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

Supervised Learning Report

Author:
Andrea Ierardi

Academic year 2019-2020

Contents

1

Abstract

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

- Linear Regression
- Decision Tree
- Random Forest
- Ranger Random Forest
- Neural Newtworks (?)

The second part is focused on the clusterisation of the data using Unsupervised Learning technics: Cluster Analysis and K-means algorithm.

- Cluster Analysis
- K-means Algorithm

A crucial part of the project was to the tuning and finding of the hyperparameters of the different models in order to get the best fit.

2

Problem Definition and Algorithm

2.1 Two main Goal

Develope predicting models for price

The project is focused on a AirBnB user point of view. It is possible to image an application in which the user has a lot of information depending on the objective. For a landlord point of view, giving the information about the longitude and the latitude of the house, he/she will have as output the estimated price given based also on the neighbourhood group in New York City. For a guest point of view, he/she will give in input information about the price range and the type of room to get as output the most suitable houses for him or predict the position of one of that.

Define clusters and groups

For the Unsupervised part, the goal is to define cluster of price in which is possible to split out the houses, firstly in the entire city, secondly filtering for Neighborhood or room type.

FOR A SPECIFIC NEIGHBOORHOOD OR FOR THE ENTIRE NY CITY

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.

3

Experimental Evaluation

3.1 Methodology

3.1.1 Data Inspection

The dataset used in this project is one of a Kaggle competiotion and is called the New York City Airbnb Open Data. It contains 48.000 data points for each different column. The dataset has different 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

We select just 5 of this feature from the dataset since we denotes them as the most important for the price of a house: latitude, longitude, room type, neighbourhood and the price itself to compare the prediction during the tests.

From the Figure 1 is possible to see that there are some outliers in the dataset that could be removed, but only by a choise of the user, since it is possible that someone wants to rent a luxury house. In fact, no outlier was removed from the dataset. Also, there were no null and missing value apart from the reviews_per_month column that we do not take in account in the project.

From Figure2, it is possible to see the distribution of all houses in New York City and the price. Moreover, it can be noticed that the prices for the most part are in the range 0-500$ and only some istances have a price greater than 1000$. Also, the fact that some houses have a cost of 0$ is strange, nobody rent a house for free.

```
     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

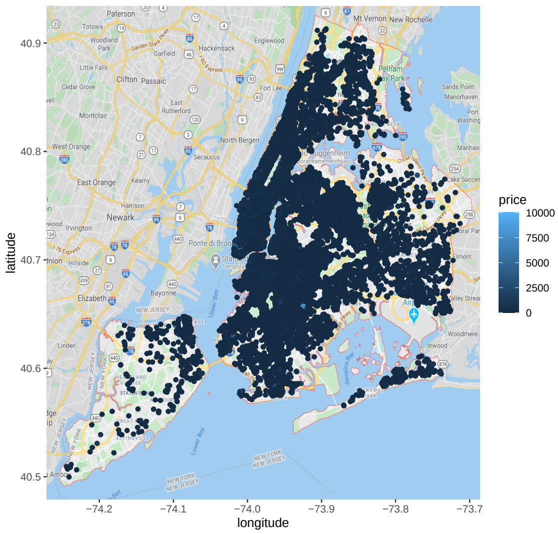

Figure 2: Distribution of all houses in NY colored by prices

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 assume 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.

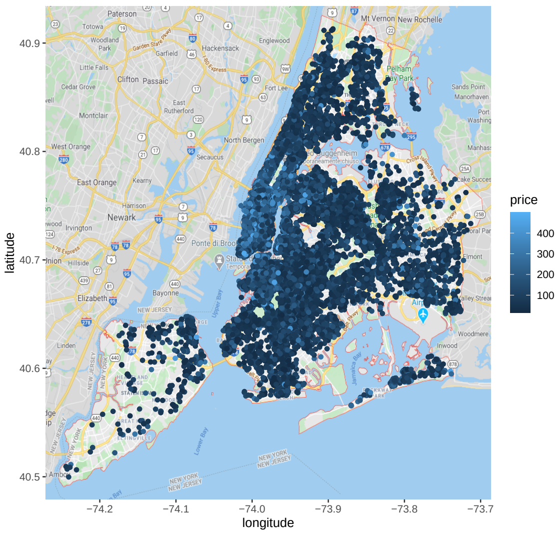

Figure 3: Distribution of all houses in NY of price between 15$ and 500$ per day

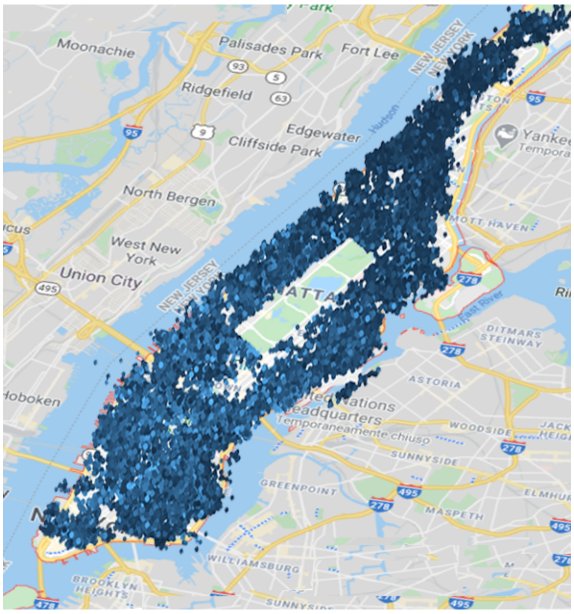

Figure 4: Approximated distribution map of all houses in Manhattan

3.2 Results

3.2.1 Linear Regression

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

3.2.2 Decison Trees

3.2.3 Random Forest

3.2.4 Ranger Random Forest

3.3 Discussion

4 Conclusion

4.1 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.

4.2 Random Forest

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

4.2.1 Ranger Random Forest

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

4.3 Linear Regression

5

Results

6

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.

7

Appendix

8

Tabelle e grafici

Here a few examples of tables and graphs.

8.1 Tabelle

Codice	*CdL*	*Lotto*	$T_{setup/lotto}$	$T_{lav/pezzo}$	$T_{proc/pezzo}$	*Quantità*	T_{tot}
100	4	250	25	0,5	0,6	1	0,6
111	2	250	20	2	2,08	1	2,08
111	3	250	15	1,5	1,56	1	1,56
112	2	250	20	2,5	2,58	1	2,58
112	3	250	15	2	2,06	1	2,06
113	3	500	15	1	1,03	2	2,06
120	1	50	30	2	2,6	0,1	0,26
121	1	25	30	3	4,2	0,1	0,42
121	1	25	30	2,5	3,7	0,1	0,37

8.1.1 Altra tabella

8.2 Grafici

9

Altro

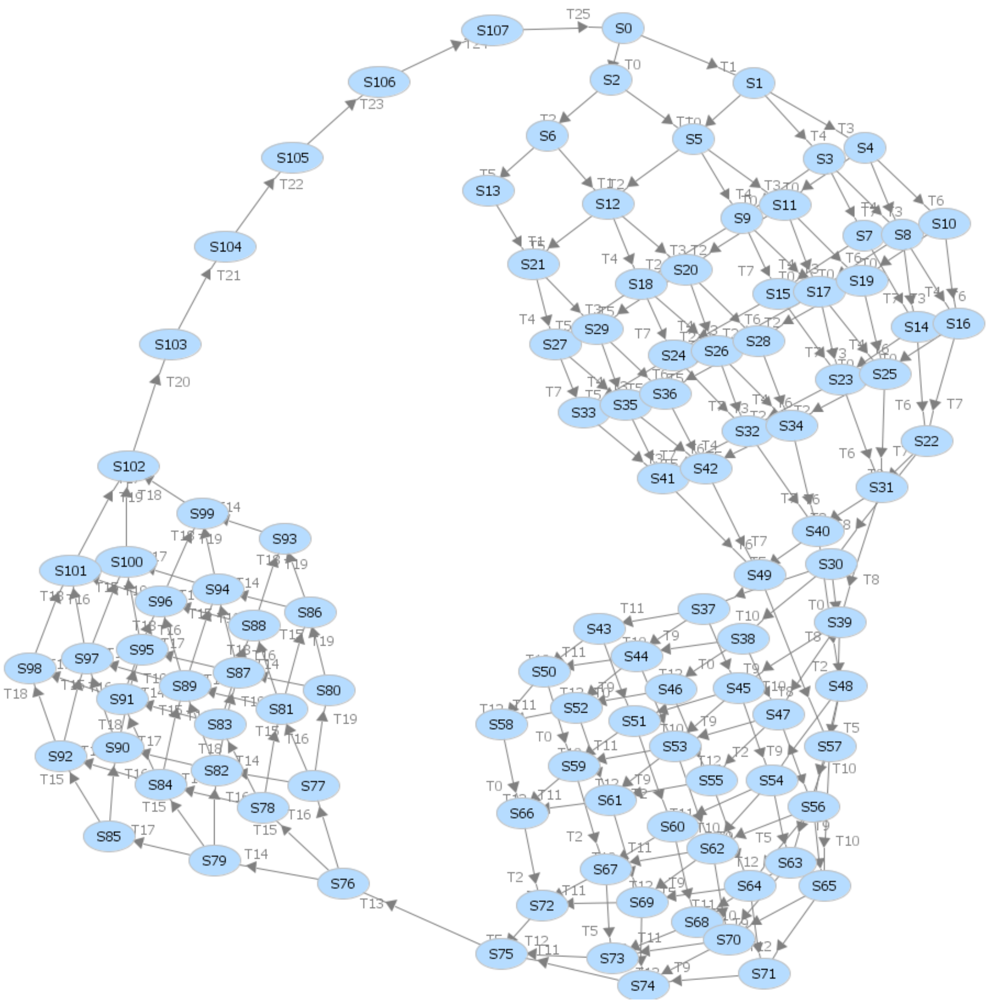

Figure 8: Didascalia.

9.1 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