seconda Edizione

[2] Mechteacher, *Knuckle Joint – Introduction, Parts and Applications*, http://mechteacher.com/knuckle-joint/

[3] Totalmateria, *G32NiCrMo8*, http://www.totalmateria.com

[4] Sandvik Coromant,*Catalogo generale 2018*, http://www.coromant.sandvik.com/it

[5] Norme UNI, Ente nazionale italiano di unificazione