Predicting Red Hat Business Value

Machine Learning Engineer Nanodegree

Yu Hou
Oct 16th, 2016

1. Definition

Project Overview

Many companies has a great deal of information about the behaviors of customers, and they want to predict type of customers according to the specific action customers do. This is the project background.

The capstone project is from Kaggle[1].

Problem Statement

As previously described, Red Hat also gather a great deal of information about customers at specific time. And what we need to do is help them classify which customers have the most potential business value for Red Hat based on their activities and characteristics. So, we can conclude that this is a classification problem.

In this problem, four separate files are provided included people, activity of test, activity of train, and submission of example. A people file and a activity file may be merged together to get a new, unified data set for train and test.

For people file, each row represents a unique person, and each person has a unique people id. Unique person has many characteristics

For activity file, each row represents a unique activity performed by person at a certain date, and each activity has a unique activity id. Unique activity also has many characteristics.

Now I will elaborate the processing.

Step1: Explore the features relation of people file and activity filewith

outcomes

Step2: Handle some irrelevant features according to step1.

Step3: Merge people file and activity file after Step2 to get a new data table

Step4: Data table split, and model Training

Step5: Parameter Tuning

Step6: Using training model to predict test input data.

Metrics

As it is a typical classification problem, I will use AUC (Area under ROC Curve [2])between the predicted and the real outcomes. The AUC is strongly relevant about the confusion matrix, the True Positive Rate (TPR) and the False Positive Rate (FPR). The more detail about AUC can be founded in my quote.

The TPR and FPR can be described here.

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

$$FPR = \frac{FP}{TN + FP} \tag{2}$$

TP is True Positive, FP is False Positive, FN is False Negative.

2. Analysis

Data Exploration

The data set can be acquired from Kaggle, and I will provide this for report. For activity file, it has 15 columns, and people file has 41 columns. If we don't explore the relation features between outcomes, the new data table merged people and activity file will has 55 columns. If all of them are dependent, the relevant group numbers we should find is 55!. This is a huge number and increase the analysis difficulty and calculation complexity. So we make a hypothesis that all of features is irrelevant and independent, this is convenient for us to analyze the relation of every feature and outcomes.

We will give some data about the people and activity file to help reader get an intuitive understanding.

people_id	char_1	group_1	char_2	date	char_3	char_4	char_5	char_6	char_7	...	char_29	char_30	char_31	char_32	char_33	char_34
ppl_100	type 2	group 17304	type 2	2021-06-29	type 5	type 5	type 5	type 3	type 11	...	False	True	True	False	False	True
ppl_100002	type 2	group 8688	type 3	2021-01-06	type 28	type 9	type 5	type 3	type 11	...	False	True	True	True	True	True
ppl_100003	type 2	group 33592	type 3	2022-06-10	type 4	type 8	type 5	type 2	type 5	...	False	False	True	True	True	True
ppl_100004	type 2	group 22593	type 3	2022-07-20	type 40	type 25	type 9	type 4	type 16	...	True	True	True	True	True	True
ppl_100006	type 2	group 6534	type 3	2022-07-27	type 40	type 25	type 9	type 3	type 8	...	False	False	True	False	False	False

Fig 1.Biref People Data

From the figure 1, we can know that people file have group id, people id, data, and char 1 to char 38(figure only display char type from 1to 34) these features.

people_id	activity_id	date	activity_category	char_1	char_2	char_3	char_4	char_5	char_6	char_7	char_8	char_9	char_10	outcome
ppl_100	act2_1734928	2023-08-26	type 4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	type 76	0
ppl_100	act2_2434093	2022-09-27	type 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	type 1	0
ppl_100	act2_3404049	2022-09-27	type 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	type 1	0
ppl_100	act2_3651215	2023-08-04	type 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	type 1	0
ppl_100	act2_4109017	2023-08-26	type 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	type 1	0

Fig 2.Biref Activity Data

From the figure2, we have the basic concept that same people may do different or same activity on different time. Compared with figure 1, we know that there some features are same in people and activity file, and the same features may have different values. In figure2, column outcome is the final label we would to predict.

Exploratory Visualization

From the figure 1 and 2, we can know that the data of people and activity file can’t directly be used to display. So data will be processed to make visualization.

We will show limited graphs to observe the relationship of feature and outcome for people and activity file.

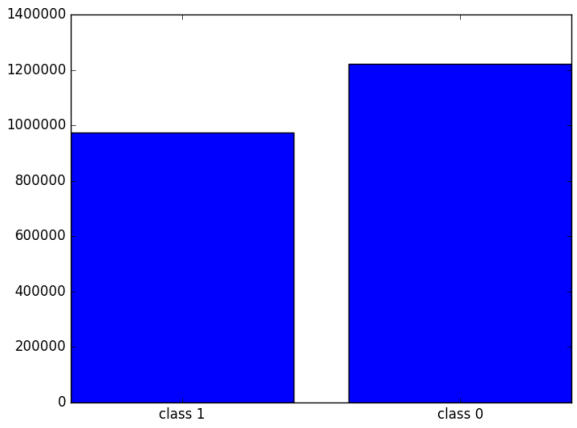

Fig 3.Numbers of class 1 and 0

Figure 3 show us that numbers of outcome 1 and 0 are almost the same. This means that the distributed outcome is balanced, so we do not shuffle them.

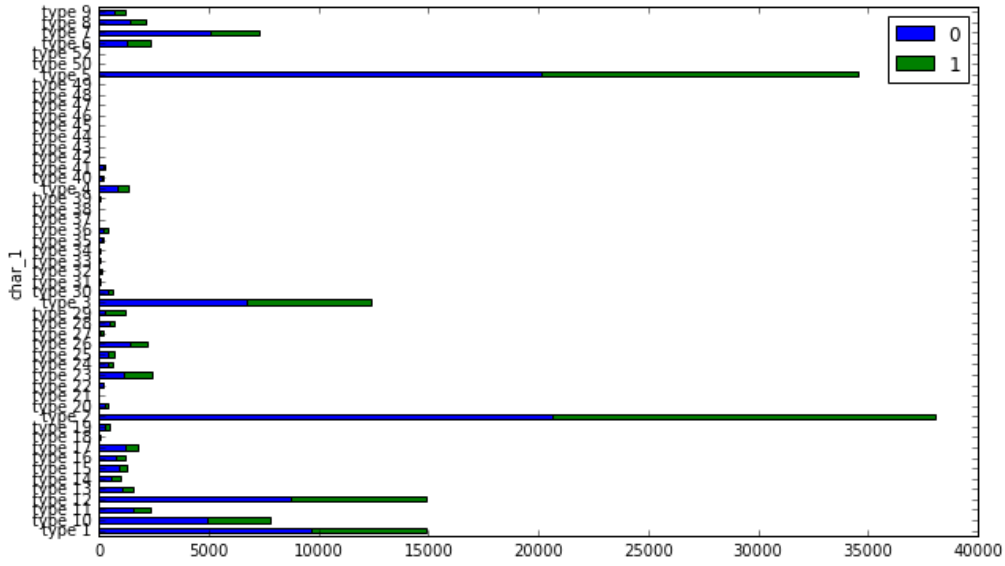

Fig 4. The relationship of char_1's type and outcome

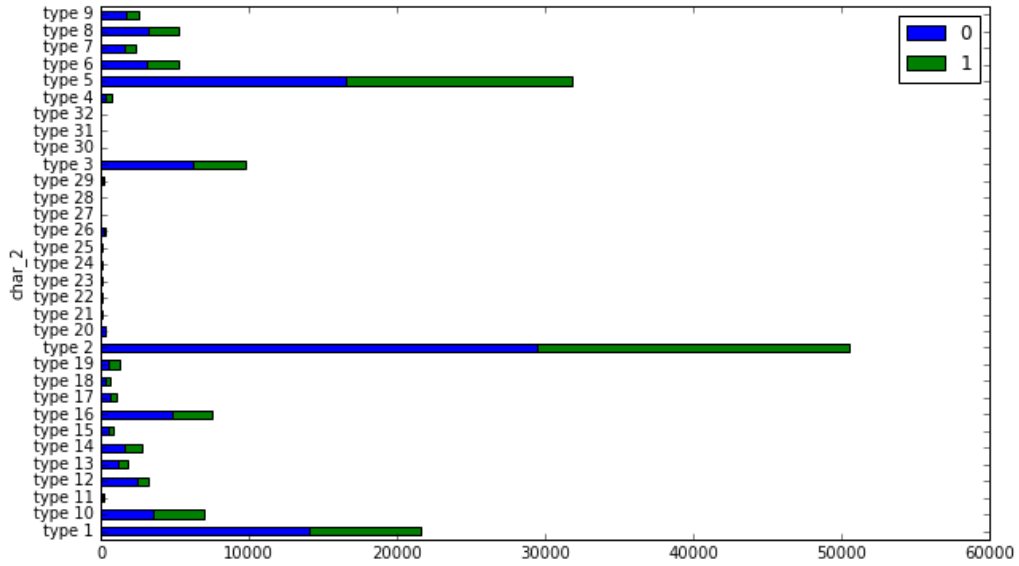

Fig 5. The relationship of char_2's type and outcome

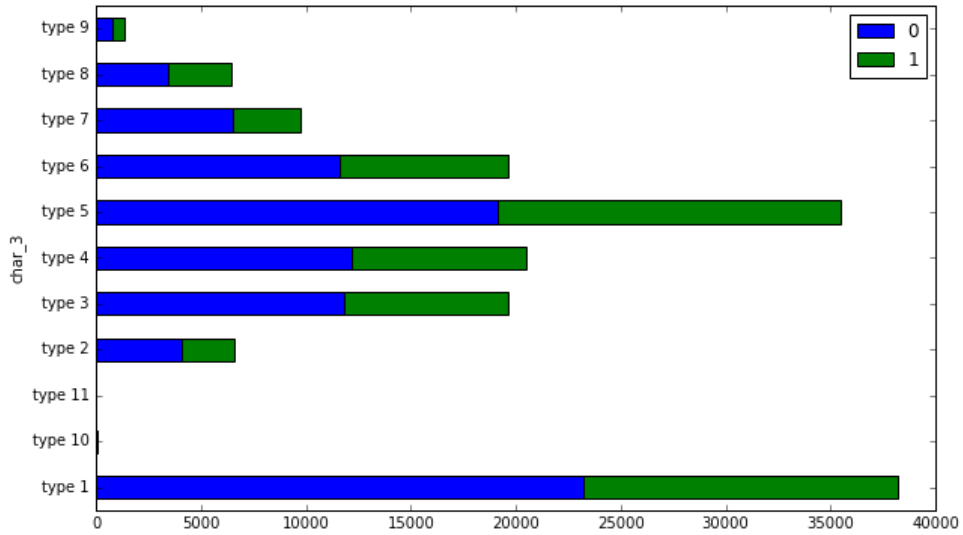

Fig 6. The relationship of char_3's type and outcome

From figure 4 to 6, three figures show that the relationship between different type of char_1 to char_3 and outcome. The more details about this can be found in my appendix. In fact, these graphs tell me that char features in activity file don't have import influence on result of outcomes. And char_10 in activity file has many types, then char_10 in people file is Boolean type. So, I would drop char features in activity file for these features are not very irrelevant. For people file, I use the same method to find the relevance of char features in people and outcome in activity file.

In people file the char features varies from 1 to 38, and number 1-10 and 38 is numeric types, number 11-37 are Boolean type. To display every char's relevance is so trouble, so I would only choose several char type from one to thirty-eight.

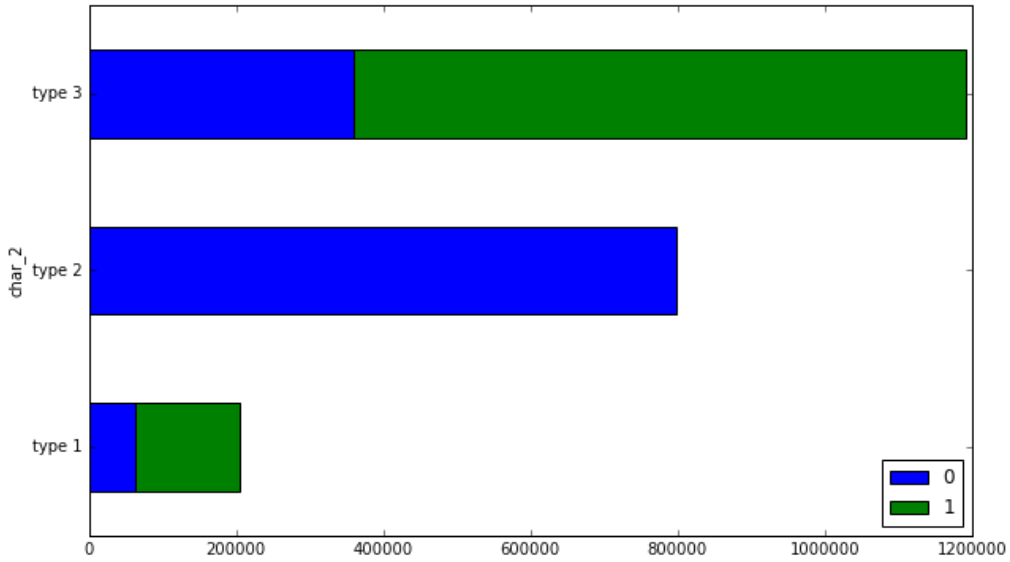

Fig 7. The relationship of char_2's type and outcome (people file)

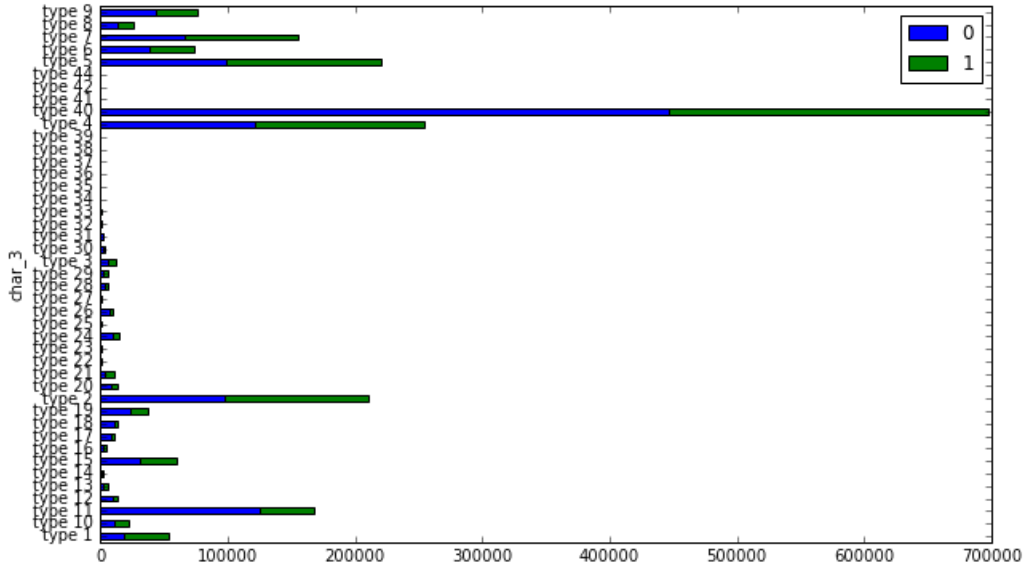

Fig 8. The relationship of char_3's type and outcome (people file)

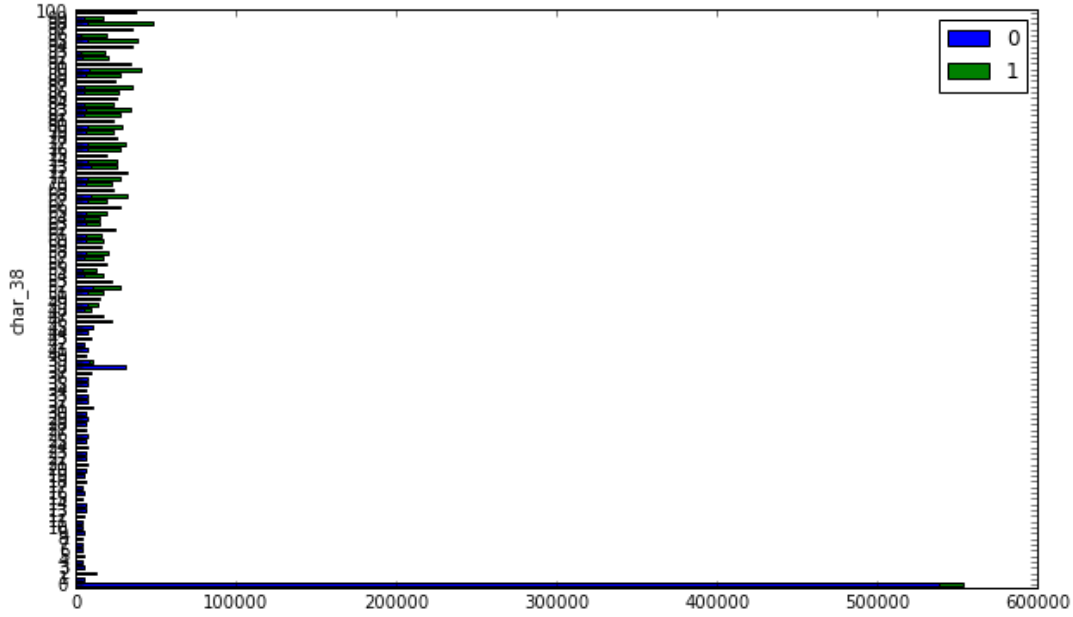

Fig 9. The relationship of char_38's type and outcome (people file)

From these three graph, we can get this information the through char_2 and char_38, the classification is easy to get, but if we use char_3, it's hard to get an accuracy result. Other char types are same as char_3 that not be simply used to get classification. Details about other char types are visible in my appendix.

Algorithm and Techniques

First of all, we will extract some important features, which have strong influence on outcomes, as the input data to train some model. So this means that some features in people and activity file will be dropped. The dropped features are selected through the graph of feature and outcomes.

This problem is a typical classification problem. So I will try Decision Tree [3],

Random Forest [4]. Their characteristic are following:

Algorithm	Advantage	Disadvantage
Decision Tree	Easy to understand, to interpret, not sensitive to missing value	Easy to over fit
Random Forest	Easy to understand, feature selection not necessary, robust to noise	Prone to the cluster which has more samples

Benchmark

In the kaggle competition, 1st has the 99% accuracy, and almost people can get 94% accuracy. So I set 95% accuracy as my baseline.

3. Methodlogy

Data Preprocessing

In this section, I write a data preprocess function to process activity and people file, and use merge function in pandas to get train and test table. This implementation can be seen in my appendix. There are some figures give the data after preprocessing and merging.

	people_id	activity_id	date	activity_category	outcome
0	100	1734928	2023-08-26	4	0
1	100	2434093	2022-09-27	2	0
2	100	3404049	2022-09-27	2	0
3	100	3651215	2023-08-04	2	0
4	100	4109017	2023-08-26	2	0

Fig 10. Activity train file after preprocessing

Figure 10 shows us the data form, the difference between activity train and test file is lack of outcome column.

people_id	char_1	group_1	char_2	char_3	char_4	char_5	char_6	char_7	char_8	...	char_12	char_15	char_18	char_28	char_30	char_37
100	2	17304	2	5	5	5	3	11	2	...	0	0	0	1	1	0
100002	2	8688	3	28	9	5	3	11	2	...	1	0	0	0	1	0
100003	2	33592	3	4	8	5	2	5	2	...	1	1	0	1	0	1
100004	2	22593	3	40	25	9	4	16	2	...	1	0	1	1	1	1
100006	2	6534	3	40	25	9	3	8	2	...	0	0	0	0	0	0

Fig 11. People file after preprocessing

	people_id	activity_id	activity_category	outcome	year_x	month_x	day_x	\
0	100	1734928	4	0	2023	8	26	
1	100	2434093	2	0	2022	9	27	
2	100	3404049	2	0	2022	9	27	
3	100	3651215	2	0	2023	8	4	
4	100	4109017	2	0	2023	8	26	

	char_1	group_1	char_2	...	char_12	char_15	char_18	char_28	\
0	2	17304	2	...	0	0	0	1	
1	2	17304	2	...	0	0	0	1	
2	2	17304	2	...	0	0	0	1	
3	2	17304	2	...	0	0	0	1	
4	2	17304	2	...	0	0	0	1	

	char_30	char_37	char_38	year_y	month_y	day_y
0	1	0	36	2021	6	29
1	1	0	36	2021	6	29
2	1	0	36	2021	6	29
3	1	0	36	2021	6	29
4	1	0	36	2021	6	29

Fig 12. Activity Train data after merging

Implementation

I adapted Decision Tree and Random Forest as training algorithm to train the data, and the metrics is AUC. The parameters of Decision Tree and Random Forest are default at the beginning. First, I imported the function called train_test_split() from sklearn.cross_validation to split the train data into two groups, then I importd DecisionTreeClassifier() from sklearn.tree, trained and testd them on the train data, predicted result on the test data. I also imported RadomForestClassifier() from sklearn.ensemble, trained and tested them on the train data, and predicted result on the teat data. Finally, I would upload the result file to Kaggle website to get the predicted accuracy.

In this situation, the complex algorithm – Random Forest result is better than Decision Tree. The accuracy of Random Forest is 0.94, the accuracy of Decision Tree is 0.84 in Kaggle public and private score. In terms of algorithm timing, Random Forest algorithm give me an obvious sense that it is slow than Decision Tree.

Fig 13. Comparison of Decision Tree and Random Forest

Refinement

I imported grid search function from sklearn for tuning the parameter of Decision Tree and Random Forest algorithm. In fact, Random Forest is based on Decision Tree.

So the parameter used to tune for them is same. I choose the max depth as the tuning parameter. The max depth is consists of 3, 4, 5.

After tuning parameter, Decision Tree and Random Forest has a different result for the same parameter. The accuracy of Decision Tree increases from 0.84 to 0.94. The accuracy of Random Forest decrease from the 0.94 to 0.90. The detailed accuracy score is showed in the below.

Post-Deadline: Sat, 22 Oct 2016 09:12:41 Edit description	redhat_dt_tune.csv	0.906125	0.905916
Post-Deadline: Sat, 22 Oct 2016 09:10:47 Edit description	redhat_rf_tune.csv	0.910670	0.909966

Fig 14. Comparison of Decision Tree and Random Forest after tuning

So, the default Random Forest has the best performance. If we want to get a better model or increase the accuracy, the feature selection is very important.

4. Results

Model Evaluation and Validation

We used AUC to estimate the performance of model that Decision Tree and Random Forest trained. We would investigated the AUC curve of tuning and before tuning. But this curve is displayed by the train data which is into training model data, and testing model data through train_test_split function from sklearn.

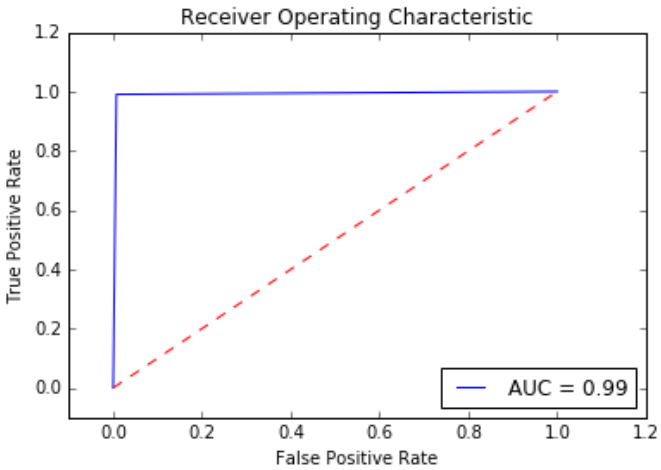

Fig 15. Decision Tree AUC before tuning

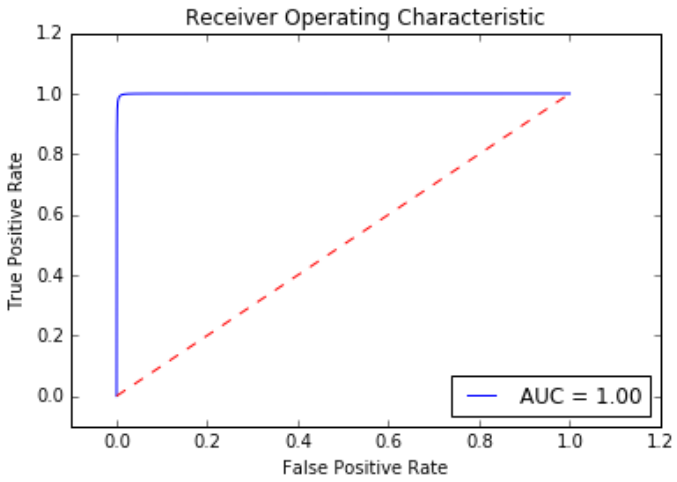

Fig 16. Random Forest AUC before tuning

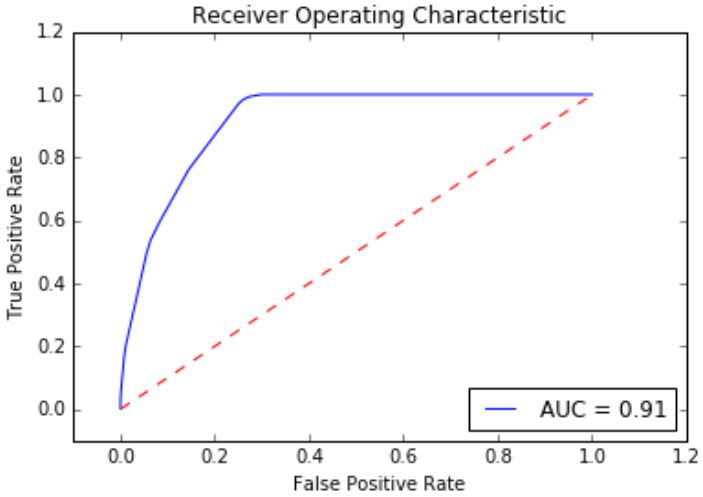

Fig 17. Decision Tree AUC after tuning

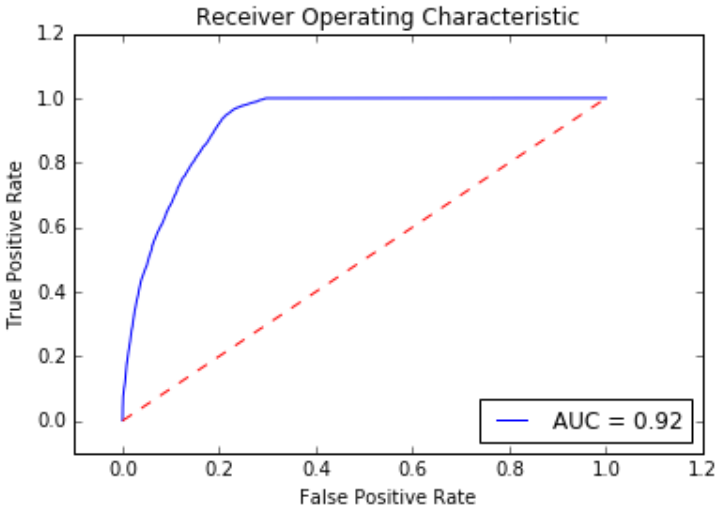

Fig 18. Random Forest AUC after tuning

The AUC curve give us a direct display. After tuning, the accuracy of two algorithm is closely to the real accuracy of predicting activity test data.

Justification

The AUC curve and the accuracy of activity data is relative closely. This means that the model performance of Decision Tree and Random Forest is pretty. But I hold the opinion that the Random Forest can get a better performance through the reasonable feature selection and parameter tuning for someone have a 99% accuracy on activity test data in Kaggle.

5. Conclusion

Free-Form Visualization

In this section, I will give the test size's influence on AUC curve, and give some graph for different test size.

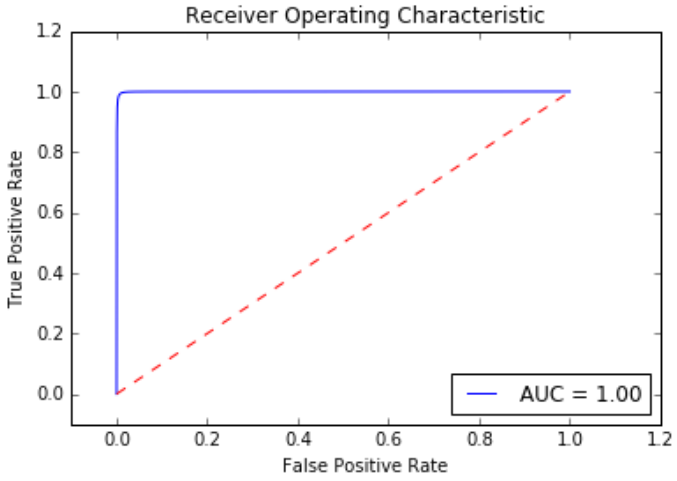

Fig 19. Random Forest AUC test_size = 0.1

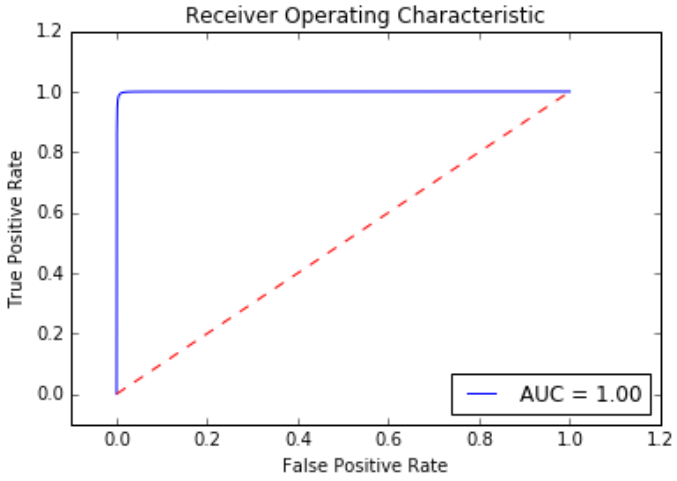

Fig 20. Random Forest AUC test_size = 0.3

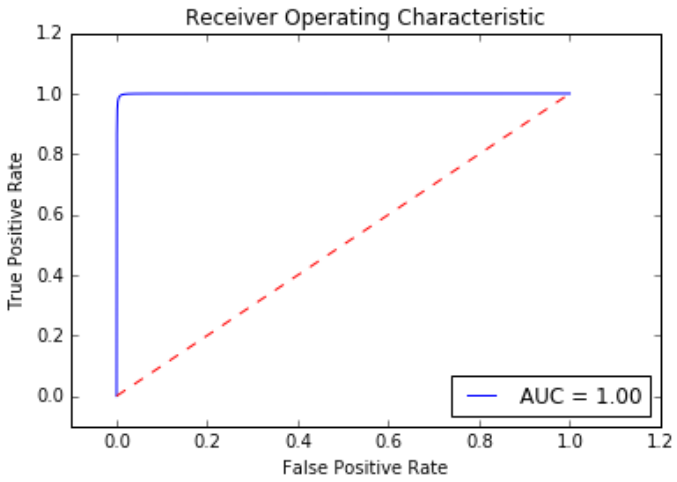

Fig 21. Random Forest AUC test_size = 0.4

When the test size varied from 0.1 to 0.4, the AUC curve of Random Forest is almost same, and the value of AUS is 1.But the accuracy of activity test data is different. When test size equals 0.2, the model has the best performance among them.

Reflection

This project is to solve a binary classification problem to predict the potential customer through the activity id. I just used some features whose total number is 55 from the activity and people file, and I hold the hypothesis that every feature is independent to handle data more convenient. Then I write some function to do data pre-processing and show the relation of feature and outcome. I used Random Forest and Decision Tree to train the data and get AUC score, curve.

Then I used the max depth as the parameter to tune Decision Tree and Random Forest. The performance after tuning on two algorithm have an opposite result on activity test data. I adapted the test size for Random Forest for knowing its influence on real accuracy on activity test data.

From different dimensions, Random Forest' model has a better performance than Decision Tree.

Improvement

In fact, the activity train file's data should be used to train model. But I don't know the real outcome of activity test data. I have to spit the activity train data to get the AUC score of trained model. This would lead that training is inadequate, the performance trained model on activity test data not accurate enough.

In addition, I hold the feature is independent, and not combine features to analysis its influence on outcome.

Furthermore, I should make some considerations on features selection and parameter tuning.

6. Reference

[1] https://www.kaggle.com/c/predicting-red-hat-business-value
[2] https://en.wikipedia.org/wiki/Receiver_operating_characteristic
[3] https://en.wikipedia.org/wiki/Decision_tree_learning
http://scikitlearn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier
[4] https://en.wikipedia.org/wiki/Random_forest
http://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html