OYO hotels rental price prediction in China

Introduction

OYO hotels, as known as OYO hotels and homes. In recent years, OYO hotels have developed rapidly in China, mainly in franchise and chain of lease, occupying many second and third-tier cities' market.
This report tries to predict the rental price of the OYO hotels in China. So, the target variable is "rental price”, and it's a regression problem. Because OYO's market layout is concentrated in second and third-tier cities, it is also the main city in China, occupying a large part of economic development. Therefore, the research on it has a great reference value.
This report downloads the dataset from Kaggle [1], but all the information can be gathered form Chinese OTA platforms. This dataset has 5834 data points and 25 features.

Features overview:

There are 7 categories features, but some of them like “host_is_superhost”,” has_availability” don’t have some meanings. So, I drop of them. And there are 18 continuous features in the dataset. What is more special is that there are two features that are latitude and longitude, and I hope to show these features through scatter in the future.

Public searching:

Because this data set is new and rare, there is only one piece of machine learning model code about this data set. I tried to understand it. The author uses this data set to predict rent price in China and chose two models. From the observation of rmse curve, I think the second model fits better, but the mse value is slightly larger than the first model. The author finally chose the second model.
The author encodes target variable with OrdinalEncoder and tunes parameters with XGBRegressor. The prediction of y (y_train, y_pred) in model 1 has mse value of 4.47 and smse value of 2.11; (y_val, y_pred)

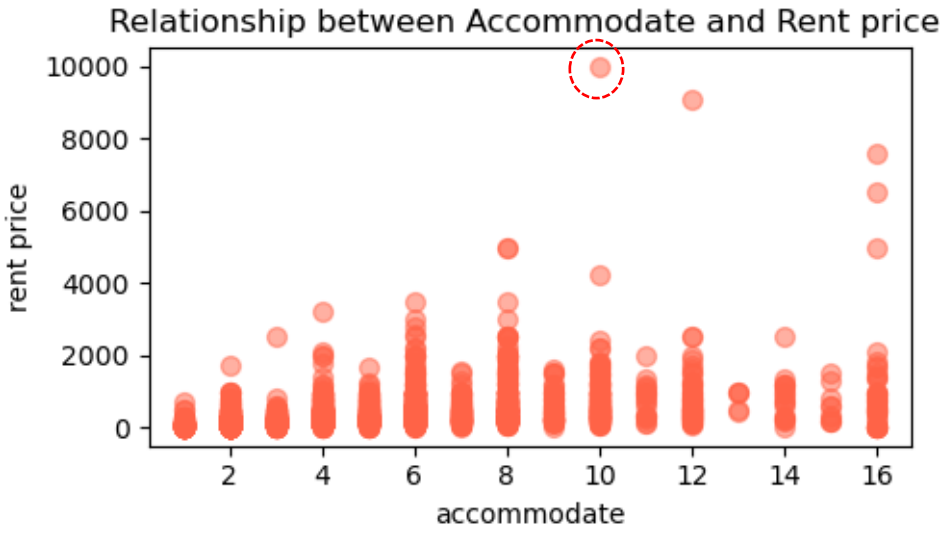
has a mse value of 45.13 and a smse value of 6.71. Model 2 has a mse value of 46.12. The author finally chose Model 1.

Exploratory Data Analysis

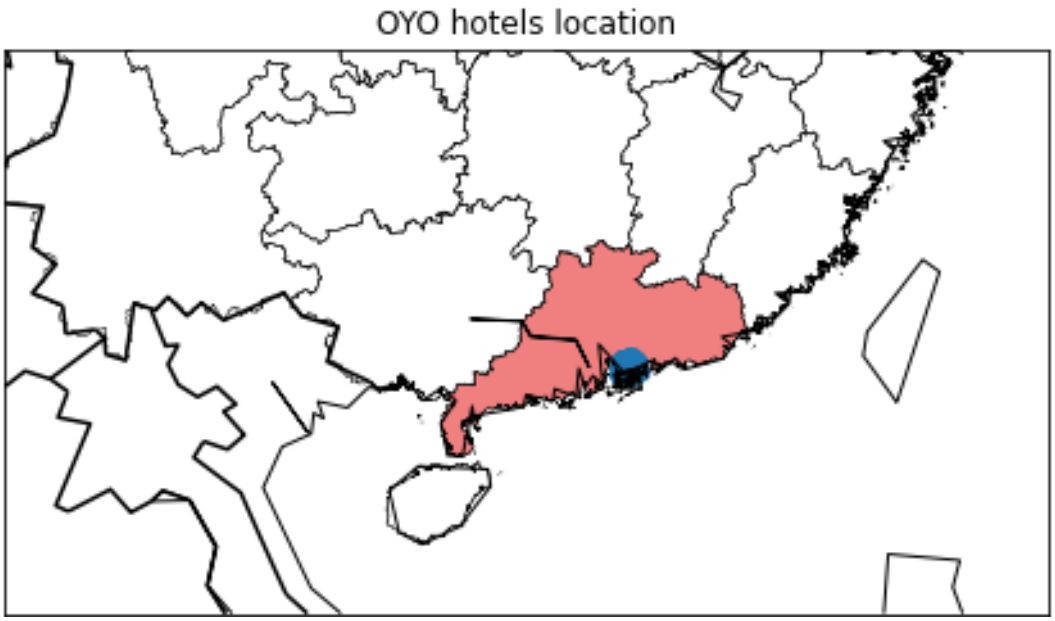
1. **The target variable "rental price" is a regression problem**, so I use "describe ()" function to exploratory the data. And the max of the price is 10000, the min of it is 0, and the average of it is 286.

Because the value difference is too large, so I enlarge the axis by log function like the figure 1 shows. It is clearly to see that the distribution is right-skewed.

2. **Relationship between Accommodate and Rent price**.
   (Accommodate which means somewhere provide lodging or sufficient space for.)
   V This is a numerical classification feature, which should have been plotted with bar plot. But there are some special points that I hope



   to express through scatter.
   As the figure2 show, if the mark point is equal to 10, the price of this point is particularly high. Because this sample point is not an ordinary hotel of OYO, but a private villa in cooperation with OYO. That' s the reason why this point' s price is so high.
3. **The OYO hotels locations in China.**



Because the dataset has two features, longitude and latitude, I choose to use basemap to draw the country map and scatter to represent the hotel distribution location.
In figure4, the blue dots are gathered by more than 5000 data points, but due to the close location of hotels, the dots seem to be dense. It can be seen that this data set collects the information of OYO hotels in Guangdong Province.

Methods

Split:
Since this data set has more than 5,000 sample points, it is not a large data set, and it’ s an i.i.d. dataset, so I apply “basic split method” to i.i.d. datasets. Meanwhile, I want to use GridSearchCV function to integrate the model and the preprocessing, and use KFold function to split the remaining dataset. So, I used 20% of the data as a test set, and the remaining 80% was trained with KFold folding four times
Preprocessing:
Firstly, I dropped worthless features, like “amenities”, “has_availability”. Secondly, for classification variables such as' bed_type ',' host_is_superhost 'is encoded with OneHot and Ordinal. For contiguous variables, such as' commodites', 'availability_30' is encoded with MinMax and Standard.

ML pipeline:

I used four models to train the data. Through GridSearchCV to integrate all the machine learning process. Due to the target variable is regression, I chose RMSE to be the metric. And RMSE is a metric for measuring error, so the lower the metric' s value, the better the model performance. I define a function that combines data split, preprocessing, and tuning model parameters, and finally outputs the best parameters and test scores.

1. L1 linear regression

Import Lasso model, define the random state equal to 42, and tune the parameter alpha traversing the list: [0.25,2.5,25]. Finally, we have the average of the test score 310.9.

2. Random Forest Regressor

Fix the random state, and tune the parameters: max_features[3,5,7,9],max_depth[3,5,7,9]. We got the test score mean: 292.01.

3. SVR (Support Vector)

I choose 'rbf' as the kernel value and tune the parameters like gamma [0.1, 10, 100], C [0.1, 1, 10]. The average of the test scores is 421.57.

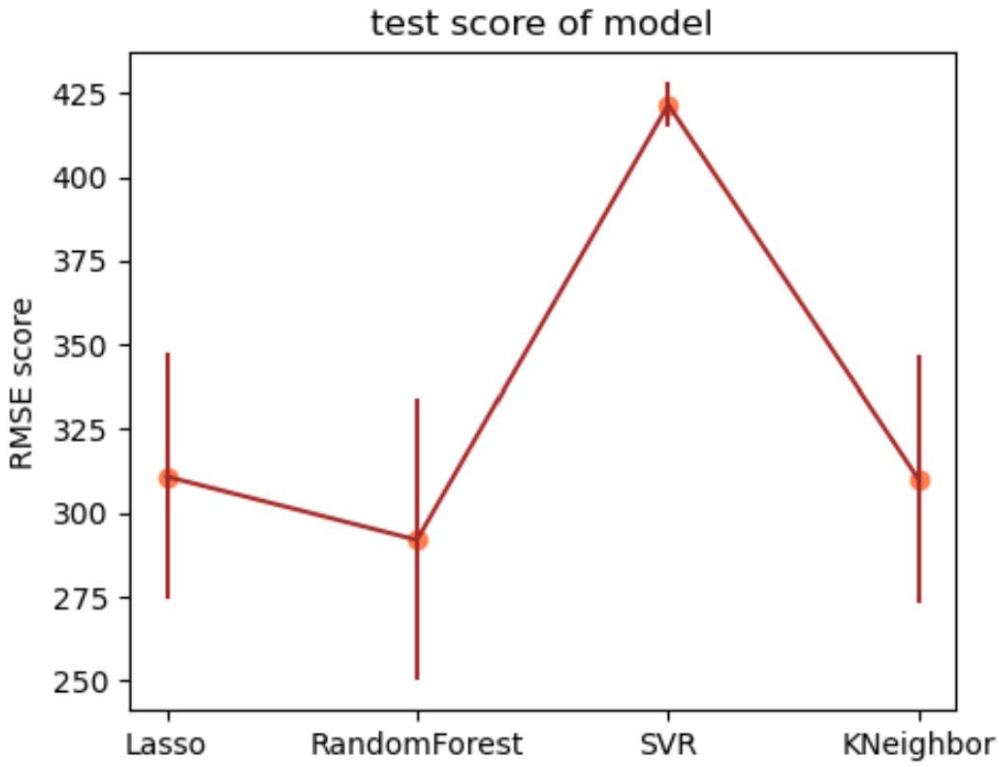
4. KNeighbors Regressor

Choose the 'uniform' as the weights value and tune the parameter n_neighbors [5,25,50]. The test score mean is 310.08.

The range of values of model tune parameters and the mean and variance values are shown in the following table:

Model	mean	std	parameters	values
Lasso	310.909	47.198	alpha	0.25,2.5,25
RandomForest	292.018	48.858	max_features	[3,5,7,9]
			max_depth	[3,5,7,9]
SVR	421.576	62.815	gamma	[0.1, 10, 100]
			C	[0.1,1,10]
Kneighbors	310.082	49.206	n_neighbors	[5,25,50]

Based on the test scores for each model, we got mean and standard deviation. So, we got the RMSE score for each model shown in the following figure.



Results

Baseline model:

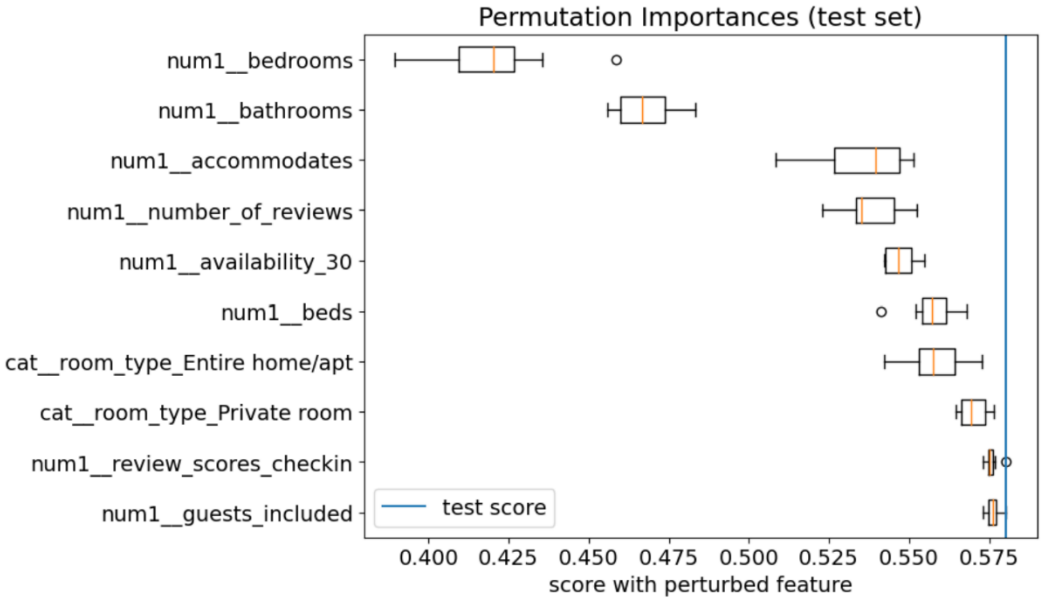
Make y_mean equal to y_pred, then we have the baseline RMSE: 445.78, and we compare all models' test score , the four models L1, Random forest, SVR, KNN is 1.8, 2.09, 0.22, 1.8 standard deviations lower than the baseline model.

Best model:

It' s obviously, the lowest RMSE is the best model. At the same time, the best parameters of random forest model are max_depth=9, max_features = 5. According to the random forest model, the predicted value and the real value are made into scatter plot and visualized, as shown in the following figure. The pink dot is the true value, and the green fork is the predicted value. It can be seen that the fitting is good.

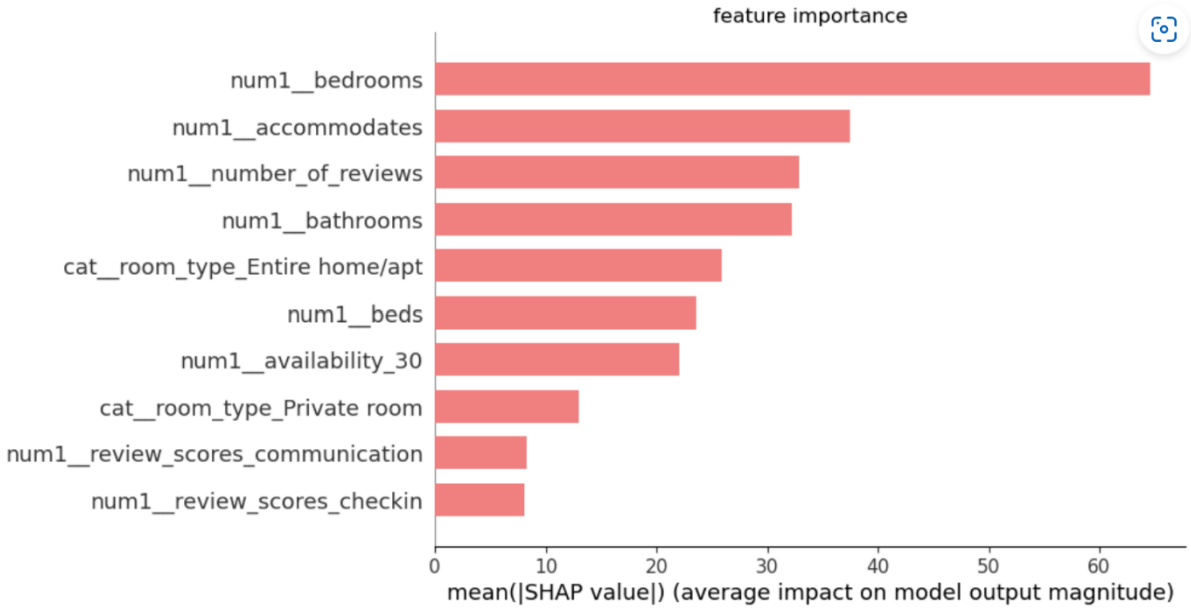
Global feature:

1. Use the permutation function to shuffle each feature, and visualized the TOP 10 importance features. The lower the score, the worse the model becomes after the feature is shuffled, and this feature has a greater influence on the prediction.

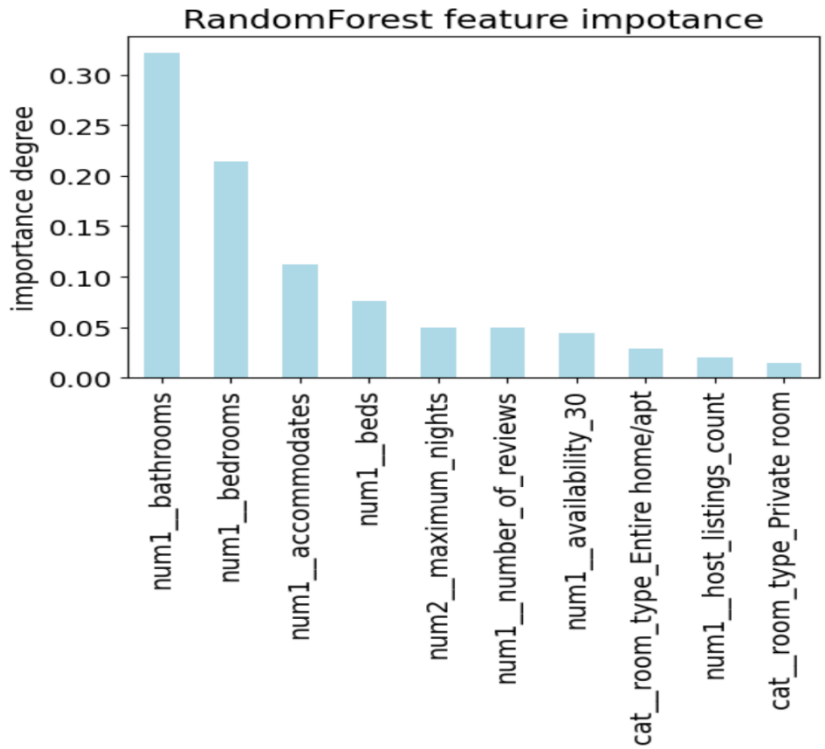


2. Use SHAP class to sort the features. Because the best model is a

random forest, use the shape. TreeExplainer to calculate the important features.

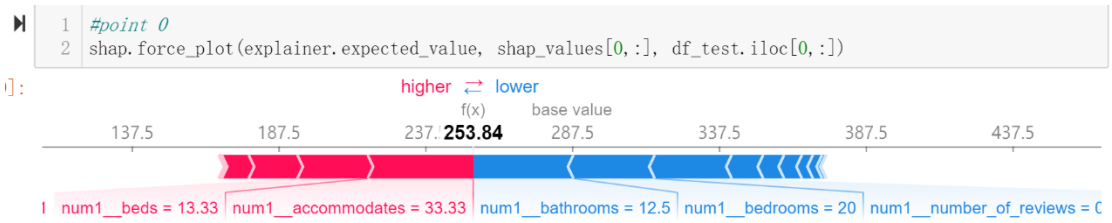


3. Using the feature _ impotances function built in the Random Forest model, the bar plot is drawn according to the result.

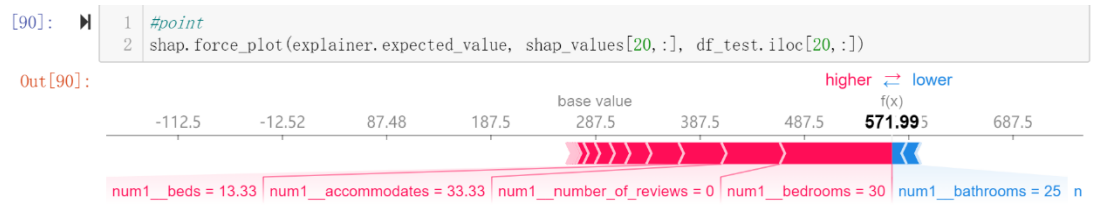


Local feature:

Choose 0 and 20 in sample points as examples to explain the importance of features.



this point is lower than the base value, the bathrooms, bedrooms and so on, contribute a large negative influence on this point. there are only few features positively distribute for this point's value like accommodates and beds.



this point's values are higher than base value, bedrooms, number of reviews and accommodates, most of which bring positive influence to change points, while bathrooms bring main negative influence to change points.

Unexpected:

Most feature importance judgments include bathroom, which is a relatively unexpected influence feature and a hotel price prediction that is easily overlooked.

Outlook

My final model is random forest, and I will want to try XGBoost to improve the model, because this model provides some advantages over other models, such as minimum error path, faster convergence of steps, and simplified calculation to improve speed.

I also want to try LIME[3] as an alternative package to SHAP to interpret the model. It will choose to train a new interpretable model like a decision tree, and its visualization will contain the prediction probabilities of each feature. I think this is easier to understand.

Meanwhile, I would like to collect more data about comment features and a wider range of OYO hotels, because comment features also have an important influence on model prediction, but there are too many missing data in this data set, and the data collection is almost concentrated in coastal cities.

References

[1] https://www.kaggle.com/code/budibudi/oyo-rental-price-prediction-in-china-003/data
[2] https://www.kaggle.com/code/budibudi/oyo-rental-price-prediction-in-china-003/notebook#Model
[3] https://towardsdatascience.com/three-interpretability-methods-to-consider-when-developing-your-machine-learning-model-5bf368b47fac

GitHub repository

My GitHub:
https://github.com/AstrosiosaurQ7/data1030_final_project.git