Data Analytics Research Assistant

DATA ANLYSIS TASK-REPORT

Fan Yang

University of Chicago, MScA

Content

Abstract

The task mainly contains about three sections: data cleansing, exploratory analysis and predictive models and the author spent 2.5 hours, 0.5 hour, 6.5 hours in analyzing and developing the code. This report summarized main procedures of the data cleansing part, important findings in the exploratory analysis and choices and performances of predictive models. The detailed code of these three sections can be found in the attached Jupyter Notebook.

Data Cleansing

Firstly, the data was loaded to OpenRefine software and examined. The following columns were groups and modified:

Race: There are numerous race category. Similar races are grouped together. For example:

1. 'American Indian or Alaska Native - Chippewa, Native' & 'American Indian or Alaska Native - Other Native American, Native';
2. All black, black or African American are grouped;
3. All Asians are grouped
4. Other, null value and I prefer not to respond are grouped as 'N.A.'

Hispanic: We can get clue of this from race. To fill the null value, we examine Hispanic together with race:

Race	Hispanic	Count_Hispanic
N.A.	null	10
	No	2
	Yes	5
White	null	2
	No	73
	Yes	11

For the null Hispanic with null race we leave it as it is. For the whites, since most of the whites are non-Hispanic, we filled the null value as No.

UG End Date: For the undergraduate end date, it is reasonable to ignore the specific date and only leave the year. This greatly reduces the number of categories in the columns which greatly facilitates further training of predictive model. Moreover, after discard the date, we are able to convert the data type to integer.

Post-Bac Work: There are many overlaps in the data, for example, '3-5 years', '4 years' and '5 years'. Therefore, 6 bins were created and the data place in each bin accordingly.
Bins: (0-1 years, 1-2 years, 2-3 years, 3-5 years, 5-7 years, 7+ years)

Academic, Leadership, GPA and Recommendations: The missing values were filled with the mode of each variable.

GRE-related data: The missing value of GRE verbal and quantitative percentile were filled with their group mean value. For the scores, both new and old GRE scoring system are included. The scores with old scoring system were mapping to the new scoring system and their percentiles remain unchanged.

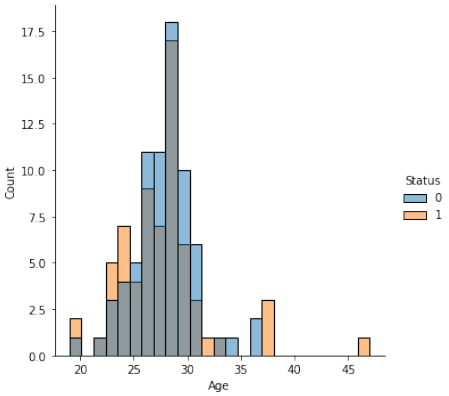
App-Financial-Amount Total: The data type was converted to integer to facilitate further analysis

College GPA Data: 0-100 GPA systems were mapped to 0-4.0 system.

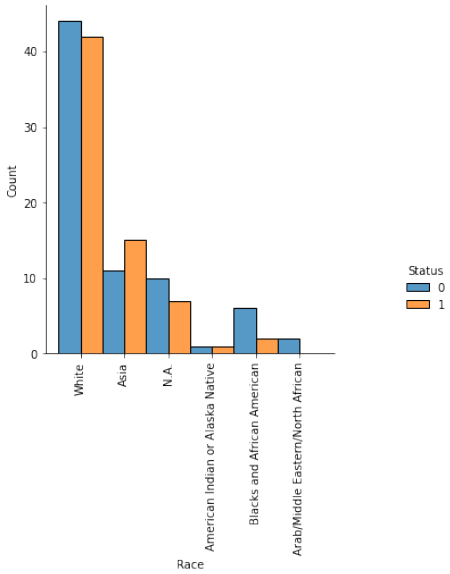
Exploratory Analysis

This section firstly examines the distribution of different variables grouped by the status and then discover the relationship among independent variables.

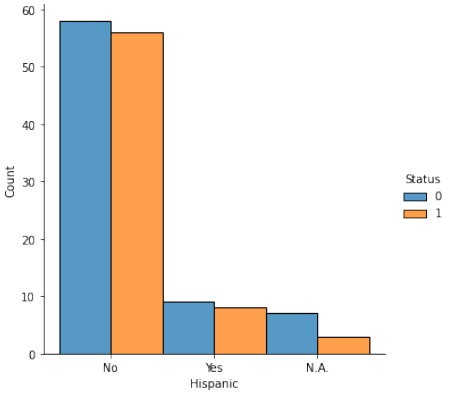
Age:



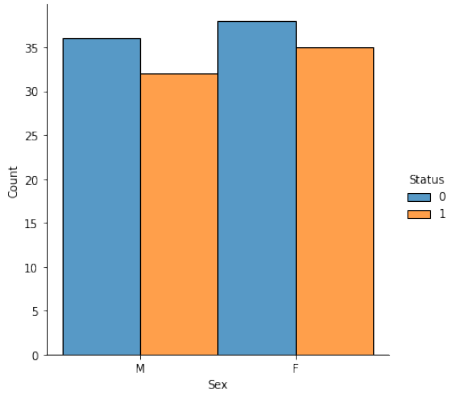
Race:



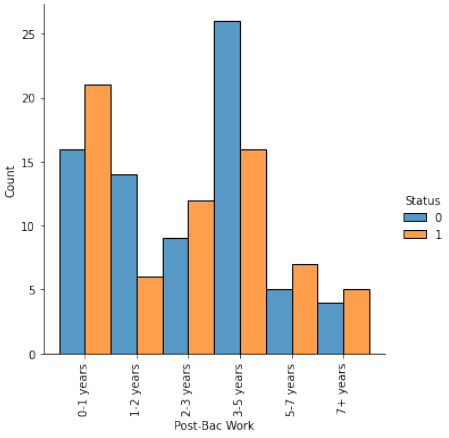
Hispanic:



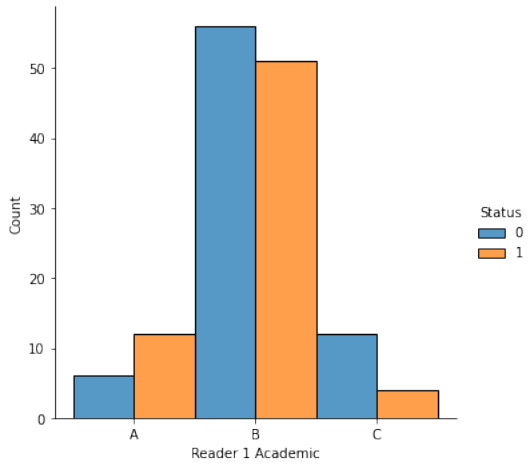
Sex:



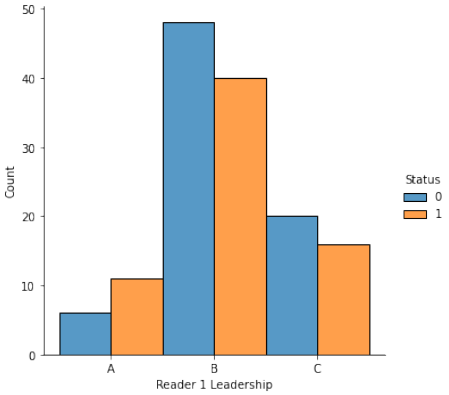
Post-Bac Work:



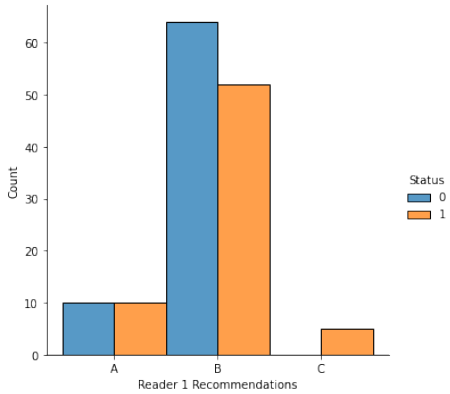
Reader 1 Academic:



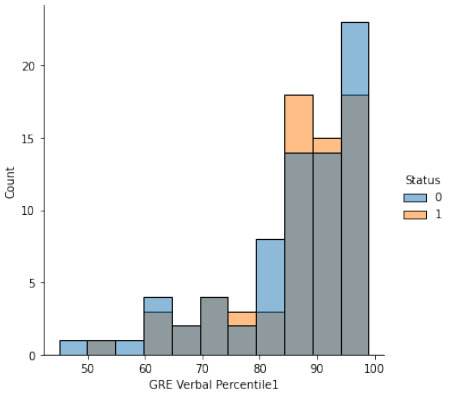
Reader 1 Leadership:



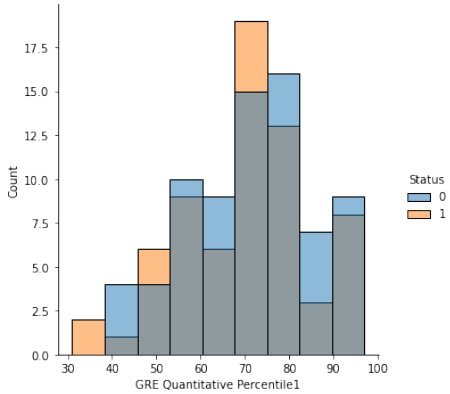
Reader 1 Recommendations:



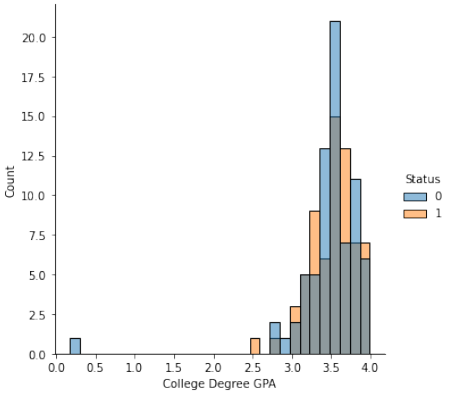
GRE Verbal Percentile1:



GRE Quantitative Percentile1:



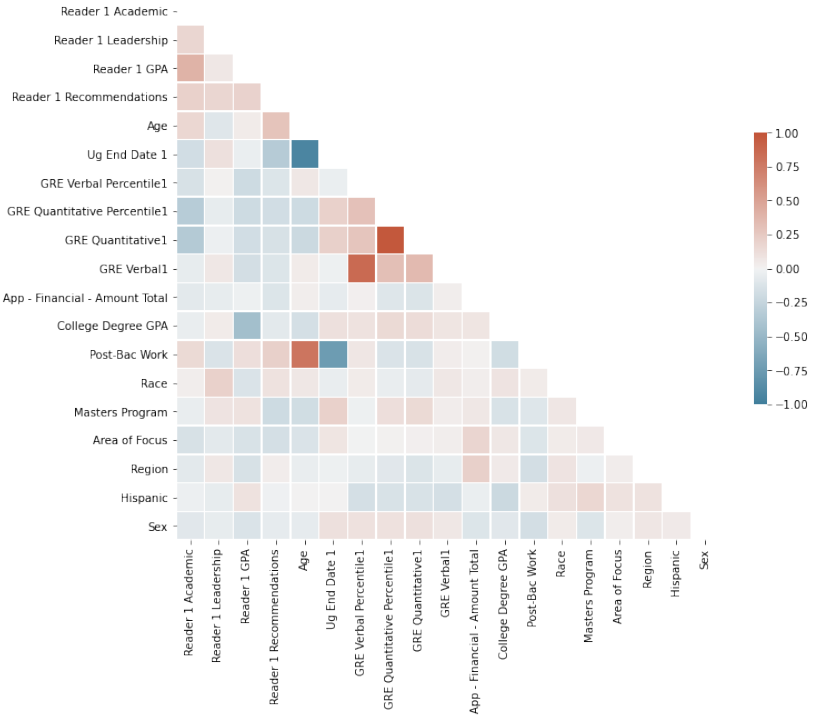
GPA:



As we can see from the data, few variable is able to distinguish status 0 and 1; therefore we foresee that the predictive model may be hard to get very good results.

Relationships between variables:

Pair plot between variables



Correlation between variables

As we can see that UG end date has a roughly positive relationship with age and use both of them can be unnecessary. However, since there are only 140 data points, the time saved in the model training by dropping one of them is almost negligible.
For GRE verbal - GRE verbal percentile, GRE quantitative – GRE quantitative data, the positive relationship is stronger, we will consider drop GRE verbal and quantitative score. Linear regression model was adopted and by getting the t-statistics of the model we can confirm that their correlations are statistically significant.

Correlation matrix between GRE quantitative and its percentile

1	0.978
0.978	1

OLS Regression Results

Dep. Variable:	y	R-squared:	0.957
Model:	OLS	Adj. R-squared:	0.957
Method:	Least Squares	F-statistic:	2783.
Date:	Tue, 17 Nov 2020	Prob (F-statistic):	8.34e-87
Time:	11:28:15	Log-Likelihood:	-185.82
No. Observations:	126	AIC:	375.6
Df Residuals:	124	BIC:	381.3
Df Model:	1		
Covariance Type:	nonrobust		

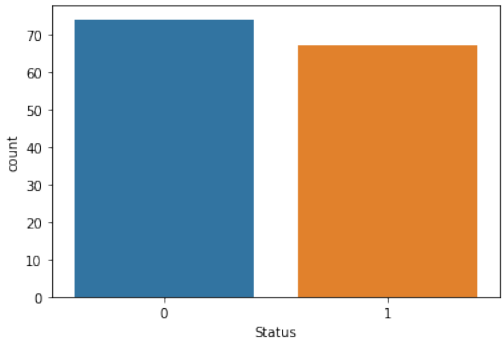
	coef	std err	t	P>|t|	[0.025	0.975]
const	136.1416	0.440	309.274	0.000	135.270	137.013
x1	0.3197	0.006	52.755	0.000	0.308	0.332

Omnibus:	23.657	Durbin-Watson:	2.002
Prob(Omnibus):	0.000	Jarque-Bera (JB):	88.310
Skew:	-0.522	Prob(JB):	6.67e-20
Kurtosis:	6.966	Cond. No.	337.

Linear model and its statistics between GRE quantitative and its percentile (GRE verbal is similar)

Predictive Model

In the predictive model, cleaned data was adopted and the main packages used for the predictive model including sklearn and xgboost. Since the importance of status 0 and 1 are consider the same, we will use accuracy to evaluate model performance instead of precision and recall. No new variable is created because linear combination between independent variable will not make a difference to the model performance.



Distribution of the label

Numerical variables were grouped by the label and averaged. The status cannot be distinguished by most of the variable means except 'App- Financial Amount Total'.

Different variables grouped by status

Status	Age	UG End Date	Verbal%	Quant%	Financial Amt	GPA
0	28	2011	86	72	1142	3.47
1	28	2011	87	70	2239	3.50

Summary

In the predictive analysis part, several classification models were adopted, and the results were compared to facilitate selection of the best model. Some classical models are used including logistic regression and decision tree from discriminative model group and Naïve Bayes for generative model group. Popular ensemble methods including random forest and Xgboost were also tried.

Due to the limitation of data points, all the models did not produce very good performances. However, there are indeed some models that out-perform the others. The accuracy of the best model in each machine learning models are summarised below. Among all the models, Xgboost gives the highest accuracy which is 0.667. From the results of all the model, it can be notice that all the models are not good at predicting status 0, this might indicates that the variables provided in the data do not possess a strong relationship with the status 0.

For the categorical data, label encoding was adopted instead of one-hot encoding as there are too few data points and one-hot encoding will significantly increase the number of columns which lead to large variance during the model fitting and will result in overfitting.

Even label encoding was adopted, the ratio of number of variable to the sample number is still high; therefore, regularization was adopted in every model to prevent overfitting. Moreover, principle component analysis (PCA) is also a potential choice. However, after adopting PCA

and use first 8 principle components (total explained variance>80%) as model input, the performance is not improved; thus, it is not included in the report.

Summary of all the models' accuracy

Model	Accuracy
Logistic Regression	0.624
Decision Tree	0.568
Random Forest	0.600
Xgboost	0.667
Naïve Bayes	0.632

Discriminative Model

Logistic Regression

Multiple logistic regression models were trained based on different ways of encoding categorical variables.

The accuracy of the model was compute using the average accuracy of K-fold cross validation with K=15. The accuracy of different models are summarized below and elaborated in detail after.

Summary of logistic regression accuracy

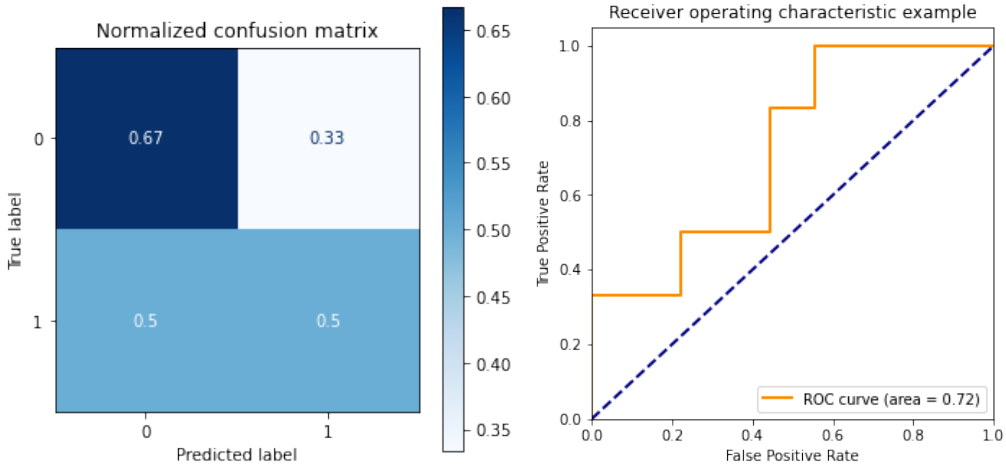
Model	Accuracy
Numeric variables	**0.477**
Numeric + ordinal categorical variables	**0.534**
Numeric + ordinal + nominal (one-hot) variables	**0.589**
Numeric + ordinal + nominal (labal encoding) variables	**0.52**
Numeric + ordinal + nominal (one-hot) variables + Tuning	**0.624**

1. Only use numerical values ('Age', 'Ug End Date 1', 'GRE Verbal Percentile1', 'GRE Quantitative Percentile1', 'App - Financial - Amount Total', 'College Degree GPA'):
   The mean accuracy on the test sets is **0.477** which even worse than random guess (around 0.528). Thus more variables need to be considered by the model.

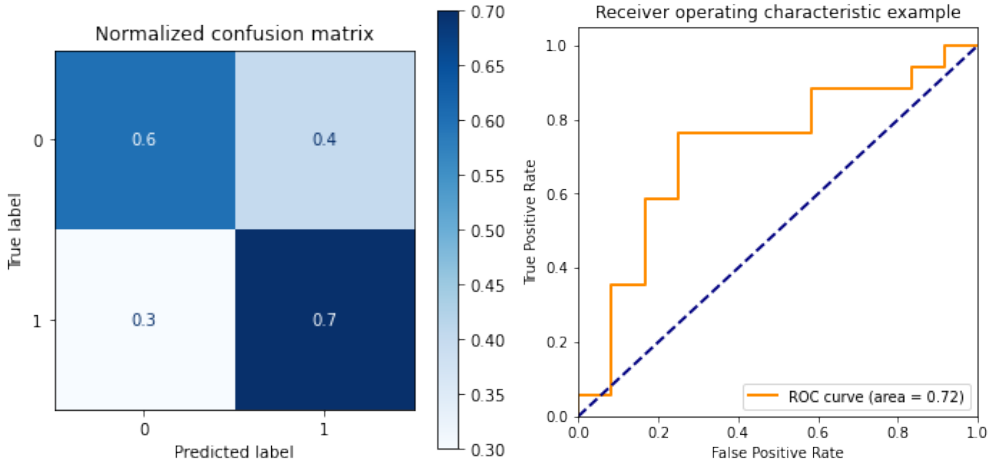
2. Include ordinal categories ('Reader 1 Academic', 'Reader 1 Leadership', 'Reader 1 GPA', 'Reader 1 Recommendations', 'Post-Bac Work'):
   At this time, I decide not to consider the nominal categories, such as regions, these categories have too many choices of values compared to the number of data points. I suspect including those variables may cause overfitting of the model.
   The codes regarding the rules of encoding the ordinal categories are shown below:

```
# Mapping
mapping1={'D':0,'C':1,'B':2,'A':3}
df['Reader 1 Academic']=df['Reader 1 Academic'].map(mapping1)
df['Reader 1 Leadership']=df['Reader 1 Leadership'].map(mapping1)
df['Reader 1 GPA']=df['Reader 1 GPA'].map(mapping1)
df['Reader 1 Recommendations']=df['Reader 1 Recommendations'].map(mapping1)
mapping2={'0-1 years': 0,'1-2 years': 1,'2-3 years': 2,'3-5 years': 3,'5-7 years': 4,'7+ years': 5}
df['Post-Bac Work']=df['Post-Bac Work'].map(mapping2)
df[['Post-Bac Work']].astype(int)
```

   The average accuracy using the same K-fold cross validation as previous model is **0.534** which is better than the previous model. The confusion matrix and ROC curve are shown below using test set size 10%. The results show that the model is bad at predicting data with status 1.



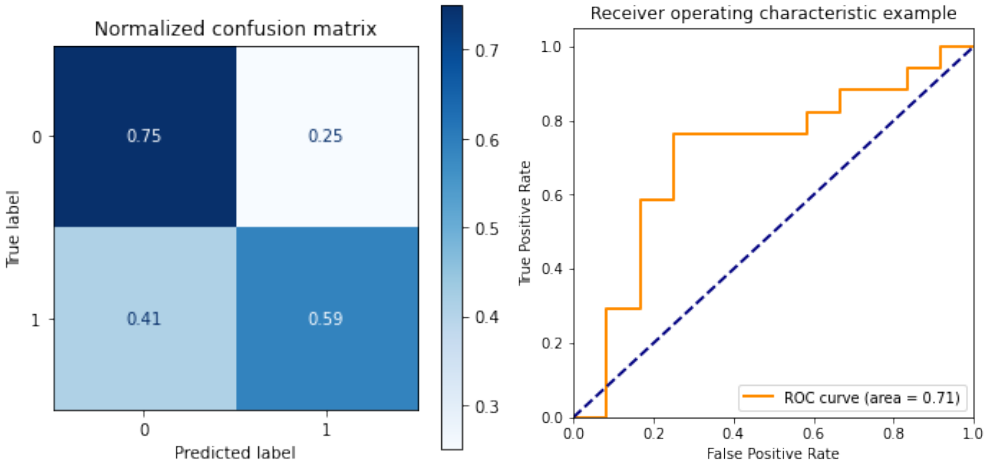
3. To check whether the model can be further improved, the nominal categorical variables were included in the model through different methods (one-hot encoding and label encoder):
   a) One-hot encoding: The nominal categorical variables are included in the model by apply one-hot encoding to them. The mean accuracy of model is **0.589** which is further improved by more than 10% compared to model 2. The confusion matrix and ROC curves are shown below. As we can noticed that the AUC for both model 2 and 3 are the same. This may cause by the randomness in splitting train and test sample sets. Since the accuracy is the average value of the cross validation results, it may be more robust when comparing performance of different models.



   b) Label encoder: In this method, the nominal categorical variables are encoded using the LabelEncoder function in sklearn package. And the model accuracy is **0.52** which is lower compare to the model using one-hot encoding method. This is reasonable as the label encoder model mapping different categorical values to numeric value which has ordinal property even though they do not have in their original format. This can negatively impact the model accuracy.

After testing different logistic regression, the best model was chosen (model with one-hot encoding), and its parameters were fine tuned. A grid search was conducted to get the best combination of parameters including solvers, balanced weight (for data with unbalanced label)

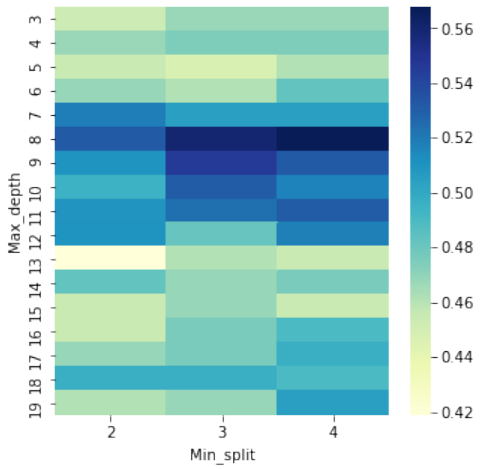
and factor for the regularization. The best logistic regression model has an accuracy of **0.624** and its confusion matrix and RUC curve are shown below.



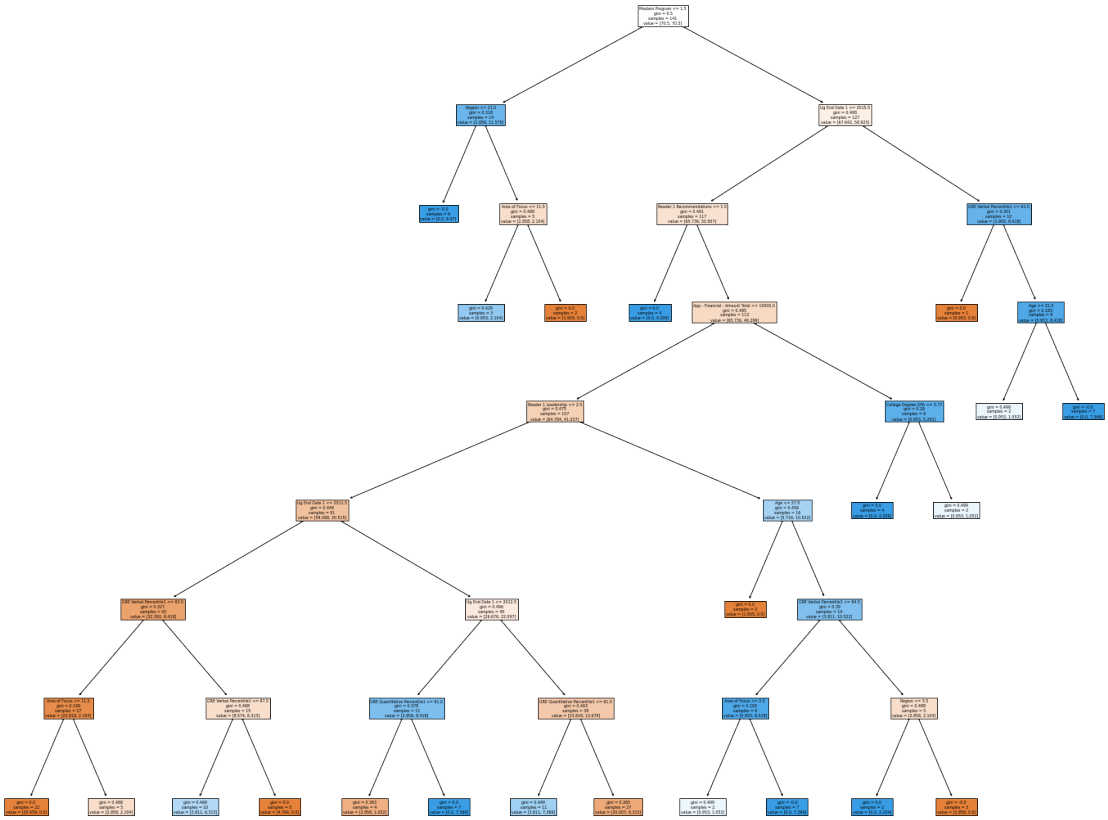
Decision Tree

Decision tree is a good candidate for the classification problem with multiple categorical variables. Thus decision tree was adopted as a candidate predictive model. A basic decision tree was adopted with tuned parameters for better model performance. Then pruning are adopted to prevent overfitting of the data. The categorical data was incorporated in the model through label encoding. One-hot encoding was not adopted due to the number of columns is excessive compared to the number of data which can lead to bad test accuracy.

Firstly, the maximum depth for the decision tree was selected based on performance. The accuracy of the model with different maximum depth and minimum split (minimum number of sample at a node that are qualified to split) are shown below. It can be noticed that depth 8 with minimum split sample number 4 has relatively high accuracy. Maximum tree depth was chosen as 8 to keep the model simple and prevent overfitting for complex tree. The accuracy of the basic decision tree is **0.568**. As the maximum decision tree depth is only 8, we refrain from pruning the tree since the tree is already considered as simple and shallow. The decision tree is visualized, the corresponding confusion matrix and ROC curve is also shown below.



Accuracy versus maximum tree depth and minimum split sample number



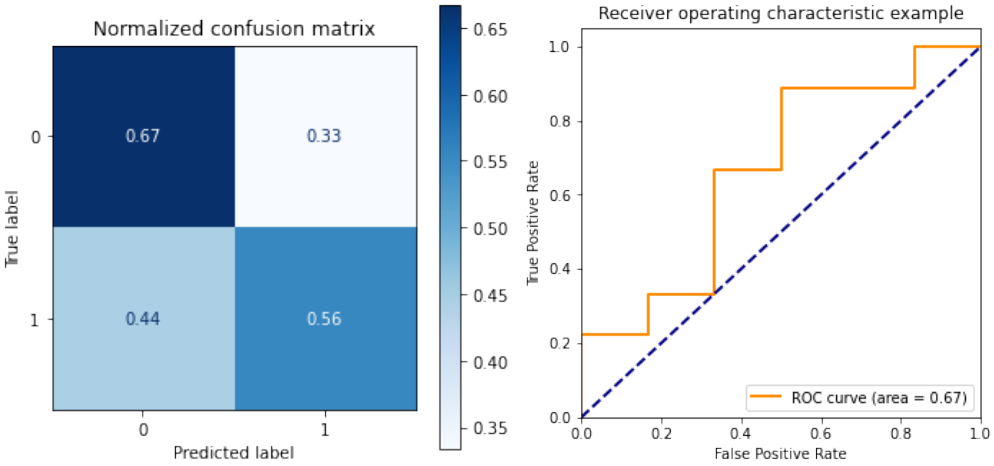
Decision tree with depth 8, minimum split sample number 4

Ensemble methods

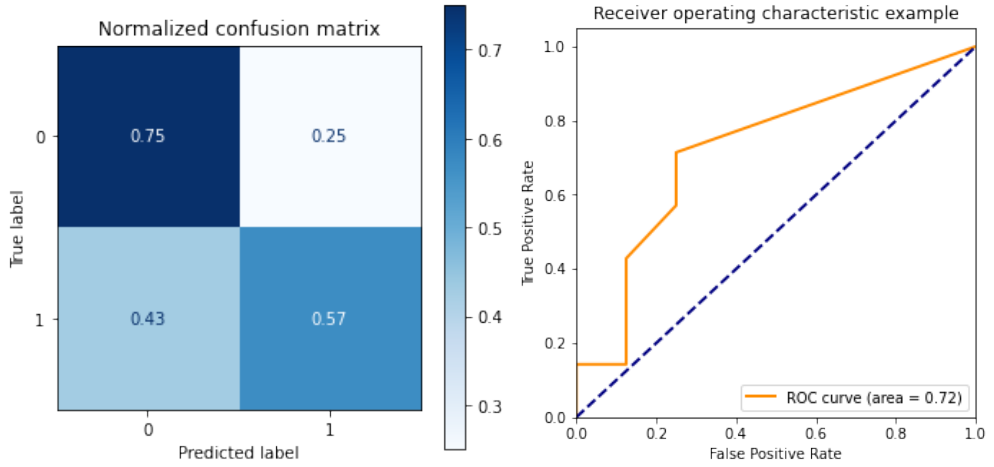
Random forest and Xgboost model were adopted to fit the data. Table and figures below provides a brief summary of the model performance and tuned parameters. The detailed information can refer to the code. From the figure showing the feature importance, it can be noticed that 'App-Financial Amount Total' is the model important feature in the classification. This agree with the previous finding that the mean 'App-Financial Amount Total' value varies the most in two different status. Different from other models, the accuracies were computed based on ratio of train:test = 9:1.

Model summary

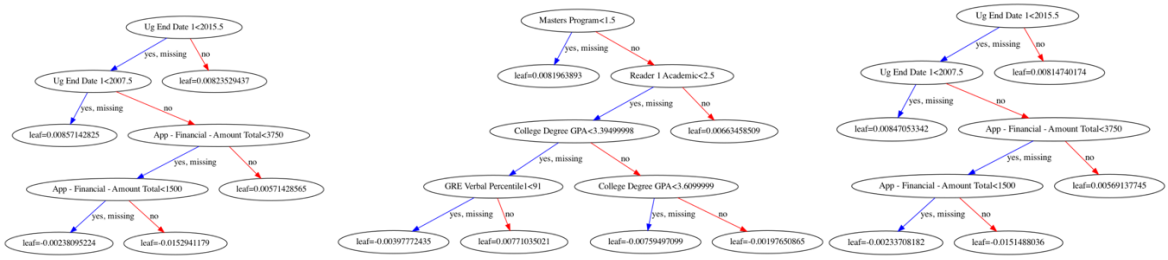
Model	Parameter settings	Accuracy
Random forest	n_estimators=60; min_sample_split=2; class_weight='balanced'; max_samples=90	**0.6**
Xgboost	Learnin_rate=0.01; n_estimators=5; max_depth=4; min_child_weight=2; objective='binary:logistic'	**0.667**



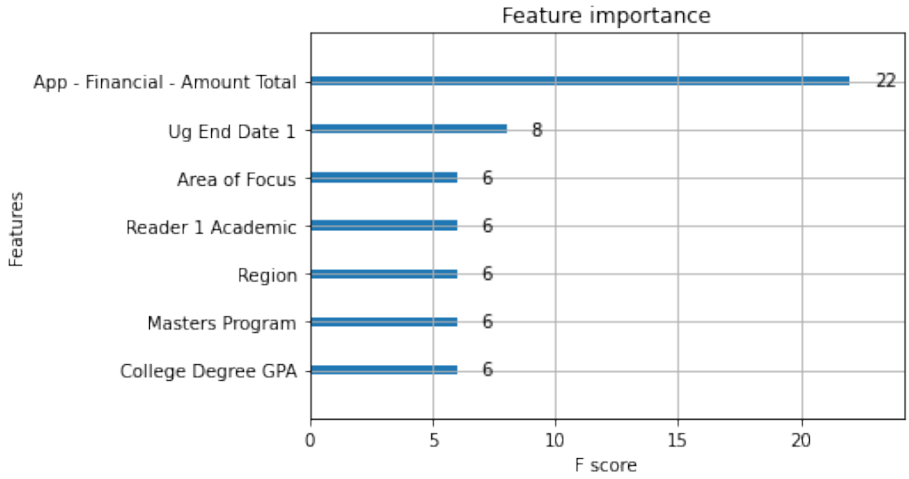
Random Forest



Xgboost



Xgboost tree visualizations: 1st, 2nd, 3rd tree



Feature importance ranking of Xgboost model.

Generative Model

Naïve Bayes

The last method applied is Naïve Bayes. Different from previous models which are discriminative models, Naïve Bayes is a generative models and we assume that all the variables are independent from each other which is a strong assumption. During the exploratory analysis, we found that 'Age' and 'UG End Date' have a strong correlation; therefore, columns 'Age' was drop in this model. The confusion matrix and ROC curve are shown in the following figures and the average accuracy computed from k-fold cross validation is **0.632**.