ASSIGNMENT 8
SUPERVISED LEARNING

PART 1

5420 Anomaly Detection, Fall 2020

- Harsh Dhanuka, hd2457

Agenda

LOAN DEFAULT

Loan default occurs when a borrower fails to pay back a debt according to the initial arrangement.

Predict which loan holders will likely default

0 = No Default 1 = Default

Which techniques to use? Which Random Forest package to use? How to convey the results?

Around $20 billion defaults only in student loans in US

Random Forest

Random Forest is a type of Bagging Model. The random forest method builds many decision trees, and then takes the average for the outcomes of all the decision trees. Further, the random forest technique draws some samples to build a model, then draws some samples again to build another model, and so on.

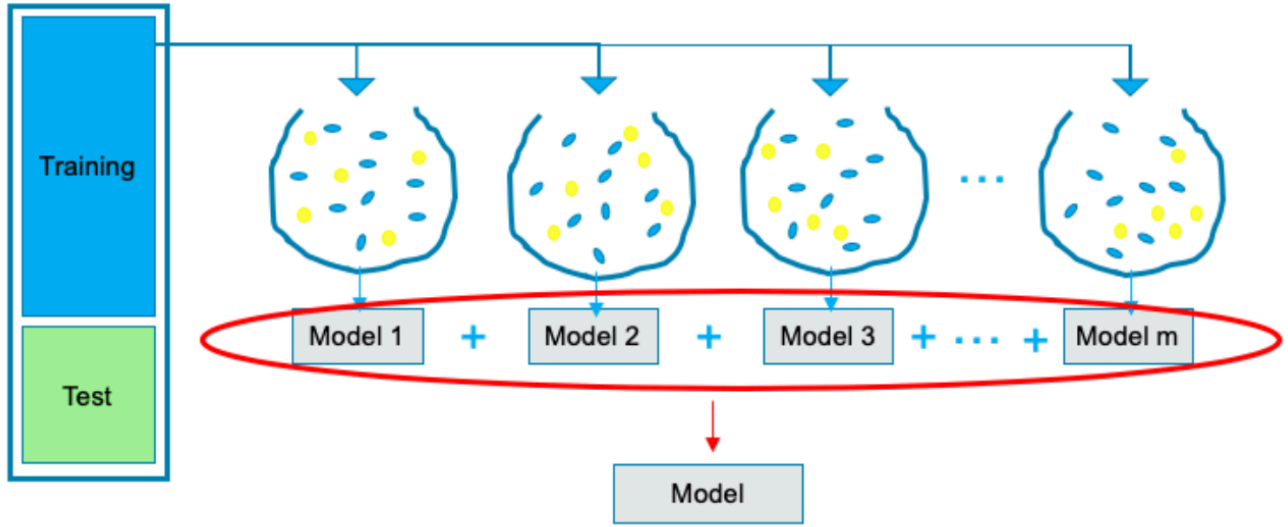


Figure (II): Random Forest — a Bagging Method

H2O

H2O is a fully open source, distributed in-memory machine learning platform with linear scalability. H2O supports the most widely used statistical & machine learning algorithms including gradient boosted machines, generalized linear models, deep learning and more. H2O also has an industry leading AutoML functionality that automatically runs through all the algorithms and their hyperparameters to produce a leaderboard of the best models.

Considerations

1. Feature Engineering

- Convert Date/Time columns
- Education Code, Gender, City converted to categorical type
- Drop empty value variables and constants
- Bin the Loan Amount column

2. Missing values

- Convert mis-represented NA's such as -99, to np.nan
- Use Iterative Imputer to fill with median

3. Dummy coding of Categorical variables not needed

4. Split to train_test

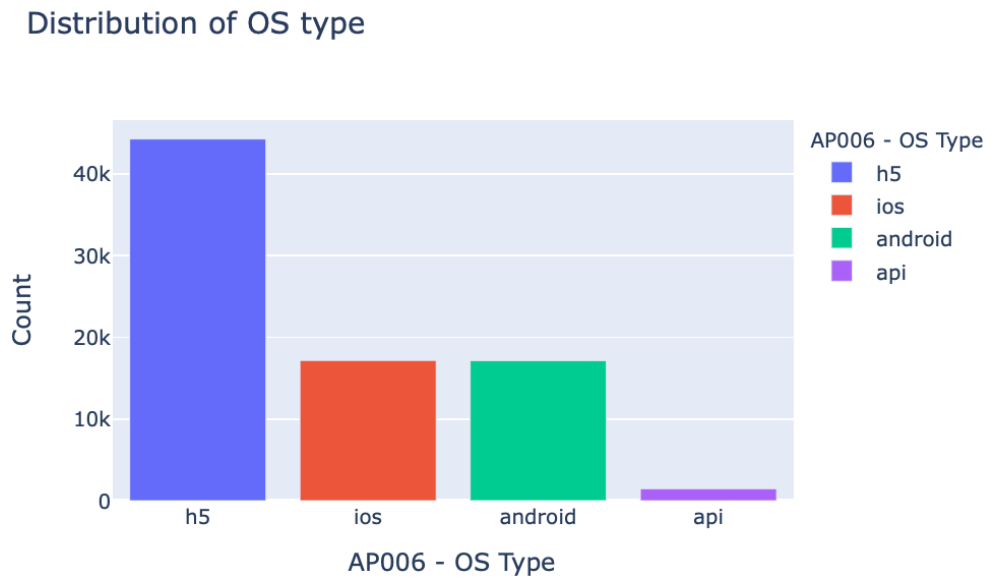
- I used 75% data in train set. Convert all dataframes to H2O hex format.

Loan Amount Bin ($)	Count
(-1, 100000]	74142
(100000, 200000]	4125
(200000, 300000]	975
(300000, 400000]	353
(400000, 500000]	166
(500000, 600000]	95
(600000, 700000]	48
(700000, 800000]	32
(1000000, 1500000]	31
(800000, 900000]	19
(900000, 1000000]	14

Name: CR009, dtype: int64

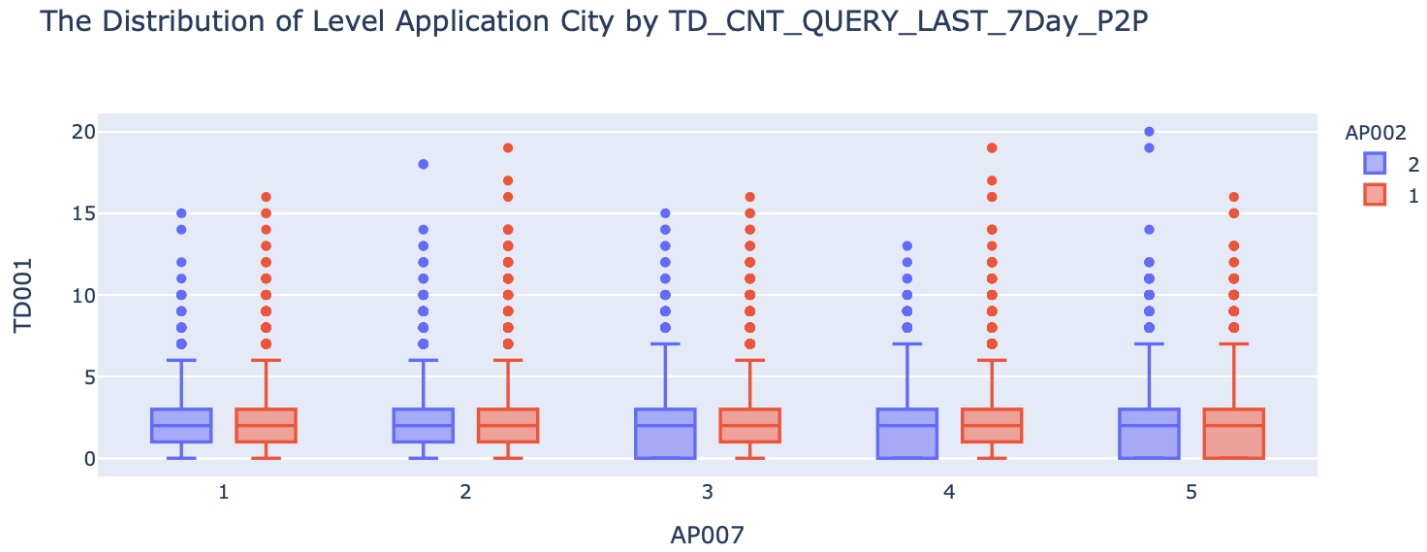
Exploratory Data Analysis

1 **Distribution of AP006 – OS Type**



2 **Distribution of Application City (AP007) by the Count of Query in 7 days**

Colored by Gender (AP002)



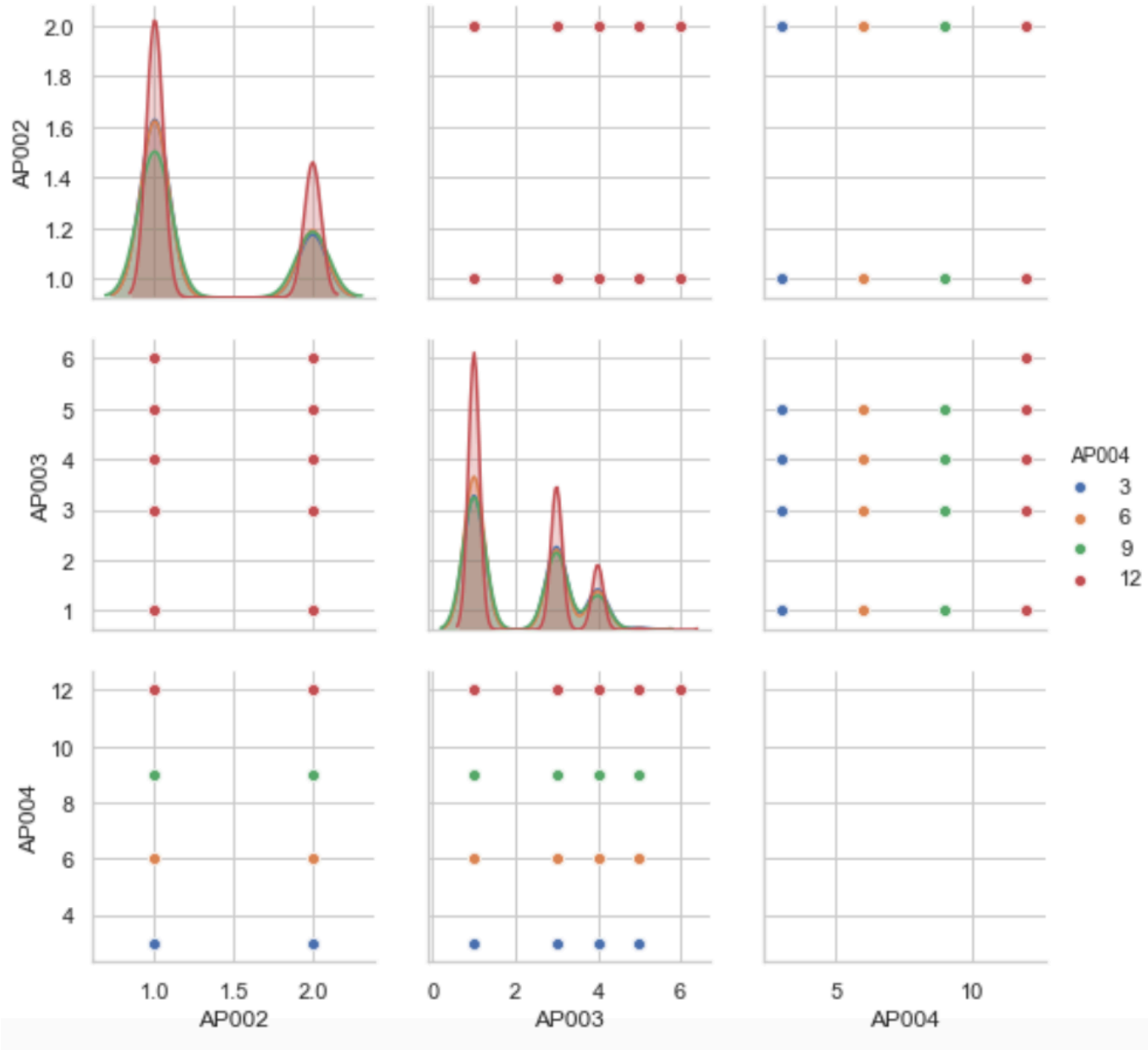
Exploratory Data Analysis

3 **Checking relations between:**

AP002 – Gender

AP003 – Education Code

AP004 – Loan Term



Random Forest Model

1 Model

Build 2 models:

1. balanced_classes = false
2. balanced_classes = true

Parameters:

ntrees = 400

nfolds = 10

min_rows = 100

balanced_classes parameter did not make any difference in results

19.3% data is minority class

2 Re-run the model after feature selection

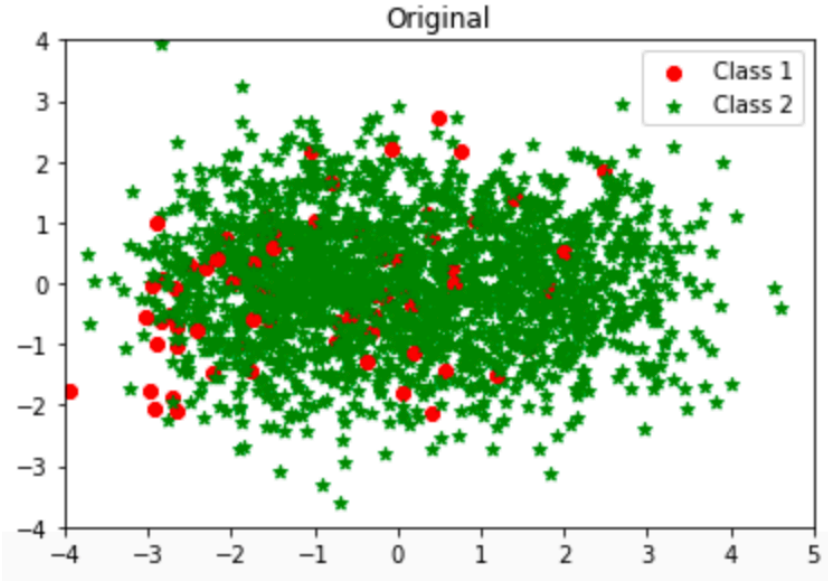
Out of the total 75 variables, my model gives the best results when I use the top 50 features

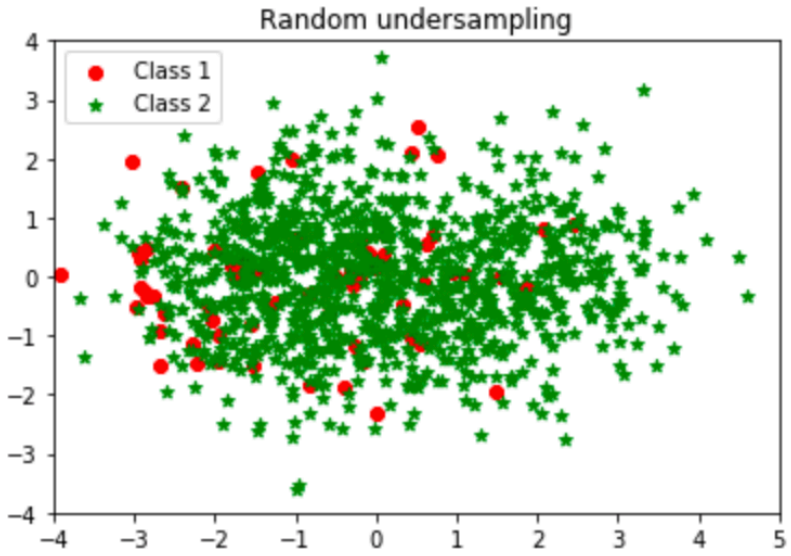
Variable Importance

	Variable	Relative_Importance	Scaled_Importance	Percentage
0	TD013	31743.115234	1.000000	0.131650
1	AP003	26869.242188	0.846459	0.111437
2	AP004	23823.884766	0.750521	0.098806
3	TD009	20168.527344	0.635367	0.083646
4	MB007	17379.031250	0.547490	0.072077
5	TD005	13576.405273	0.427696	0.056306
6	CR015	8044.451660	0.253424	0.033363
7	TD014	7797.213379	0.245635	0.032338
8	MB005	6884.897461	0.216894	0.028554
9	Loan_app_day_name	6594.753418	0.207754	0.027351

Random Under Sampling

Random Under Sampling is to under-sample the majority class randomly and uniformly. This can potentially lead to loss of information. But if the examples of the majority class are near to others, this method might yield good results.





Random Over Sampling

Random oversampling simply replicates randomly the minority class examples. Random oversampling is known to increase the likelihood of occurring overfitting. On the other hand, the major drawback of Random undersampling is that this method can discard useful data.

Random Forest Model

Discuss in detail the model which gave me the best 'Lift' and 'Precision Recall' score

1 **Manual Over Sampling**

I make both the classes in 1:1 ratio

So, 0's (no defaults) and 1's (defaults) both have 64512 entries each

Goal

- Get high Lift score
- Increase Precision Recall score

2 **Random Forest Model with manual Over Sampling**

Parameters:

ntrees = 400

nfolds = 10

min_rows = 100

balance_classes = False

3 **Predictions**

For predicting the loan default, I will not use a test set from the new oversampled/balanced data set.

I will use the originally split 25% dataset which was done before the 1st model.

Random Forest Model

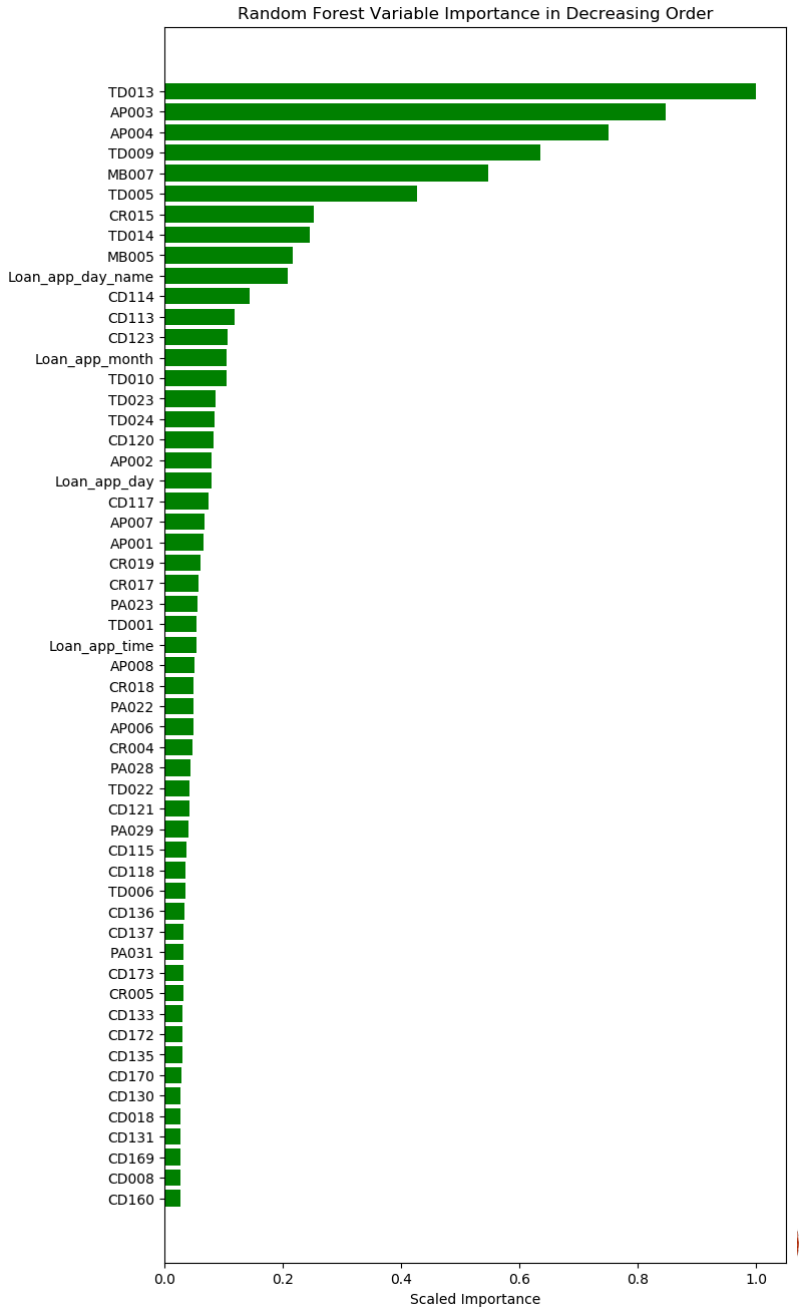
4 Feature Importance

TD013 – TD Count of Queries in last 6 months (P2P)

AP003 – Education Code

The 10 most important features are:

	Variable	Relative_Importance	Scaled_Importance	Percentage
0	TD013	31743.115234	1.000000	0.131650
1	AP003	26869.242188	0.846459	0.111437
2	AP004	23823.884766	0.750521	0.098806
3	TD009	20168.527344	0.635367	0.083646
4	MB007	17379.031250	0.547490	0.072077
5	TD005	13576.405273	0.427696	0.056306
6	CR015	8044.451660	0.253424	0.033363
7	TD014	7797.213379	0.245635	0.032338
8	MB005	6884.897461	0.216894	0.028554
9	Loan_app_day_name	6594.753418	0.207754	0.027351



Random Forest Model

5 Gains Table and Lift

Lift is a measure of the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model.

decile	count	actual	non_actual	cum_count	cum_actual	cum_non_actual	percent_cum_actual	percent_cum_non_actual	if_random	lift	K_S	gain
0	2000	1182	818	2000	1182	818	0.30	0.05	391.8	3.02	25.0	59.10
1	2000	717	1283	4000	1899	2101	0.48	0.13	783.6	2.42	35.0	47.48
2	2000	569	1431	6000	2468	3532	0.63	0.22	1175.4	2.10	41.0	41.13
3	2000	468	1532	8000	2936	5064	0.75	0.31	1567.2	1.87	44.0	36.70
4	2000	322	1678	10000	3258	6742	0.83	0.42	1959.0	1.66	41.0	32.58
5	2000	250	1750	12000	3508	8492	0.90	0.53	2350.8	1.49	37.0	29.23
6	2000	189	1811	14000	3697	10303	0.94	0.64	2742.6	1.35	30.0	26.41
7	2000	131	1869	16000	3828	12172	0.98	0.76	3134.4	1.22	22.0	23.92
8	2000	66	1934	18000	3894	14106	0.99	0.88	3526.2	1.10	11.0	21.63
9	2000	24	1976	20000	3918	16082	1.00	1.00	3918.0	1.00	0.0	19.59

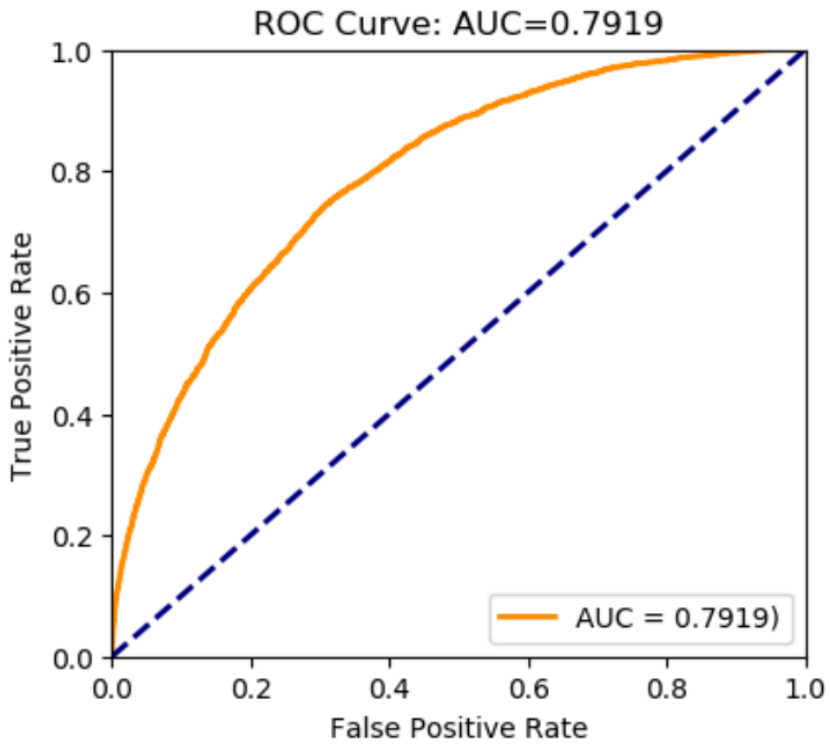
Random Forest Model

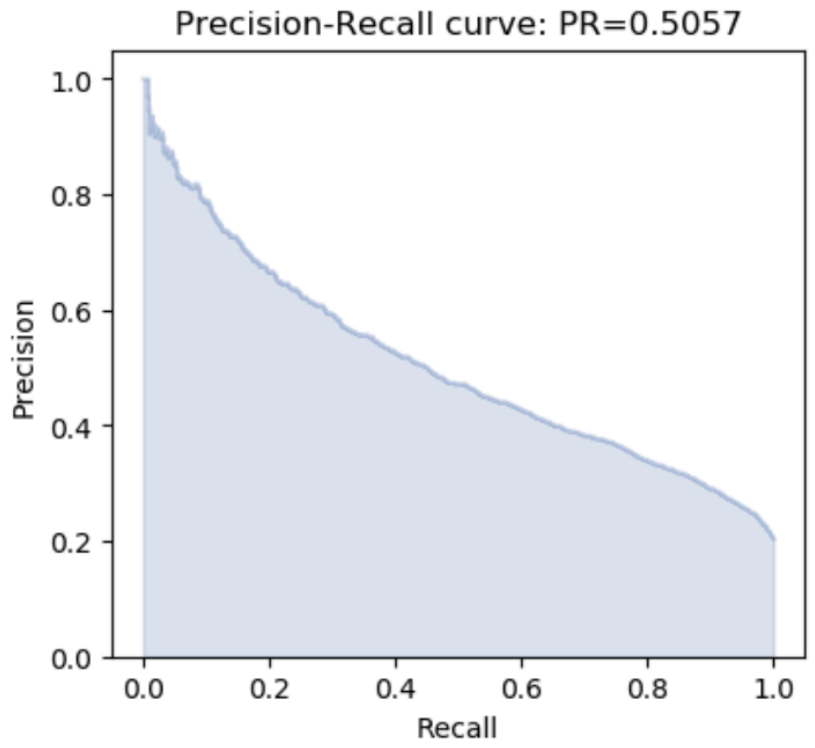
6 **AUC and Precision Recall**

ROC: The ROC curve plots the true positive rate vs. the false positive rate

AUC: A value between 0.5 (random) and 1.0 (perfect), measuring the prediction accuracy

Recall (R) = The number of true positives / (the number of true positives + the number of false negatives)





Business Insight

- **H2O package, Random Forest Model** is a very effective and efficient tool to build a machine learning model for predicting loan defaults. Also, H2O package is very handy to display the variable importance, handle correlations, and also dummy code the categorical variables.

- **Lift:** For the final model I built after tuning all the different models on various different values of each parameter, the highest Lift score I obtained is 3.02, which is very good as per industry standards. A Lift score of above 2 is suitable for the model to be of acceptable standards.

- **AUC** is 0.79 and **PR** score is 0.50 which is of acceptable standards. So the model is stable and fine.

Random Forest through the H2O package is a good approach to predict loan default.

However, we should not undermine other good boosting models such as gbm, xgboost, or Auto-ML and others. These might provide better results as well.