Predicting Red Hat Business Value

Machine Learning Engineer Nanodegree

Yu Hou
Oct 16th, 2016

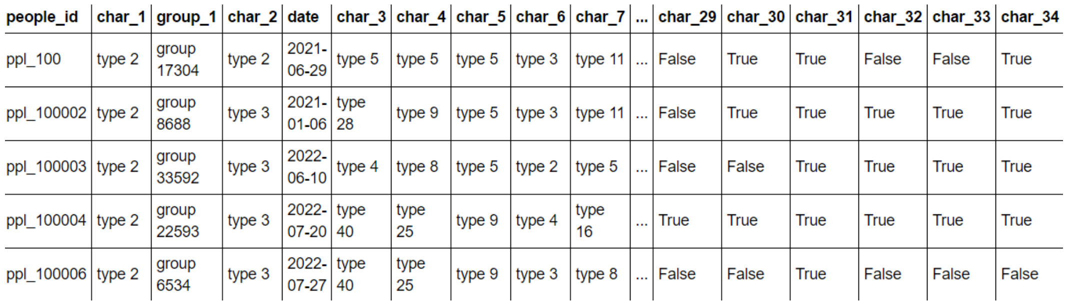
1. Definition

Project Overview

The capstone project is from Kaggle[1].

As most companies, Red Hat can get a lots of information about over time about the behavior of customers who interact with them. They are looking for some better methods to use these behavioral data to predict which individuals they should take some action to approach and even know what time and how to approach them.

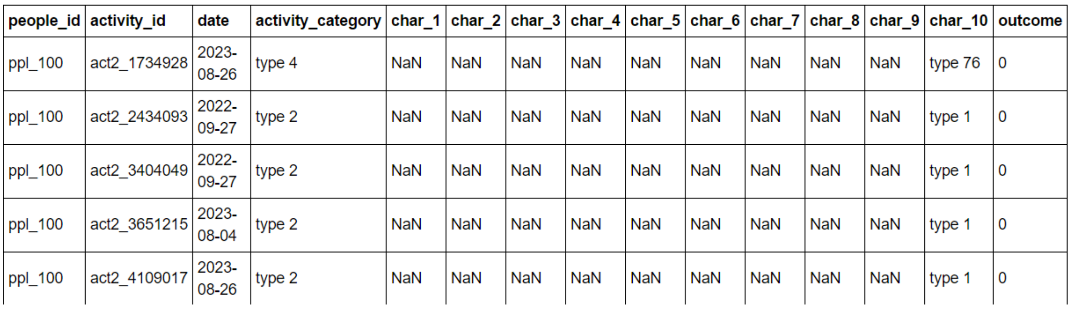
From the above, we can know that this problem domain is customer behavior analysis, and through analysis to make some business decision. In this problem, data set is composed of people data and activity data.



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

People data includes all of unique people and the corresponding characteristics, which do some activities over time. Each row in the people data represents a unique person. Each person has a unique people_id.



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

Activity data includes all of unique activities and the corresponding characteristics, which do some activities over time. Each row in the activity data represents a unique activity that a person have performed on a certain date. Each activity has a unique activity_id.

People data and activity data can be joined together to get a single, unified data table as the final data set.

Problem Statement

In this problem, the challenge for us is to create a classification algorithm that can accurately identify which customers have the most potential value for Red Hat based on their characteristics and activities. The most potential for Red Hat means that which customers can buy the service or article. The most potential as predicted value is range from 0 to 1, and more close to 1, the customer has more potential for Red Hat. More accuracy, this a classical binary classification problem for the business value outcome of customers is defined yes or no attached to each unique activity

Now I will elaborate the processing.

Step1: Explore the features relation of people file and activity file with outcomes

Step2: Handle some irrelevant features according to step1, then Merge people file and activity file after Step2 to get a new data table

Step3: Data table split, and model Training

Step4: Parameter Tuning

Step5: Using training model (including Decision Tree and Random Forest) to predict test input data.

Metrics

In Kaggle, this completion is evaluated on area under the ROC [2] curve between the predicted. In sklearn, the function is sklearn.metrics.roc_auc_score. This function 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.

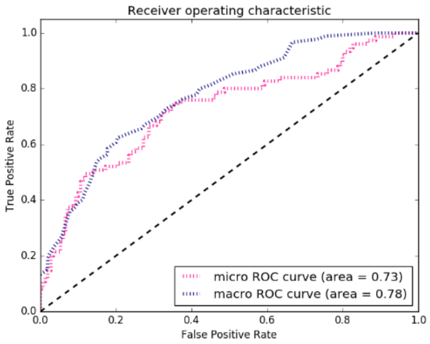


Fig 3.The example of ROC and area

The reason using the area under ROC is that if ROC of different models cross, it's hard to identify the performance of models, and here area is a pretty measure to evaluate the performance of models. The area under ROC curve really represents the performance of learning model. The area is larger, the model's performance is better.

And the running time is also a measure to compare the algorithm time cost. The time module in Python is used to get the time cost.

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.

In this data set, there are 189118 people samples and 2197291 activities samples. And Figure 1 and 2 tell us that the people and activity data is not numeric, Boolean and Char type. So the regular statistical description such as mean, median max, min, can’t show the relation of features and outcomes. I will calculate the percentage of different types including outliers and missing value for different features and binary classification.

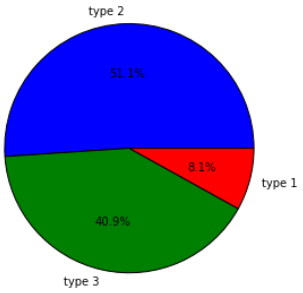


Fig 3.percentage of different types for people char_2 feature

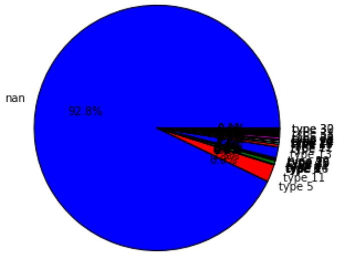


Fig 4.percentage of different types for activity char_2 feature

We can know that char_2 feature in people have 3 different attributes and approximate quantity, there are no missing value and outliers for feature char_2. Other features in people data also can be analyzed through this method and graphs are displayed in the code file.

From figure 4, NAN is the vast majority of char_2 feature in activity data, and its types is more than char_2 feature in people data. So when people and activity data is joined together, it’s important to explain to how to handle the same feature for different data table, and missing value-NAN. Other features can be handled by the same way.

Exploratory Visualization

From the figure 1 and 2, we can know the data formats of people and activity data, it’s hard to observe the features’ impact on outcome and the relevance of different features. 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 data.

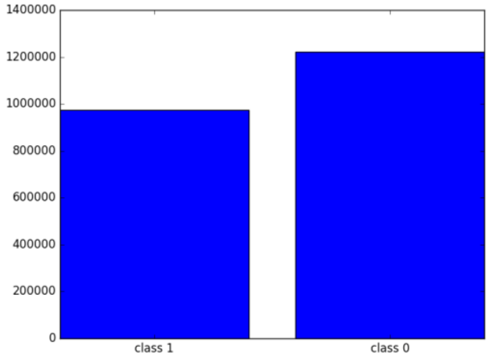


Fig 5.Numbers of class 1 and 0

Figure 3 show us that numbers of outcome 1 and 0 are almost the same. This means that the distributed outcome is balanced. But it is worth recommending to shuffle data set to avoid test set or train set is not uniform.

Now, we will show little graphs to show the relationship of outcome and a single feature.

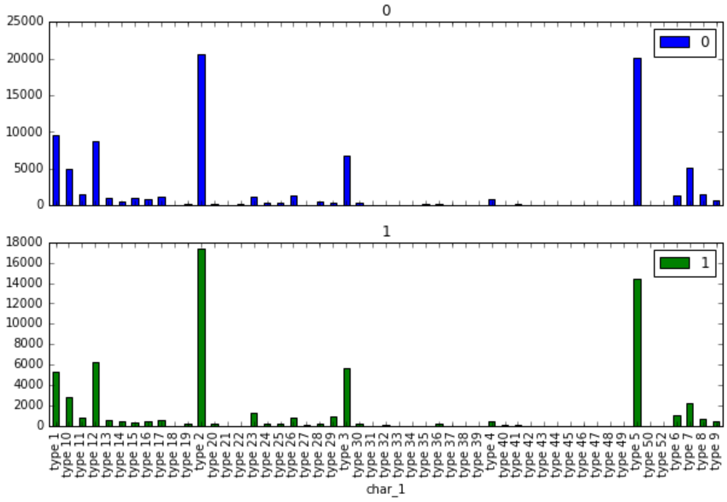


Fig 6. The relationship of char_1’s type and outcome

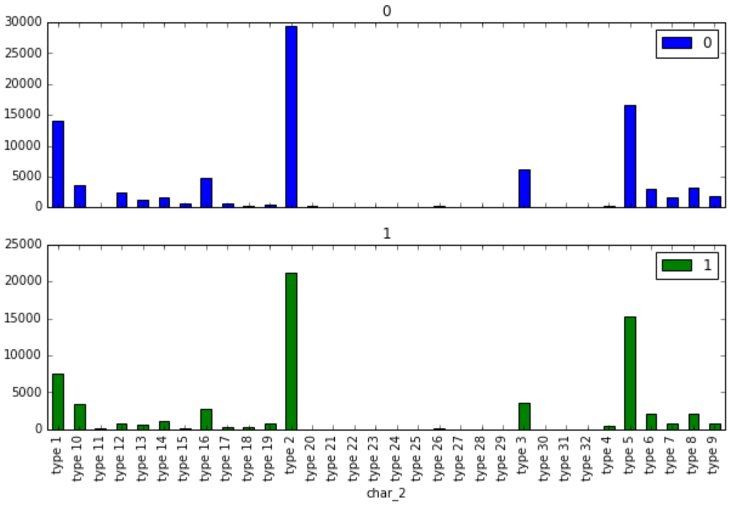


Fig 7. The relationship of char_2's type and outcome

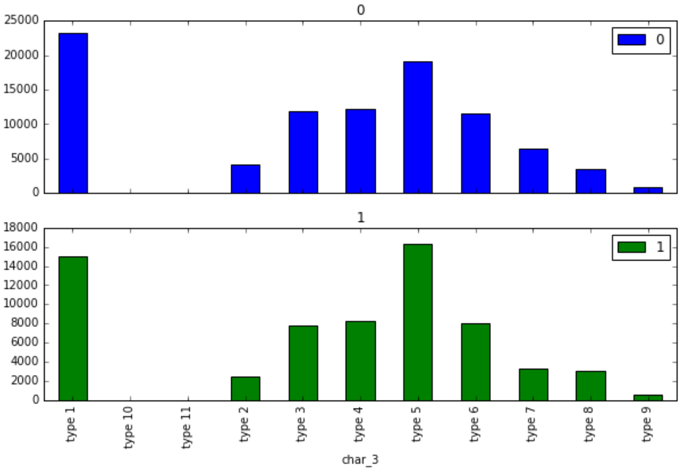


Fig 8. The relationship of char_3's type and outcome

From figure 6 to 8, 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, it's hard to only use char features to identify customer potential. Otherwise, we find that almost char features (deleting the missing value) total number is not equal 2197291 for the missing value is the majority in char features of activity data. The figure 9 shows char_3' percentage for different types (including the missing value). This result is common from char_1 to char_9, the missing value is the majority. So these char_1 to char_9 in activity features would be dropped.

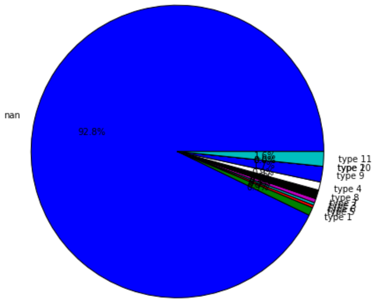


Fig 9. The percentage of char_3's types

And char_10 in activity file has many types that's hard to display in a graph, then char_10 in people file is Boolean type. The char_10 is conflicted in activity and people data. So I drop char_10 feature both in two files. For people data, 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.

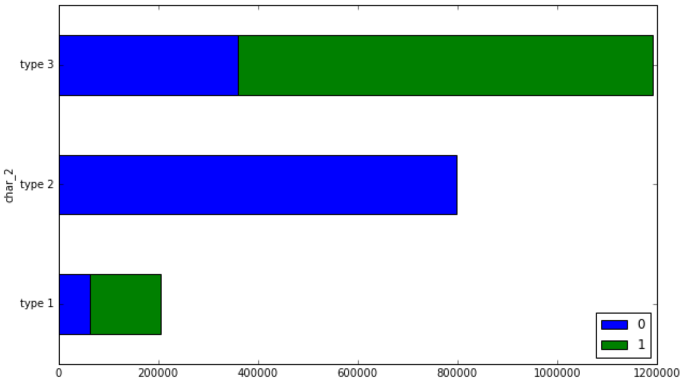


Fig 10. The relationship of char_2's type and outcome (people file)

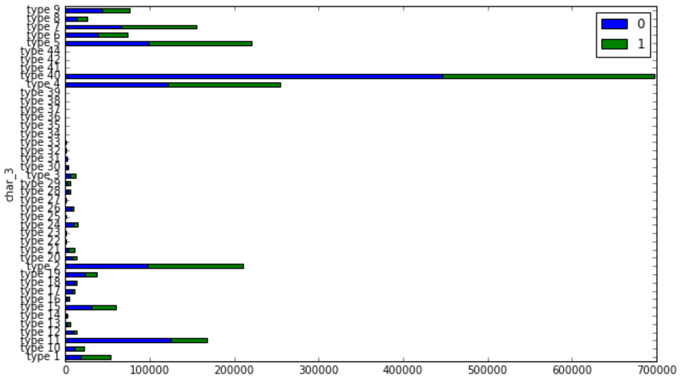


Fig11. The relationship of char_3's type and outcome (people file)

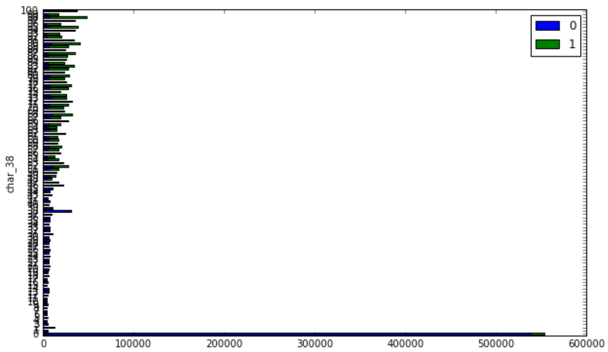


Fig 12. The relationship of char_38's type and outcome (people file)

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. I would drop these features as we do in activity data.

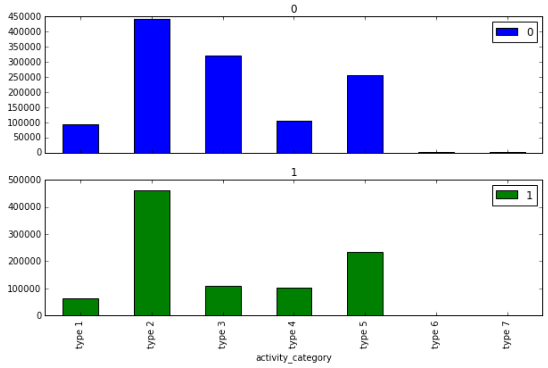


Fig13. The relationship of activity category's type and outcome (data file)

Figures shows other features with same analysis and handled method. Details about other features 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

From the part exploratory visualization, we show use some features to get the graph of relationship between the feature and outcomes and the percentages of missing value in whole feature. For example, the feature char_2 in people data is easy to identify the outcomes, so I regard this as a strong feature. On the opposite, char_3 in people data and activity category in activity data are hard to distinct the customer's potential, so I think that it is an irrelevant feature that can be dropped.

Others features like char_2 and char_3 in people and activity data is evaluating its importance through a function called get_relation (the function give a relation graph of feature and outcome, located in the code file). For features including NAN, we should calculate the NAN's percentage in this feature. If its percentage is over 50%, we would drop it. Then using the get_relation function to determine whether drop it. The final step, if this feature is reserved, we would set NAN as 0. In this problem, NAN in features is over 90%, so we drop it.

For categorical features, we would turn them to numeric features. And then, we will normalize the numeric features for training (function name Normalization is defined in code file). In this problem, we use k-fold cross validation, so we would run 10 times for training model.

I would explain why I don't use the feature_selection module built in sklearn. The operation such as SelectKBest, SelectPercentile in module is like a black box for us. If we use a function defined by ourselves, we can know all details and what we want to observe and looking for. And it also gives us some advantages to understand the feature selection.

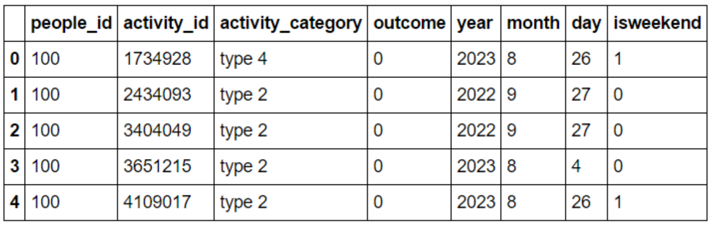
Benchmark

In Kaggle, this competition is used the area under ROC curve as a measurement (corresponding to metrics.roc_auc_score function in sklearn). So, we would use the same method to measure validation and test data. For validation set, the area should be more than 0.98. For test set, the area should be more than 0.94 as 1st is more than 0.99, and most people's area is about 0.94.

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	activity_category	outcome	year	month	day	isweekend
0	100	1734928	type 4	0	2023	8	26	1
1	100	2434093	type 2	0	2022	9	27	0
2	100	3404049	type 2	0	2022	9	27	0
3	100	3651215	type 2	0	2023	8	4	0
4	100	4109017	type 2	0	2023	8	26	1

Fig 14. Activity train file after preprocessing

Figure 14 shows the activity data after preprocessing. We don't know the new features (including year, month, day and isweekend) influence on outcome.

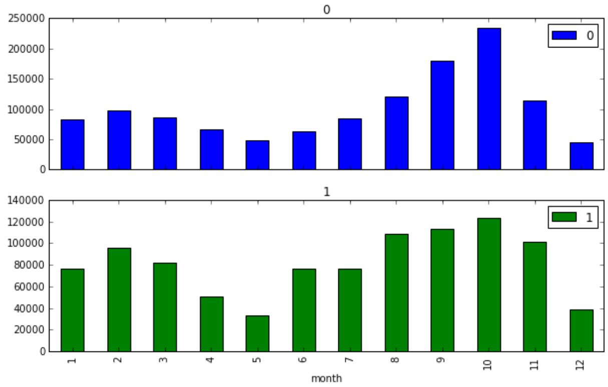


Fig 15. Month's impact on outcome of activity data

Just figure 15 and 13 shows that activity category and moth is not strong relevant to outcome, as well as year, day and isweekend. The activity table last holds that activity id, people id and outcome.

In part Exploratory Visualization, we know that features in people data only group_1, char_2, char_38 have a strong relation with outcome. As for date feature, the below graph shows us the result. We would drop the features about date.

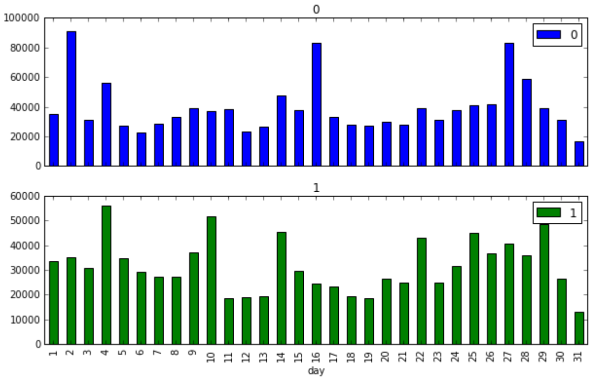
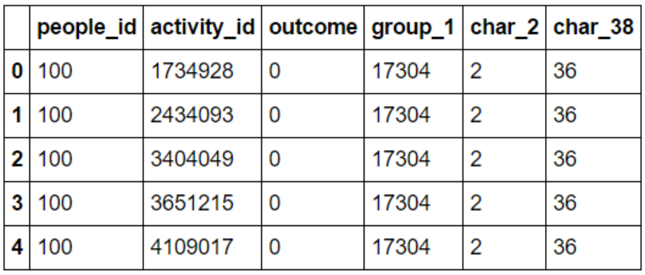


Fig 16. Day's impact on outcome of people data

In the final, the train and test data including should be the below format.



	people_id	activity_id	outcome	group_1	char_2	char_38
0	100	1734928	0	17304	2	36
1	100	2434093	0	17304	2	36
2	100	3404049	0	17304	2	36
3	100	3651215	0	17304	2	36
4	100	4109017	0	17304	2	36

Fig 17. Useful data set format

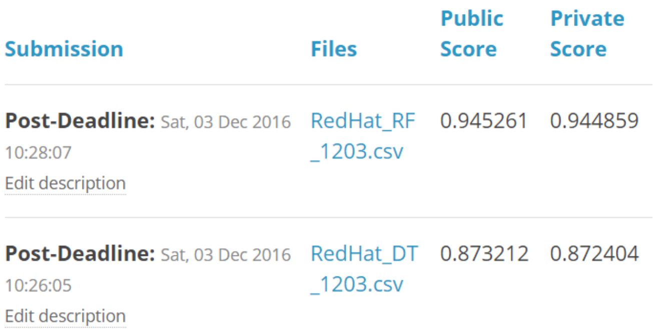
When we get the useful data set, we would do the last step to normalization to zoom the values of features for training and testing. The normalization function is implemented in the appendix, more details can be seen.

Implementation

I adapted Decision Tree and Random Forest as training algorithm to train the data, and the metrics is area under the ROC curve which is measured by ruc_aoc_score in skelearn. The parameters of Decision Tree and Random Forest is set max_features used log2, max depth used 3. The reason I used Decision Tree and Random Forest in this problem is that Decision Tree is convenient for training classification problem, and easy to understand, Random Forest is representative ensemble learning.

First, I used sklearn.cross_validation.KFold method to do cross validation for generalising ability. Then, I initialized the Decision Tree and Random Forest to train the training data set. After this step, I used the trained model including to predict the test data, and get two output files. 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 area of Random Forest is 0.94, the area of Decision Tree is 0.87 in Kaggle public and private score. The upper one is the result of Random Forest, and the below is Decision Tree result.



Submission	Files	Public Score	Private Score
Post-Deadline: Sat, 03 Dec 2016 10:28:07 Edit description	RedHat_RF_1203.csv	0.945261	0.944859
Post-Deadline: Sat, 03 Dec 2016 10:26:05 Edit description	RedHat_DT_1203.csv	0.873212	0.872404

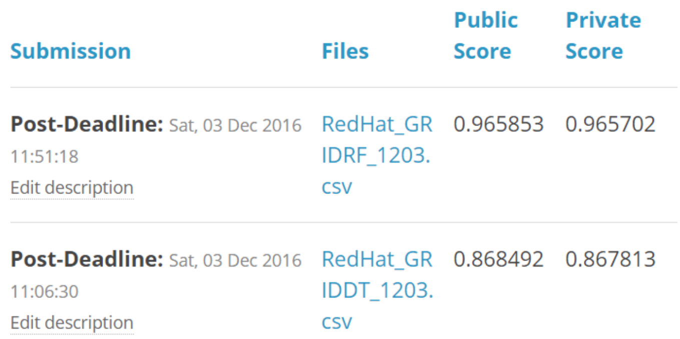
Fig 18. Comparison of Decision Tree and Random Forest

In terms of time costing, the training time of Random Forest is far longer the Decision Tree, and the predicting time distance is small. The time costing of Decision Tree is 22 seconds, and Random Forest is about 96 seconds. The predicting time costing is 2.79 and 3.08 seconds for 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. I would use the same parameter to tune the model, and don’t add the base learning model for Random Forest because of time costing. I choose the max depth, max features and criterion as the tuning parameter. The max depth is consists of 3, 4, 5. The max features are composed of log2 and sqrt. The criterion includes gini and entropy.

After tuning parameter, Decision Tree and Random Forest has a different result for the same parameter. The accuracy of Decision Tree decreases from 0.87 to 0.86. The accuracy of Random Forest increase from the 0.94 to 0.96. The detailed accuracy score is showed in the below.



Submission	Files	Public Score	Private Score
Post-Deadline: Sat, 03 Dec 2016 11:51:18 Edit description	RedHat_GRIDRF_1203.csv	0.965853	0.965702
Post-Deadline: Sat, 03 Dec 2016 11:06:30 Edit description	RedHat_GRIDDT_1203.csv	0.868492	0.867813

Fig 18. Comparison of Decision Tree and Random Forest after tuning

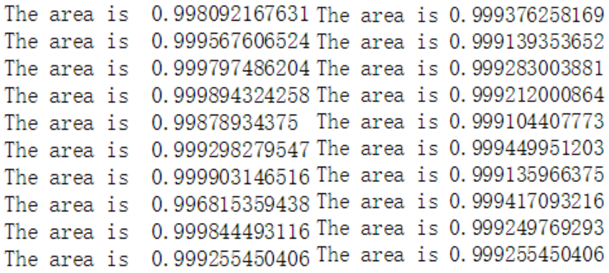
The upper one is Random Forest accuracy after using Grid Search, the below is Decision Tree. So, the default Random Forest has the best performance. But, the training time of Radom Forest used Grid Search is about 1224 seconds, Decision Tree's time costing is about 251.82 seconds.

So after tuning, I would use the parameter of Random Forest' best estimator as the final parameter. The parameter is that max features used log2, criterion used gini and max depth used 5.

4. Results

Model Evaluation and Validation

We give the bellowed graph to have a direct on the comparison of validation and test set before tuning. The average area both Random Forest and Decision Tree is over 0.99, but the test of Random Forest is about 0.96, and Decision Tree's area is about 0.87. We find that Random Forest's validation and test area is closely, and the Decision Tree is overfitting for the bigger distance of validation and test area.



```
The area is  0.998092167631 The area is 0.999376258169
The area is  0.999567606524 The area is 0.999139353652
The area is  0.999797486204 The area is 0.999283003881
The area is  0.999894324258 The area is 0.999212000864
The area is  0.99878934375  The area is 0.999104407773
The area is  0.999298279547 The area is 0.999449951203
The area is  0.999903146516 The area is 0.999135966375
The area is  0.996815359438 The area is 0.999417093216
The area is  0.999844493116 The area is 0.999249769293
The area is  0.999255450406 The area is 0.999255450406
```

Fig 19. Area of Validation set (Left: Decision Tree, Right: Random Forest)

After tuning, Random Forest still have a better performance than Decision Tree.

Justification

The average area of validation data for Random Forest is about 0.9949, and the area for Random Forest before tuning is about 0.9452, the area after tuning is 0.9658. The average area of validation data for Decision Tree t is about 0.9945, and the area for Random Forest before tuning is about 0.8732, the area after tuning is 0.8678.

Compared with benchmark, Random Forest is over than the area both validation and testing set. Only the validation set is meeting the benchmark, and the area of testing set is lower than benchmark for testing set. So I would choose Random Forest as the final algorithm to train model.

5. Conclusion

Free-Form Visualization

In this project, we use the area under ROC curve as the metric. So we can use ROC curve calculating the area in the curve to display the training accuracy changes. In fact, we also should give this curve for the testing set. However, the testing set is unlabeled, we don't know the real outcome, we just upload the result file to Kaggle and get the area under ROC curve.

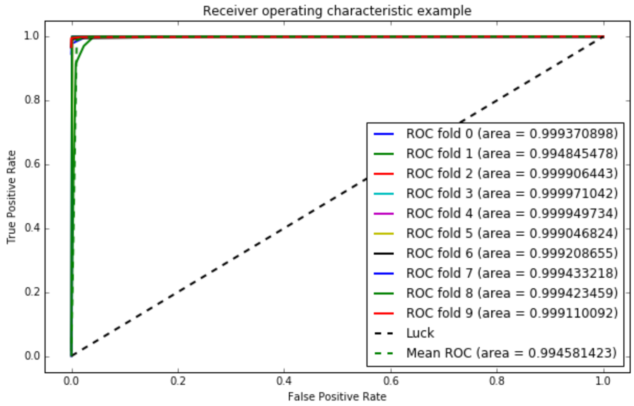


Fig 19. ROC curve of Decision Tree (10 folds)

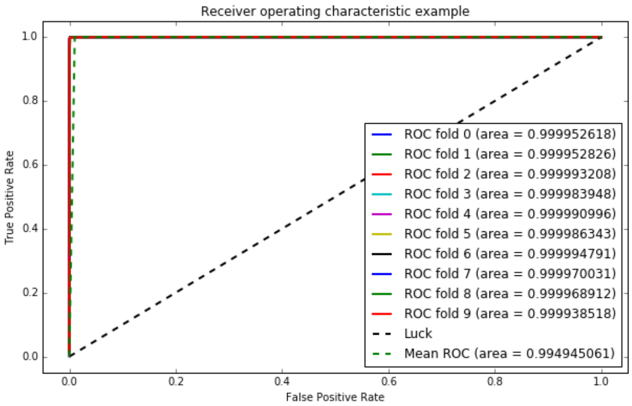


Fig 20. ROC curve of Random Forest (10 folds)

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 wrote some function to do data pre-processing and show the relation of feature and outcome. The 55 features are analyzed in divided way for people and activity data. I used Random Forest and Decision Tree to train the data and using area under ROC curve.

Then I used Grid Search to tune Decision Tree and Random Forest. The performance after tuning on two algorithm have an opposite result on activity test data.

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

Improvement

In fact, I use cross validation to make full use of training data to get a better performance. But in parameter tuning, I just try several combinations, there are more groups to examine.

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
[4] https://en.wikipedia.org/wiki/Random_forest