Data Analysis3 Assignment-2 Jo Kudo

Introduction

In Porto, our company manages a portfolio of small to medium-sized apartments that accommodate between 2 to 6 guests per unit. Currently, we are in the process of developing a pricing strategy for our upcoming apartment listings, which have yet to be introduced to the market. This report is dedicated to the construction of a predictive pricing model tailored for Airbnb accommodations located in Porto. Our approach involves a thorough evaluation and comparison of various predictive models. The key criterion for this comparison is the RMSE, or Root Mean Squared Error, which is a standard measure used to quantify the accuracy of price predictions. By examining the RMSE values, we aim to determine the most suitable model for forecasting prices that have not been previously established. The objective is to select a predictive model that not only provides precise pricing recommendations but also complements our strategic objectives for market entry. Identifying the optimal model is critical to ensuring that our new listings are competitively priced, thereby positioning our company as a strong contender in the local accommodation sector.

▶ Code

Feature Engineering

The information used for this report comes from an Airbnb website. It has 75 different details about the places people can rent. These details help us build a good model to suggest prices for these rentals. The original dataset looks like below:

▶ Code

	id	scrape_id	description	host_id	host_list
count	1.360100e+04	1.360100e+04	0.0	1.360100e+04	13601.00
mean	3.217099e+17	2.023122e+13	NaN	2.053080e+08	35.03683
std	4.049059e+17	0.000000e+00	NaN	1.803547e+08	111.77322
min	4.133900e+04	2.023122e+13	NaN	1.442280e+05	1.000000
25%	2.451509e+07	2.023122e+13	NaN	3.903119e+07	2.000000
50%	4.606103e+07	2.023122e+13	NaN	1.437019e+08	5.000000
75%	7.344801e+17	2.023122e+13	NaN	3.785917e+08	18.00000
max	1.047130e+18	2.023122e+13	NaN	5.511028e+08	2457.000

8 rows × 40 columns

The report begins by selecting data for accommodations suitable for 2 to 6 guests, in line with the guidelines provided. Following this, variables initially presented as text are converted into numerical form for analytical purposes.

▶ Code

```
<bound method Series.info of 0        5
1        5
2        5
3        2
4        4
        ..
13596    4
13597    3
13598    3
13599    6
13600    3
Name: accommodates, Length: 13601, dtype: int64>
```

One of the changed variables is the following looking.

▶ Code

```
<bound method Series.info of 0        5.0
1        5.0
2        5.0
3        2.0
4        4.0
        ...
13596    4.0
13597    3.0
13598    3.0
13599    6.0
13600    3.0
Name: accommodates, Length: 12197, dtype: float64>
```

Additionally, the dataset retains a selection of property types to investigate the hypothesis that certain categories of accommodations may significantly impact pricing. Finally, the rows which have at least one “null” value are dropped, because all columns need to have a value for making any model.

▶ Code

Model building, prediction and model selection

Three models are built and compared with their RMSE. The following models are considered:

1. Random Forest The first model is from Random Forest, which is a machine learning algorithm that builds multiple decision trees and merges them together to get a more accurate and stable prediction. It uses a technique of random sampling of training data points and features when building trees.

Following step is conducted; - splitting train and test data - conducting dmatrices - conducting random forest method for building model and calculating total RMSE on holdout data

▶ Code

```
data_train.shape= (7226, 8)
data_holdout.shape= (3098, 8)
```

▶ Code

```
X_train=
[[  1.     0.     0.    ...   2.   176.     4.59]
 [  1.     0.     0.    ...   2.     7.     4.14]
 [  1.     0.     0.    ...   1.    86.     4.83]
 ...
 [  1.     0.     0.    ...   1.    54.     4.35]
 [  1.     0.     0.    ...   1.   115.     4.73]
 [  1.     0.     0.    ...   1.    60.     4.73]]
```

▶ Code

```
Fitting 5 folds for each of 12 candidates, totalling 60
fits
[CV 1/5] END max_features=6, min_samples_leaf=5;,
score=-103.531 total time=   0.5s
[CV 2/5] END max_features=6, min_samples_leaf=5;,
score=-53.818 total time=   0.3s
[CV 3/5] END max_features=6, min_samples_leaf=5;,
score=-55.231 total time=   0.3s
[CV 4/5] END max_features=6, min_samples_leaf=5;,
score=-64.285 total time=   0.3s
[CV 5/5] END max_features=6, min_samples_leaf=5;,
score=-149.338 total time=   0.3s
[CV 1/5] END max_features=6, min_samples_leaf=10;,
score=-104.214 total time=   0.3s
[CV 2/5] END max_features=6, min_samples_leaf=10;,
score=-55.339 total time=   0.3s
[CV 3/5] END max_features=6, min_samples_leaf=10;,
score=-56.310 total time=   0.3s
[CV 4/5] END max_features=6, min_samples_leaf=10;,
score=-65.146 total time=   0.3s
[CV 5/5] END max_features=6, min_samples_leaf=10;,
score=-149.817 total time=   0.3s
[CV 1/5] END max_features=6, min_samples_leaf=15;,
score=-104.428 total time=   0.3s
[CV 2/5] END max_features=6, min_samples_leaf=15;,
score=-55.734 total time=   0.2s
[CV 3/5] END max_features=6, min_samples_leaf=15;,
score=-56.719 total time=   0.2s
[CV 4/5] END max_features=6, min_samples_leaf=15;,
score=-65.374 total time=   0.2s
[CV 5/5] END max_features=6, min_samples_leaf=15;,
score=-150.007 total time=   0.2s
[CV 1/5] END max_features=8, min_samples_leaf=5;,
score=-103.232 total time=   0.4s
[CV 2/5] END max_features=8, min_samples_leaf=5;,
score=-52.611 total time=   0.6s
[CV 3/5] END max_features=8, min_samples_leaf=5;,
score=-53.582 total time=   0.7s
[CV 4/5] END max_features=8, min_samples_leaf=5;,
score=-63.340 total time=   0.5s
[CV 5/5] END max_features=8, min_samples_leaf=5;,
score=-148.842 total time=   1.1s
[CV 1/5] END max_features=8, min_samples_leaf=10;,
score=-103.732 total time=   3.0s
[CV 2/5] END max_features=8, min_samples_leaf=10;,
score=-54.061 total time=   1.1s
[CV 3/5] END max_features=8, min_samples_leaf=10;,
score=-55.193 total time=   0.8s
[CV 4/5] END max_features=8, min_samples_leaf=10;,
score=-64.499 total time=   0.8s
[CV 5/5] END max_features=8, min_samples_leaf=10;,
score=-149.401 total time=   0.9s
[CV 1/5] END max_features=8, min_samples_leaf=15;,
score=-103.934 total time=   1.7s
[CV 2/5] END max_features=8, min_samples_leaf=15;,
score=-54.718 total time=   1.6s
[CV 3/5] END max_features=8, min_samples_leaf=15;,
score=-55.877 total time=   1.4s
[CV 4/5] END max_features=8, min_samples_leaf=15;,
score=-64.764 total time=   1.1s
[CV 5/5] END max_features=8, min_samples_leaf=15;,
score=-149.590 total time=   1.0s
[CV 1/5] END max_features=10, min_samples_leaf=5;,
score=-102.820 total time=   1.5s
[CV 2/5] END max_features=10, min_samples_leaf=5;,
score=-51.807 total time=   2.4s
[CV 3/5] END max_features=10, min_samples_leaf=5;,
score=-52.824 total time=   0.8s
[CV 4/5] END max_features=10, min_samples_leaf=5;,
score=-62.862 total time=   0.5s
[CV 5/5] END max_features=10, min_samples_leaf=5;,
score=-148.487 total time=   0.7s
[CV 1/5] END max_features=10, min_samples_leaf=10;,
score=-103.362 total time=   0.4s
[CV 2/5] END max_features=10, min_samples_leaf=10;,
score=-53.656 total time=   0.5s
[CV 3/5] END max_features=10, min_samples_leaf=10;,
score=-54.559 total time=   0.6s
[CV 4/5] END max_features=10, min_samples_leaf=10;,
score=-64.172 total time=   0.6s
[CV 5/5] END max_features=10, min_samples_leaf=10;,
score=-149.085 total time=   0.6s
[CV 1/5] END max_features=10, min_samples_leaf=15;,
score=-103.704 total time=   0.8s
[CV 2/5] END max_features=10, min_samples_leaf=15;,
score=-54.258 total time=   0.7s
[CV 3/5] END max_features=10, min_samples_leaf=15;,
score=-55.293 total time=   0.9s
[CV 4/5] END max_features=10, min_samples_leaf=15;,
score=-64.648 total time=   0.6s
[CV 5/5] END max_features=10, min_samples_leaf=15;,
score=-149.391 total time=   1.4s
[CV 1/5] END max_features=12, min_samples_leaf=5;,
score=-102.648 total time=   1.0s
[CV 2/5] END max_features=12, min_samples_leaf=5;,
score=-51.152 total time=   0.8s
[CV 3/5] END max_features=12, min_samples_leaf=5;,
score=-52.703 total time=   1.6s
[CV 4/5] END max_features=12, min_samples_leaf=5;,
score=-62.353 total time=   0.9s
[CV 5/5] END max_features=12, min_samples_leaf=5;,
score=-148.088 total time=   1.1s
[CV 1/5] END max_features=12, min_samples_leaf=10;,
score=-103.164 total time=   0.7s
[CV 2/5] END max_features=12, min_samples_leaf=10;,
score=-52.743 total time=   1.2s
[CV 3/5] END max_features=12, min_samples_leaf=10;,
score=-53.789 total time=   0.6s
[CV 4/5] END max_features=12, min_samples_leaf=10;,
score=-63.580 total time=   0.6s
[CV 5/5] END max_features=12, min_samples_leaf=10;,
score=-148.786 total time=   0.7s
[CV 1/5] END max_features=12, min_samples_leaf=15;,
score=-103.404 total time=   0.8s
[CV 2/5] END max_features=12, min_samples_leaf=15;,
score=-53.667 total time=   0.5s
[CV 3/5] END max_features=12, min_samples_leaf=15;,
score=-54.353 total time=   0.5s
[CV 4/5] END max_features=12, min_samples_leaf=15;,
score=-64.032 total time=   0.5s
[CV 5/5] END max_features=12, min_samples_leaf=15;,
score=-149.073 total time=   0.6s
```

	rmse	mean_price	rmse_norm
Total	62.09	80.09	0.78

2. Linear Ordinary Least Squares (OLS) Regression The second model is from Linear OLS Regression. The aim is to draw a line that best fits the data by minimizing the sum of the squared differences between the actual data points and the line.

▶ Code

```
ols_rmse= 90.13535869925323
```

3. LASSO (Least Absolute Shrinkage and Selection Operator) The third model is from LASSO, which is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. The key feature of LASSO is its ability to shrink the coefficients of less important variables to exactly zero.

▶ Code

```
lasso_rmse= 83.37626925962692
```

Comparing the three model with RMSE

Here is a table, which illustrates the comparison of the three models with RMSE. To calculate RMSE, cross validation method is used. According to this table, RMSE from Random Forest see the lowest score, which means Random Forest is the best model for predicting the price of the future accommodation.

▶ Code

	model	CV RMSE
0	OLS	90.135359
1	LASSO	83.376269
2	random forest	62.090000

Conclusion

In the end, when looking at how well three different ways of guessing prices did, each one had a different score for how close the guesses were to the real prices. The score used is called RMSE, which stands for how much the guesses are off, on average. Out of all of them, the Random Forest was the best at predicting what prices will be like later on. This is because it combines a lot of simple guessers into one big guesser that makes better predictions and doesn’t mess up when things get tricky. So, for figuring out future prices well, Random Forest is the best choice.