Prediction of Airbnb short-term rental housing value in Beijing

Member 1	Wenjun Han	hwenjun@vt.edu
Member 2	Geping Chen	gepingc@vt.edu
Member 3	Weiting Li	weitil6@vt.edu

Introduction

Airbnb allows guests to rate their stay through review rating. Housing experience value is one of the important indexes to determine if the guests feel the housing services provided good value for the price. An accurate prediction model for the housing experience value is an efficient tool for customers to compare and choose short-term rental. However, Airbnb short-term rental housing experience can be influenced by many other factors such as host attitude, distance to metro, cleaning fee, etc. Providing an Airbnb rental housing rating prediction model is challenging since many factors are involved in the modeling process. Therefore, a data-driven model is an appropriate solution to handle the complicated datasets.

Project Problem Statement

This project aims to apply our knowledge in data science to investigate the features of Airbnb data and establish a prediction model to forecast the Airbnb short-term rental housing experience value in Beijing, China. We want to develop a supervised learning model to predict the value rating for short-term rental housing. The finalized model is expected to accurately predict the Airbnb housing value ratings. It is expected to be an efficient tool for the users to estimate the true financial value of the housing experiences and provide references for their decision-making.

Data Set Description

The collected dataset for the project is called Beijing Airbnb Data. The dataset is available on Kaggle.

The raw data consists of 28452 rows. The raw data we have consists of 106 columns including 1 column called review_scores_value, which is the target of our project. Except for the id, name, listing_url section, there are about 68 features that can be used to analyze. The dataset includes numerical data, categorical data, date data, text information. Numerical data like review_scores_checkin, square feet, records the value of the house. Categorical data like property_type, room type. Date data including the host_since, first_review etc. Text information includes some description of the house, notes written by the host etc.

The data still needs to be cleaned and expanded. There is a lot of NaN data; meaningless columns like id, host_name etc; columns with a lot of text information that needs to be processed into numerical data. Besides these There are some columns that can be transferred into new features. The amenities column records the various facilities owned by the host. We believe that these facilities have played a key role in the value of the house posted on airbnb. Therefore we transferred such columns into several features. After processing we have a total of 233 columns including 232 features and a target.

Flowchart

The data analysis process consists of data augmentation, data pre-processing, modeling and model evaluation. The data augmentation algorithm randomly chooses data samples from existing datasets without replacement to expand the datasets.

After data augmentation, data pre-processing includes 3 main parts: data cleaning, feature engineering and PCA. In the data cleaning part, meaningless columns are dropped, and meaningful text columns are encoded 0 or 1 based on whether the columns contain information. During feature engineering, since the column of amenities contains lots of meaningful data, the information of amenities is expanded into new column features. Through the feature engineering, the column features are expanded to 232 in total. After that, PCA is applied to reduce the data dimension and 10 PCA components are kept for further analysis.

In the modeling phrase, 5 machine learning models are applied for prediction. They are decision tree, naive bayes, KNN, random forest and SVM models. To compare the model results, average precision, average F-1 score and average recall scores are calculated based on the 5-fold cross validation results. The models would be evaluated using the criterion.

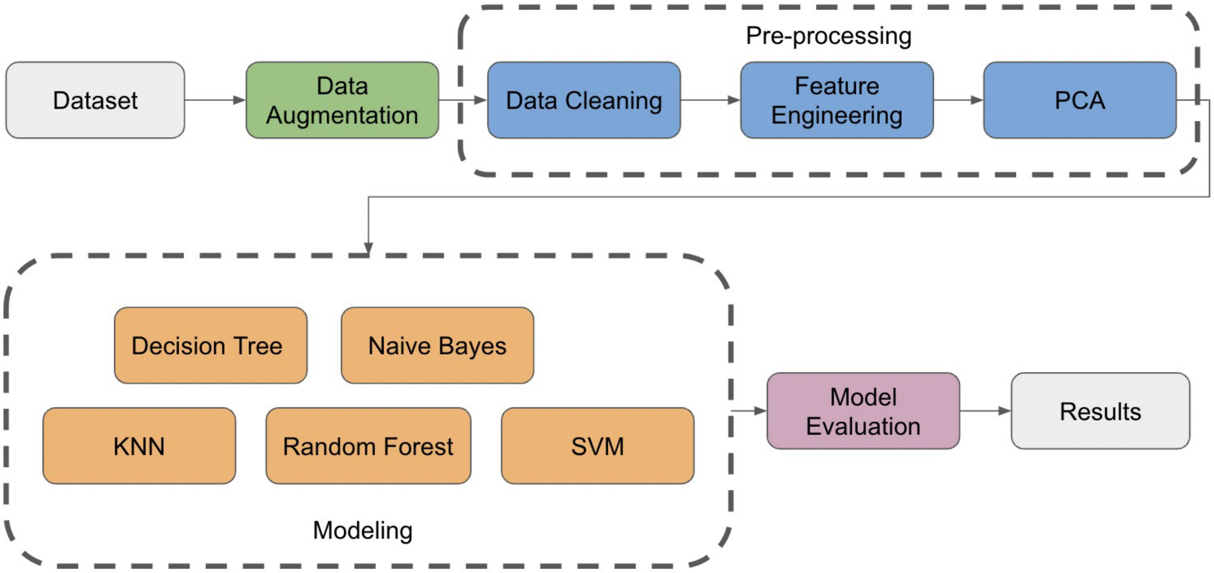


Figure 1. Flowchart of Airbnb data analysis

Data Augmentation

The raw dataset contains 28452 records which are not meeting the project requirements. Data augmentation method is applied to expand the existing datasets. For each column, records from the raw dataset are randomly selected with replacement and appended to the existing dataset. Through the method, our dataset records reach 227616 records and efficiently enlarge our data.

Pre-Processing Steps

Data Cleaning and Feature Engineering

For the meaningless columns like id, house name, scrape id, listing url, and for some columns like experiences_offered that have the same data for all rows, we simply drop these columns.

For long text columns that we think are meaningful and will definitely affect the result like description and transit tips provided by the host, we transfer them into 1 and 0, 1 means this row has data while 0 means NaN. For example, whether the house has a description matters a lot for customers to choose the house.

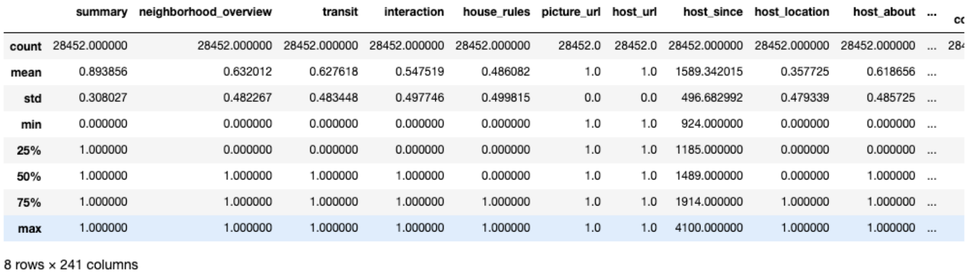
For date data like host_since, we calculate the time difference and transfer it from date data into numerical data.

For categorical data like host_location, where host currently is meaningful for customers to choose and score the house, we find all rows that contain Beijing in the string, and transfer it into type 1, and transfer other data including NaN into 0. And for categorical data like neighbourhood_cleansed, some categories have a very small amount of data, we combine these categories into the same type.

For some numerical data such as review_score_cleaning, count private rooms, we replace the NaN data by the mean of this column. At last, we transfer all boolean values into 1 and 0. And make sure all the data in the dataset are numerical.

In order to get more features, we process some of the text columns into new features. For the column amenities, it records the various facilities owned by the host like whether the house has TV, wifi or even a BBQ grill. These are all important features for customers to choose the house and therefore we parse all these strings into new features. We used the set to store all the amenities provided in the dataset and these are all the columns that we are going to append in our dataset. And used a list to store all the amenities in each row. And for each row, if the amenities are in the set, then we set the corresponding column into 1, while 0 for not in the set. After some process, we have 232 features in the end.

After doing all this, we can now use the describe function provided by the pandas package and get a statistical view of our data.



	summary	neighborhood_overview	transit	interaction	house_rules	picture_url	host_url	host_since	host_location	host_about	...	co
count	28452.000000	28452.000000	28452.000000	28452.000000	28452.000000	28452.0	28452.0	28452.000000	28452.000000	28452.000000	...	284
mean	0.893856	0.632012	0.627618	0.547519	0.486082	1.0	1.0	1589.342015	0.357725	0.618656	...	
std	0.308027	0.482267	0.483448	0.497746	0.499815	0.0	0.0	496.682992	0.479339	0.485725	...	
min	0.000000	0.000000	0.000000	0.000000	0.000000	1.0	1.0	924.000000	0.000000	0.000000	...	
25%	1.000000	0.000000	0.000000	0.000000	0.000000	1.0	1.0	1185.000000	0.000000	0.000000	...	
50%	1.000000	1.000000	1.000000	1.000000	0.000000	1.0	1.0	1489.000000	0.000000	1.000000	...	
75%	1.000000	1.000000	1.000000	1.000000	1.000000	1.0	1.0	1914.000000	1.000000	1.000000	...	
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.0	1.0	4100.000000	1.000000	1.000000	...	

8 rows × 241 columns

Figure 2. Data description of raw dataset

Correlation and PCA

Before further processing, we plot the frequency plot of the housing value as in Figure 1. We found that the value ratings are concentrated on 9 to 10. Based on the distribution, we classify the values into 4 groups. Group 0 contains the ratings below or equal to 8. Group1 represents values between 8 to 9.5. Group 2 stands for values between 9.5 to 9.8, while Group 3 represents those values above 9.8. The frequency distribution of the 4 groups of values are shown in Figure 4.

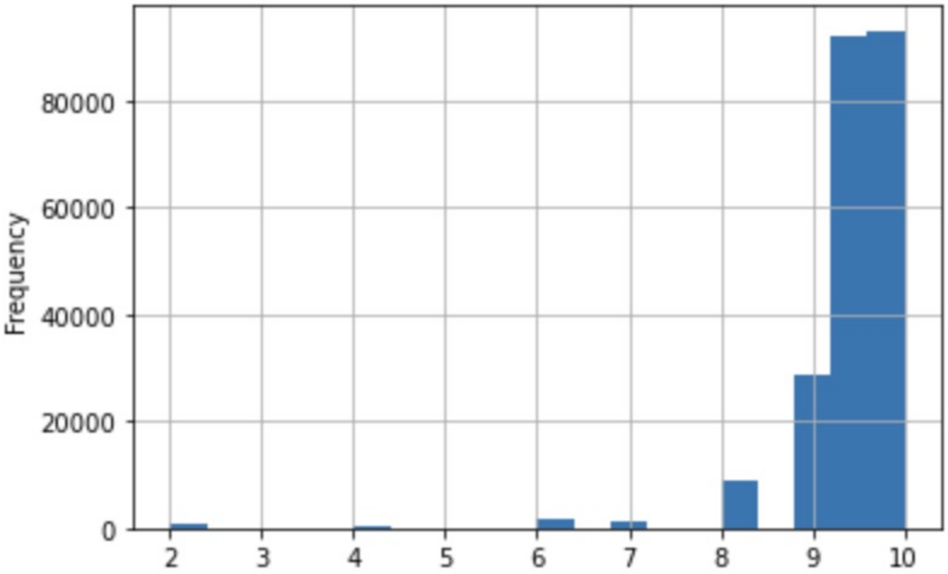


Figure 3. Frequency distribution plot of housing experience value of Airbnb

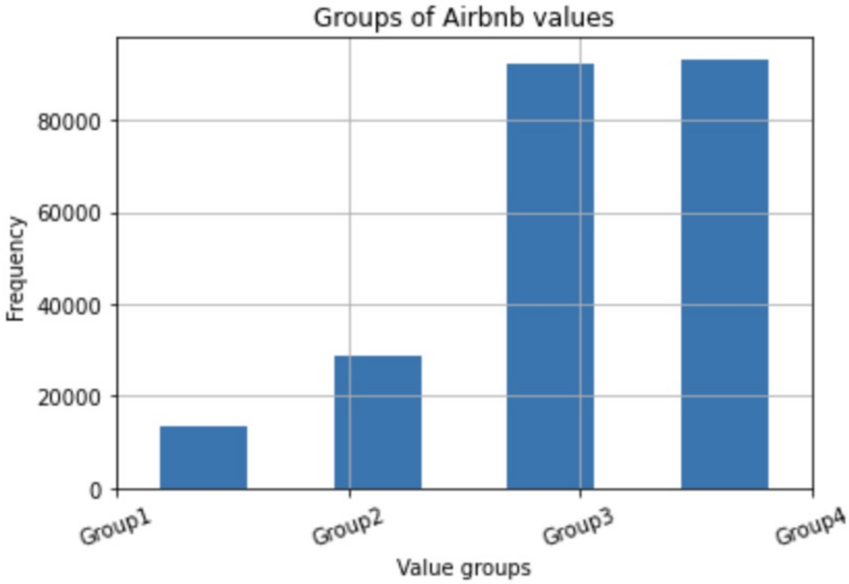


Figure 4. Frequency distribution plot of housing value in groups

Pearson correlation coefficients are calculated and a correlation matrix was constructed. We found that the housing rating variable is not highly influenced by any other single variable. L2 normalization is applied to fit the data into a common scale. PCA is applied for dimension reduction. 10 PCA components are kept for further analysis. The explained variance ratio of the first two components are 0.996 and 0.002. We find that the first 1 component explains nearly most of the variance. The relationship of the number of components and the variance ratio is plotted as in Figure 3.

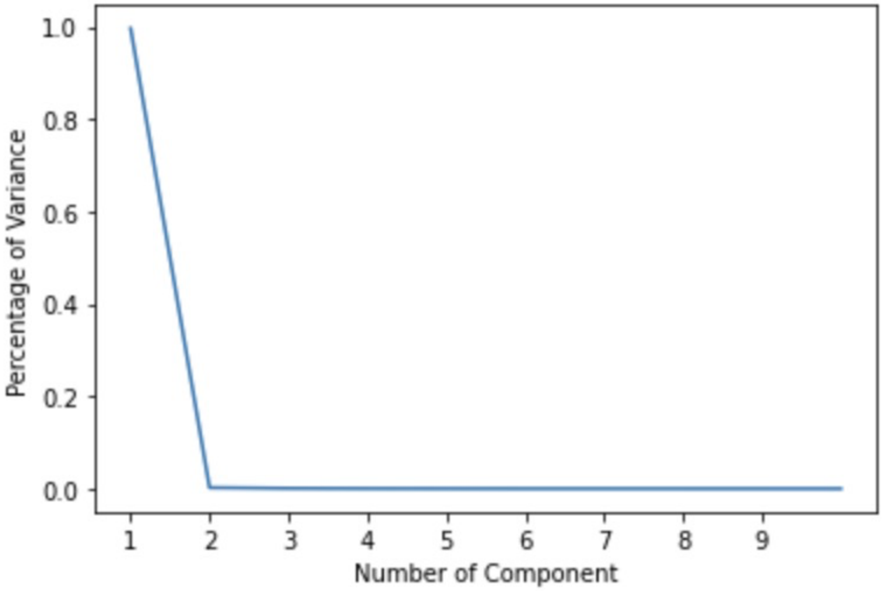


Figure 3. PCA components and explained variance ratio

Modeling

Since we are predicting the value groups of Airbnb housing, we applied classification methods for machine learning modeling. The following models are used in this project:

Decision Tree

The Decision Tree model is applied for the classification with 5 fold cross validation. Entropy is used for the tree development as criteria.

Naive Bayes

Naive Bayes classifier is applied with 5 fold cross validation. Naive Bayes is chosen since some variables are correlated and prior conditions may exist. The Gaussian Naive Bayes model is used since we have more than 2 groups of values.

KNN classification

In our project, we also used Knn classification to further study our subject. The k-nearest neighbors algorithm(Knn) is a simple, supervised machine learning algorithm that is used to solve classification problems. The reason we enact the Knn algorithm is because it makes highly accurate predictions. After we tweak the k value. We found that when k reaches 1 which means 1 neighbor, the results of the algorithm start to converge, and thus we get our most accurate prediction. When k equals to 1, the mean squared residual becomes 0.69.

Random Forest

We use the Random Forest classifier provided in the sklearn package to train and evaluate the dataset. The depth of the tree is 50. I've tested from 10 to 50, and 50 is good. And we use the 5-fold to split the dataset into train and test data. Evaluation method is the confusion matrix.

SVM

SVM is used to classify the data. And we choose the SVC model. 5-fold is used to cross validate our dataset. Confusion matrix is used to evaluate the model.

Evaluation

We evaluated the classification models with calculating the F1-score, precision and recall as criteria. Confusion matrices of the classification models were established for the comparison as well. Following are the evaluation results of each of the models.

Decision Tree

For the 5 fold cross validation, we obtained the classification report as the following table.

Table 1. Classification evaluation criterion for the Decision Tree classifier

	Precision	Recall	F1-score
Average	0.715	0.588	0.554

From the table above, the precision, recall and F1-score of the decision tree model is 0.715, 0.588 and 0.554 respectively The confusion matrices of the 5 fold cross validation results are shown as in Figure 4. We can find that the model easily confuses Group 3 with Group 2.

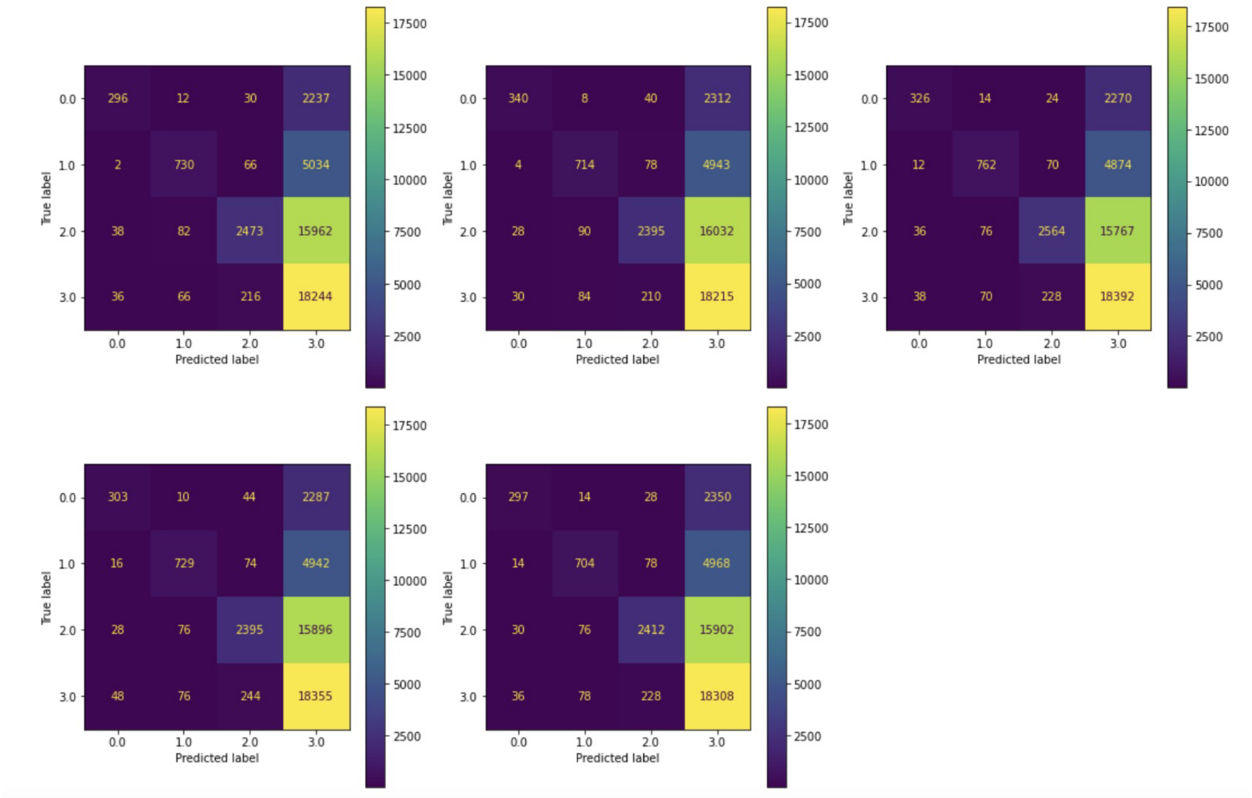


Figure 4. Confusion matrices of the Decision Tree model

Naive Bayes

For the 5 fold cross validation, we obtained the classification report as the following table.

Table 2. Classification evaluation criterion for the Naive Bayes classifier

	Precision	Recall	F1-score
Average	0.354	0.352	0.219

From the table above, the precision, recall and F1-score of the decision tree model is lower than the decision tree. The confusion matrices of the 5 fold cross validation results are shown as in

Figure 5. The matrices are similar to the Decision Tree results. It could hardly distinguish Group 2 and Group 3 while the true classification is Group 2. However, when the true classification is Group 3, the prediction results of Group 3 can be clearly identified.

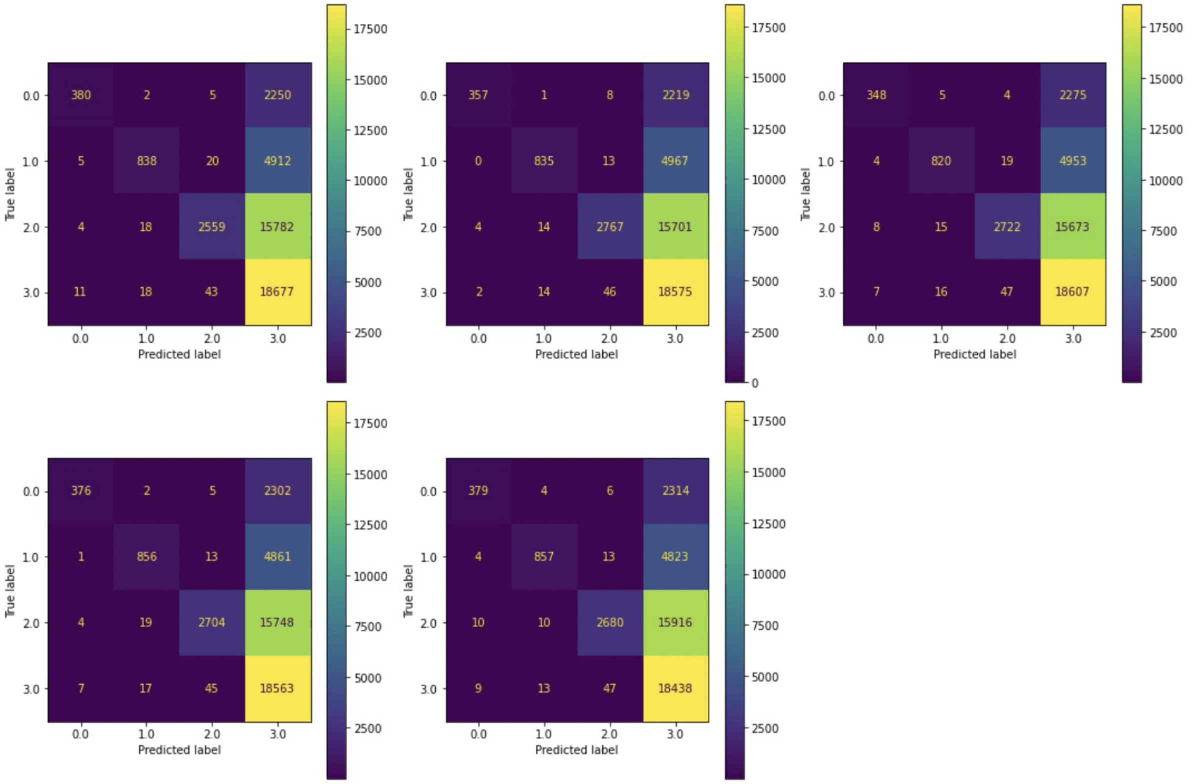


Figure 5. Confusion matrices of the Naive Bayes model

Random Forest

For the 5 fold cross validation, we obtained the classification report as the following table.

Table 3. Classification evaluation criterion for the *Random Forest* classifier

	Precision	Recall	F1-score
Average	0.680	0.482	0.384

From the table above, the precision, recall and F1-score of the decision tree model is 0.680, 0.482 and 0.384 respectively. The confusion matrices of the 5 fold cross validation results are shown as in Figure 6. The confusion matrix result is similar to the decision tree model and naive bayes model.

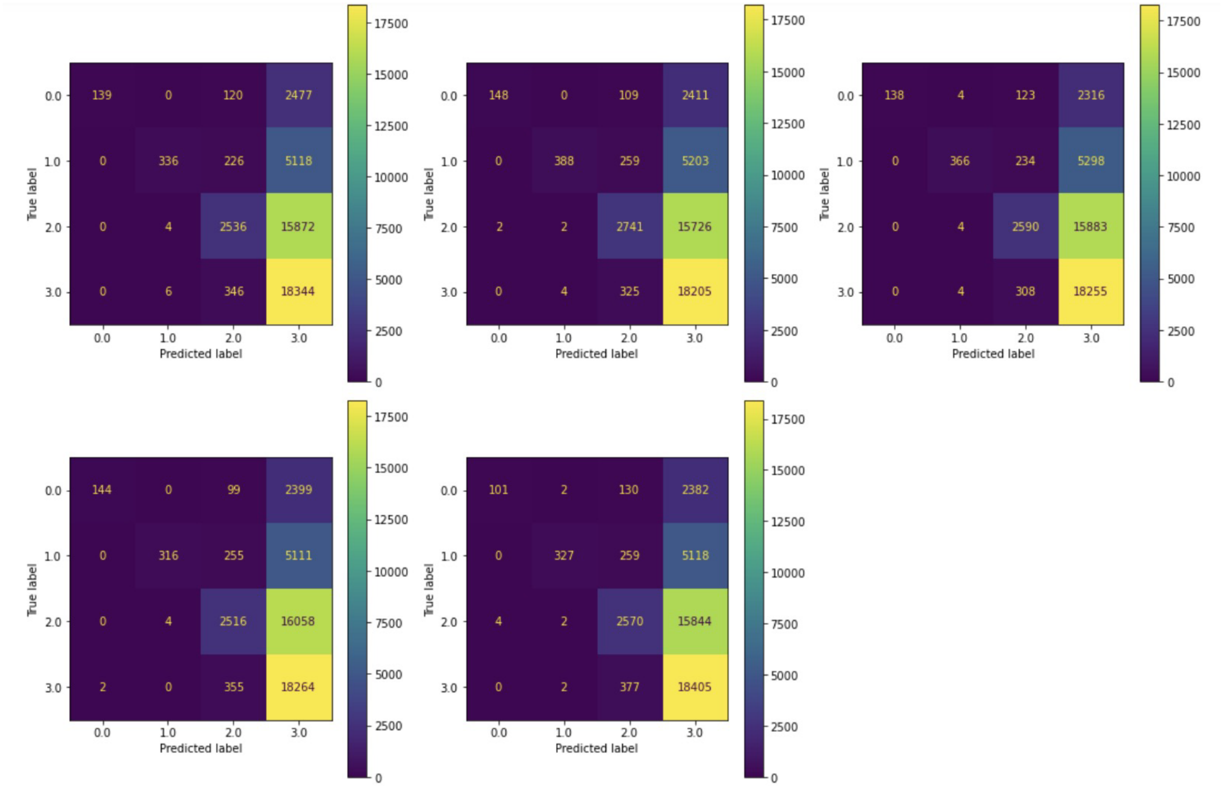


Figure 6. Confusion matrices of the Random Forest model

SVM

For the 5 fold cross validation, we obtained the classification report as the following table.

Table 4. Classification evaluation criterion for the *SVM* classifier

	Precision	Recall	F1-score
Average	0.476	0.416	0.245

The confusion matrices of the 5 fold cross validation results are shown as in Figure 9. We can find that the model is able to clearly identify Group 2, while it easily confuses Group 3 with Group 0 and 1.

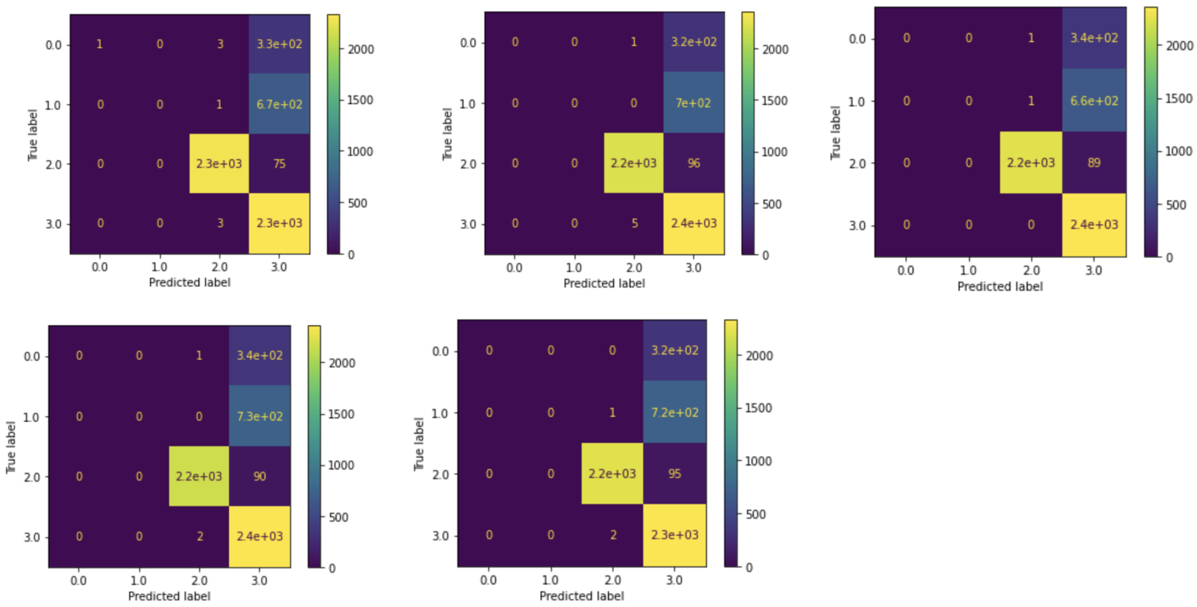


Figure 7. Confusion matrices of the SVM model

KNN Classification

For the KNN classification, here is the result from cross validation.

Table 5. Classification evaluation criterion for the *KNN* classifier

	Precision	Recall	F1-score
Average	0.872	0.872	0.872

The classification results of KNN are a lot better than previous models. The precision, recall and f1-score are 0.872. From the confusion matrix plots in Figure 8, we can see that all 4 classes are basically classified correctly with much fewer errors than the methods mentioned above. The KNN performs best among all the models we selected.

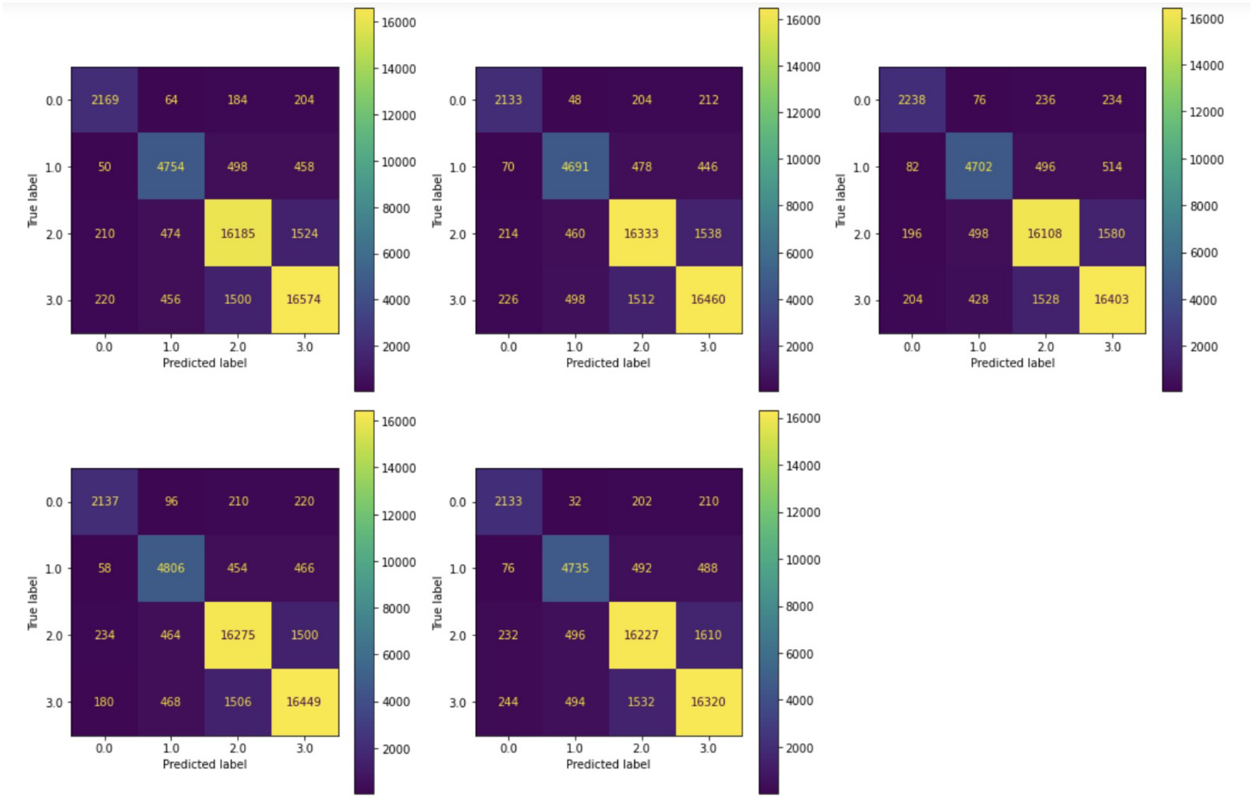


Figure 8. Confusion matrices of the KNN model

Conclusion

As we make progress towards our project, we started by data cleaning and using the PCA algorithm to further divide our dataset so that our analysis will become smoother. And we enacted several different classification methods such as SVM, Naive Bayes, Random Forest and KNN etc.. As a result, we found that the one that has the best outcome is the KNN. With 5 folds of cross validation process, KNN Method still remains a mean F1- score of 0.872. We think the reason for this is that KNN makes predictions just-in-time by calculating the similarity between an input sample and each training instance. As a result, this is very accurate to be used as a future prediction model. We believe that our model can be integrated into a good software like in apple stores or android stores, and it should be helpful to gather information for future customers to determine whether an airbnb fit their satisfaction before they reserved.

Future

We have used many models involved in the class so far, after reading the course modules, we decided to try to find a better way to perfect our model. We are considering neural networks, pipeline and parameter tuning in the future in order to perfect our project. Additionally, as we only studied the classification side of our dataset, we can try to use regression as the next step of our project to further exploit our data.

Individual Contribution to Project

Wenjun Han	● Introduction and problem statement ● Correlation, Normalization and PCA ● Modeling and Evaluation of Decision Tree and Naive Bayes
Geping Chen	● Dataset Description ● Data Cleaning and Feature Engineering ● Modeling and Evaluation of Random Forest and SVM
Weiting Li	● Methodology ● Evaluation ● Conclusion