Individual report

Introduction

We use the dataset from Kaggle, which is about Airbnb new user bookings. We only use the dataset called train_users_2 as training set and test_users as test set. And the label of this dataset is about the destination for the Airbnb users. In order to training the features in the dataset we use four models: navie bayes, decision tree, random forest and support vector machine. And the result shows that the most models among these four models is decision tree. I focus on data preprocessing to deal with missing value and create features and fit decision tree model.

My Individual Work

I am responsible for part of the model training part. I wrote code to train decision tree , random forest, and support vector machine.

Background of the algorithms

Random forest

In order to fix the overfitting problem of decision tree, random forest was proposed. Random forest is an ensemble method. Ensemble methods aggregated more or less predictions of many different algorithms in order to increase predictive accuracy, random forest is an ensemble of decision trees. Individual decision tree sometimes will not result in best decision, so we need to combine those weak learners to get an ensemble learner.

We first draw n bootstrap samples from the original data, then grow a classification tree for each of the bootstrap samples. At each node, rather than choosing the best split among all predictors, randomly sample m of the predictors and choose the best split from among those variables. Optimizing the algorithm could be achieved by setting the number of trees grown and track the change of classification error rate. Then we could predict new data by aggregating the prediction of n trees. Moreover, random forest could output features importance by determining which variables are closest to the root of tree and then moving out.

Support Vector Machine

Support vector machine was proposed by Vladimir N.Vapnik and Alexey Ya. Chervonenkis in 1963. The key objective of SVM is to draw a hyperplane that separates the two classes optimal. In a classification problem, there is the possibility of different hyperplanes. However the objective of SVM is to find the one which gives us the highest margin.

To maximize the margin, we need to minimize $\left(\frac{2}{\|w\|}\right)$ subject to

$$y_i\ (WX_i + b) - 1 \geq 0$$

for all i.

The final SVM equation can be written mathematically as

$$\mathrm{L} = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j X_i X_j$$

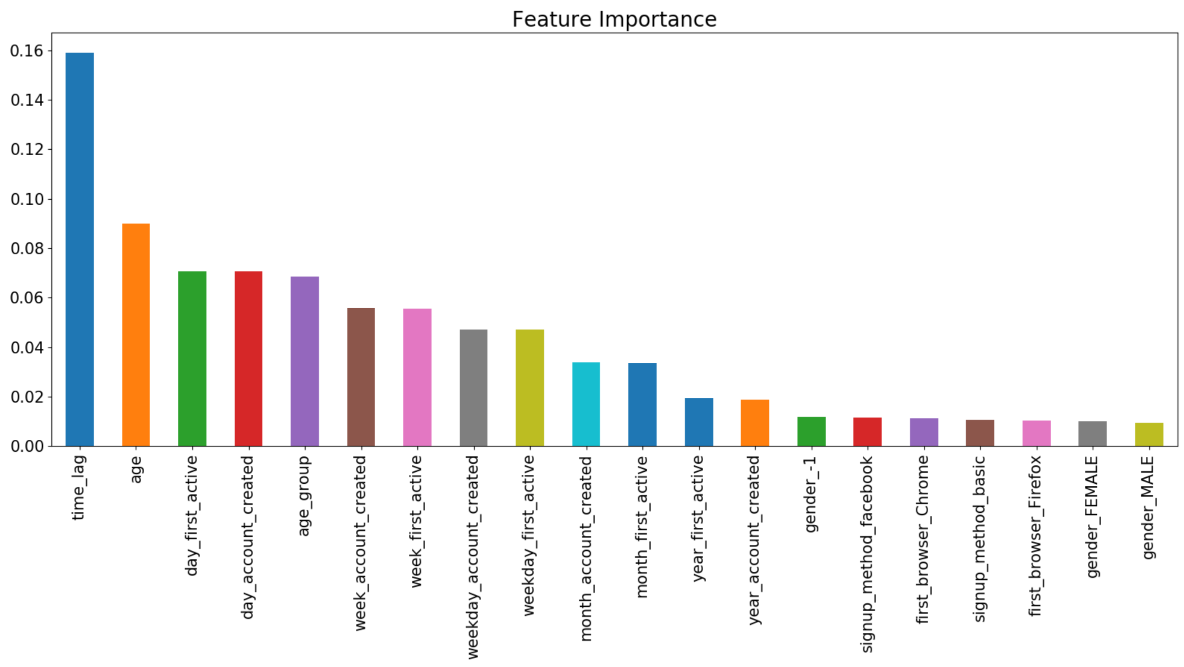
In 1992, Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik suggested a way to create nonlinear classifiers by applying the kernel trick (originally proposed by Aizerman et al.[14]) to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a nonlinerThis allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be nonlinear and the transformed space high dimensional; although the classifier is a hyperplane in the transformed feature space, it may be nonlinear in the original input space.

Results

i. Random Forest

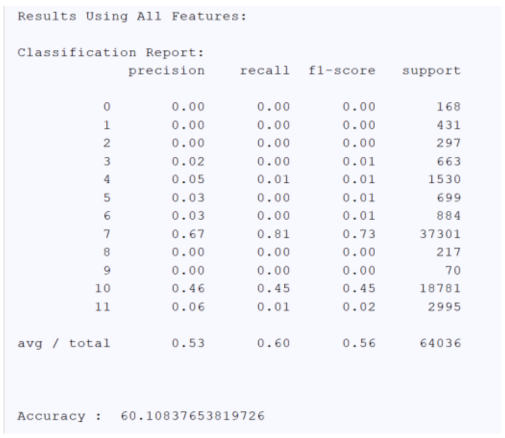
A random forest is simply a collection of decision trees whose results are aggregated into one final result. When inputting a training dataset with features and labels into a decision tree, it will formulate some set of rules, which will be used to make the predictions. Random Forest randomly selects observations and features to build several decision trees and then averages the results.

The result of feature importance of the attributes is shown below:

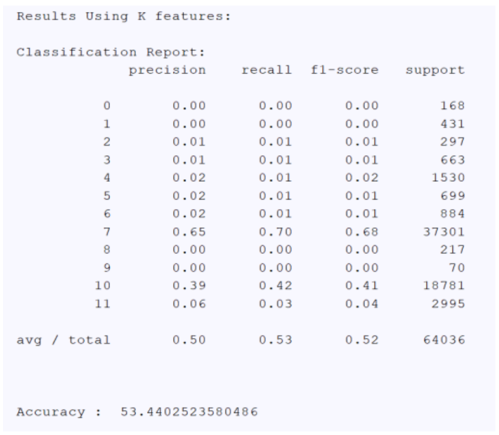


From the plot above, we can see that the most important feature is time_lag, which is the time difference between the users create an account to the users first browsing Airbnb. And the second significant feature is age, which indicates that the destination of Airbnb users' choice is sensitive to the age attribute. We just show the first twenty important features above. The result of feature importance also told us that sign-up method, the first browser the users used and the gender of users are not that important for the destination the users chose.

The result of the accuracy for the random forest as following:



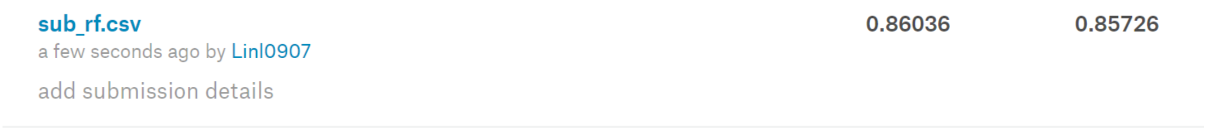
```
Results Using All Features:

Classification Report:
             precision    recall  f1-score   support

          0       0.00      0.00      0.00       168
          1       0.00      0.00      0.00       431
          2       0.00      0.00      0.00       297
          3       0.02      0.00      0.01       663
          4       0.05      0.01      0.01      1530
          5       0.03      0.00      0.01       699
          6       0.03      0.00      0.01       884
          7       0.67      0.81      0.73     37301
          8       0.00      0.00      0.00       217
          9       0.00      0.00      0.00        70
         10       0.46      0.45      0.45     18781
         11       0.06      0.01      0.02      2995

avg / total       0.53      0.60      0.56     64036



Accuracy :  60.10837653819726
```



```
Results Using K features:

Classification Report:
             precision    recall  f1-score   support

          0       0.00      0.00      0.00       168
          1       0.00      0.00      0.00       431
          2       0.01      0.01      0.01       297
          3       0.01      0.01      0.01       663
          4       0.02      0.01      0.02      1530
          5       0.02      0.01      0.01       699
          6       0.02      0.01      0.01       884
          7       0.65      0.70      0.68     37301
          8       0.00      0.00      0.00       217
          9       0.00      0.00      0.00        70
         10       0.39      0.42      0.41     18781
         11       0.06      0.03      0.04      2995

avg / total       0.50      0.53      0.52     64036



Accuracy :  53.4402523580486
```

By using all features in the dataset, the accuracy is 0.601. Then we used 15 most important attributes to fit the model. The result was supposed to be better than using all features, since we dropped the useless features and remain those who are significant. But the fact is, using 15 important features perform worse than using all of them. The accuracy reduced to 0.534 when using these features. It seems that the accuracy will decrease dramatically if we use less features to fit models, even if these features are not that important.

Then we tested the prediction in Kaggle made by using the model trained by all of the features. The result is shown below:



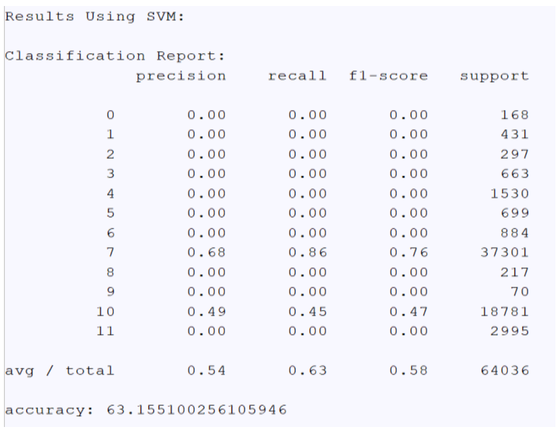
sub_rf.csv
a few seconds ago by Linl0907
add submission details

0.86036 0.85726

The result seems well in this classification problem. The score is no better than the one which got by decision tree algorithm, which is 0.86884. So random forest did not show its advantage in this dataset. Random forest theoretically performs better than decision tree. The reason maybe that the parameters that I set randomly are not the best one to fit this dataset. I think after tuning the parameter, random forest could give a higher score of accuracy.

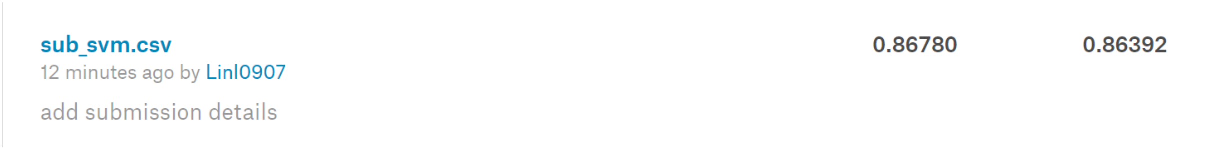
ii. SVM

The accuracy score of SVM is shown below, which indicates that it performs well.



```
Results Using SVM:

Classification Report:
             precision    recall  f1-score   support

          0       0.00      0.00      0.00       168
          1       0.00      0.00      0.00       431
          2       0.00      0.00      0.00       297
          3       0.00      0.00      0.00       663
          4       0.00      0.00      0.00      1530
          5       0.00      0.00      0.00       699
          6       0.00      0.00      0.00       884
          7       0.68      0.86      0.76     37301
          8       0.00      0.00      0.00       217
          9       0.00      0.00      0.00        70
         10       0.49      0.45      0.47     18781
         11       0.00      0.00      0.00      2995

avg / total       0.54      0.63      0.58     64036

accuracy: 63.155100256105946
```

And the score given by Kaggle is, which shows below.



sub_svm.csv
12 minutes ago by Linl0907
add submission details

0.86780 0.86392

Although SVM shows a high AUC score on Kaggle. I don’t think it is an appropriate algorithm according to this dataset. Since the dataset is too big to train and it is a multi-class problem, which leads time consuming when fitting the model. And it costs the computer more than 1 hour to train 70% of the dataset. Compared to decision tree, SVM is more computational expensive.

Summary and Conclusion

In this project I see some algorithms which ought to perform well but did not give a good accuracy score. Algorithms like support vector machine and random forest are both accurate methods which have been proved mathematically and was approved by a lot of data scientists. But they have a common problem when implementing to the real data that is computational expensive. Although they are accurate, we cannot use the whole data to train the model when the dataset is large. The model with part of the input will not perform well. So I think an algorithm with less accuracy but could been run fast on computers and handle big data is better.

Code Percentage

The percentage of code is 36.2%.

Reference

Python, P. D. A. U., & Swamynathan, M. Mastering Machine Learning with Python in Six Steps.