AARSH TODI (Aug-Sept)

TECHNOCOLABS MACHINE LEARNING INTERNSHIP

PROJECT REPORT

Problem Statement:

We had a dataset from one of the credit card company, including approximately 30,000 account holders. The dataset's response variable was whether a particular account defaults or not after a period of time. Our job was to help company develop a model where they know which accounts are expected to default and can take actions accordingly.

Dataset:

The dataset given to us had 30,000 rows and 25 columns. We were also told that the data is collected over a period of last 6 months. The response variable for the data is '**default payment next month**'.

Original format of the dataset: XLSX

A brief explanation of every column in the dataset is as follows:

ID: The account ID column.

LIMIT_BAL: Amount of the credit provided (in New Taiwanese (NT) dollar) including individual consumer credit and the family (supplementary) credit.

SEX: Gender (1 = male; 2 = female).

EDUCATION: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).

MARRIAGE: Marital status (1 = married; 2 = single; 3 = others).

AGE: Age (year).

PAY_1–PAY_6: A record of past payments. Past monthly payments, recorded from April to September, are stored in these columns.

PAY_1 represents the repayment status in September; PAY_2 = repayment status in August; and so on up to PAY_6, which represents the repayment status in April.

The measurement scale for the repayment status is as follows: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; and so on up to 8 = payment delay for eight months; 9 = payment delay for nine months and above.

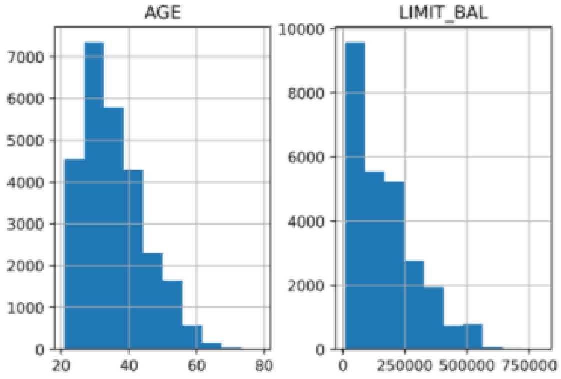
BILL_AMT1–BILL_AMT6: Bill statement amount (in NT dollar).

BILL_AMT1 represents the bill statement amount in September; BILL_AMT2 represents the bill statement amount in August; and so on up to BILL_AMT7, which represents the bill statement amount in April.

PAY_AMT1–PAY_AMT6: Amount of previous payment (NT dollar). PAY_AMT1 represents the amount paid in September; PAY_AMT2 represents the amount paid in August; and so on up to PAY_AMT6, which represents the amount paid in April.

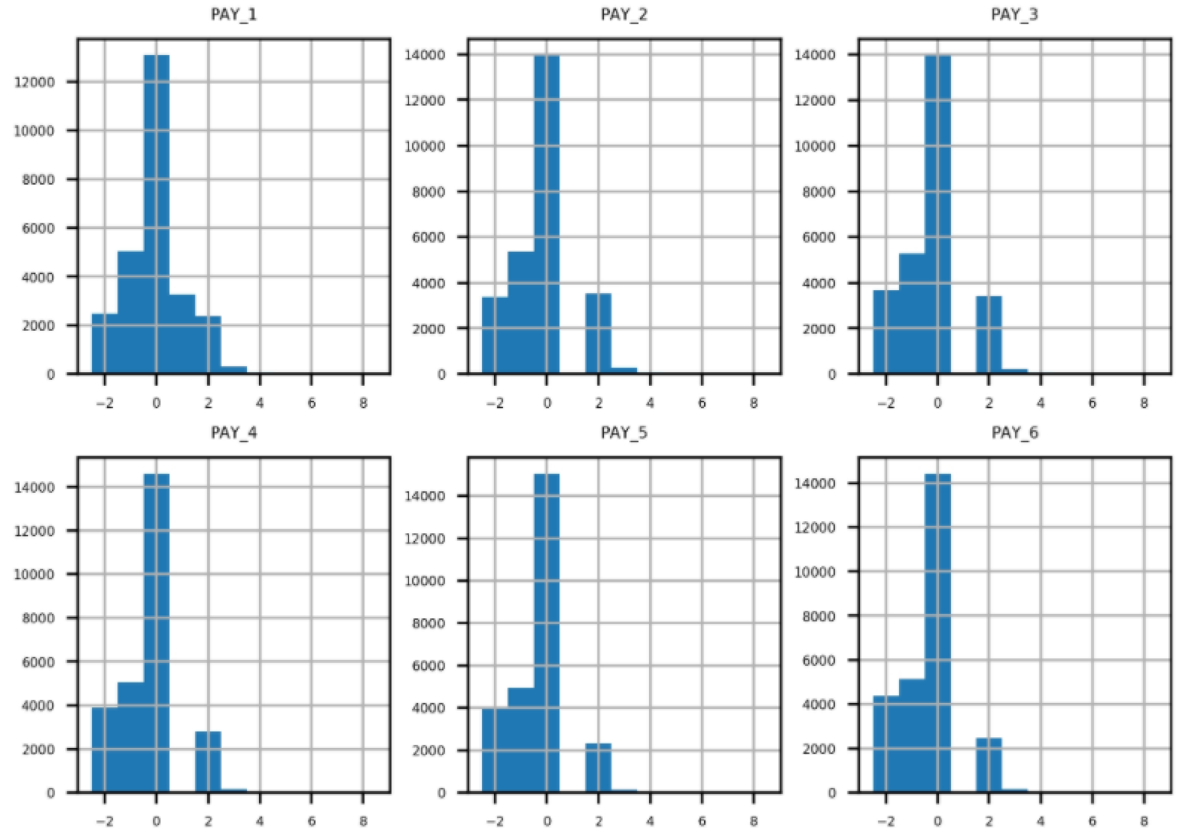
Data Cleaning:

This step of the model making involves Exploring the data and then cleaning the data. Here we first checked for the basic integrity of the data i.e. checking for uniqueness of accounts as we know having multiple rows with same ID does not make sense. Once we were done exploring the data we went for data cleaning with involved removing these multiple rows found in the data and then also checking for Null columns. We further checked the categorical values in columns to make sure we can map the values to the information provided to us by the customer. We found some irrelevant values in Marriage and Education column and we found it fit to put these values in others category. After this we plotted histograms for nearly all the columns as a part of data exploration.



Data Preprocessing and Preparation:

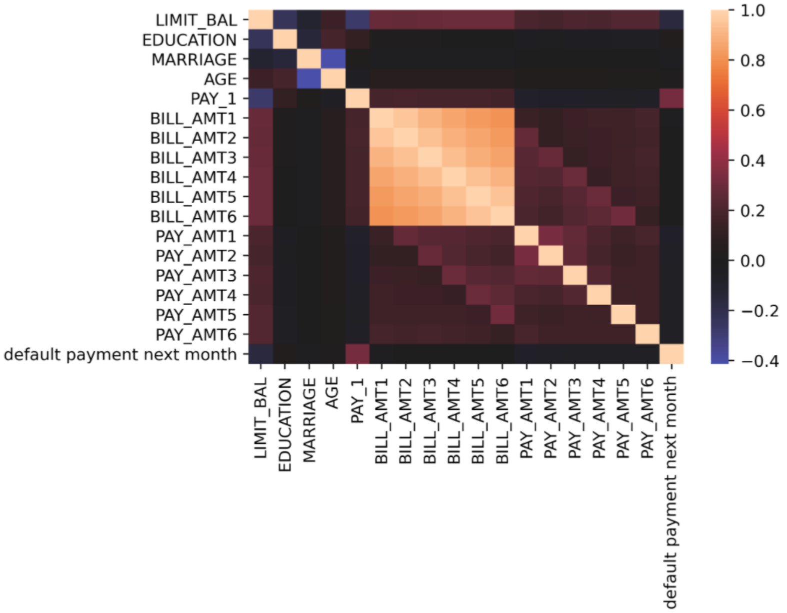
In this step we deep dive into the data and try to find answers to question “Do the given column make sense, should it be considered in model making. In our project we dived deeper into Pay, Bill and Bill amount columns. Established the creditability of each column. We find inconsistencies in the PAY column and had to contact client regarding the findings, which clarified that there was a glitch in data collection and thereby we decided to not to include PAY2-PAY6 columns. We were able to come to this conclusion with the help of some logical reasoning. Below are the histograms that show the inconsistency in data.



In July (PAY_3), there are very few accounts that had a 1-month payment delay; this bar is not really visible in the histogram. However, in August (PAY_2), there are suddenly thousands of accounts with a 2-month payment delay. This does not make sense: the number of accounts with a 2-month delay in a given month should be less than or equal to the number of accounts with a 1-month delay in the previous month.

Not only this it was also observed that some of the values of PAY_1 column were missing and thereby we had to go for imputation as the number of rows missing are significant and can affect our results. We tried different strategies and found that mode strategy works better in case of our project.

Correlation between Features, Correlation with response variable how strongly the change in related feature is going to affect the response variable, these are the good variables for our models. If two predictors are strongly correlated then we only need to use one of them. Features that are not all correlated to the response variable can be trimmed. Correlation between features helps us in making our model simpler. The Pearson correlation in our project:



We can observe from the heat map above that PAY_1 is strongly correlated to response variable and BILL_AMTs are correlated strongly.

Model Selection:

Model Selection is the part of Machine Learning project where we decide the algorithm for our dataset. In case of this project we started off with Logistic Regression. Later we went for an even stronger method based on Decision Tree. The project uses Random Forest Classifier which is based on a number of decision trees and is an ensemble Learning technique. Random Forest Classifier is known to work better than Logistic Regression because Logistic Regression uses a linear decision boundary.

Hyper parameter Tuning:

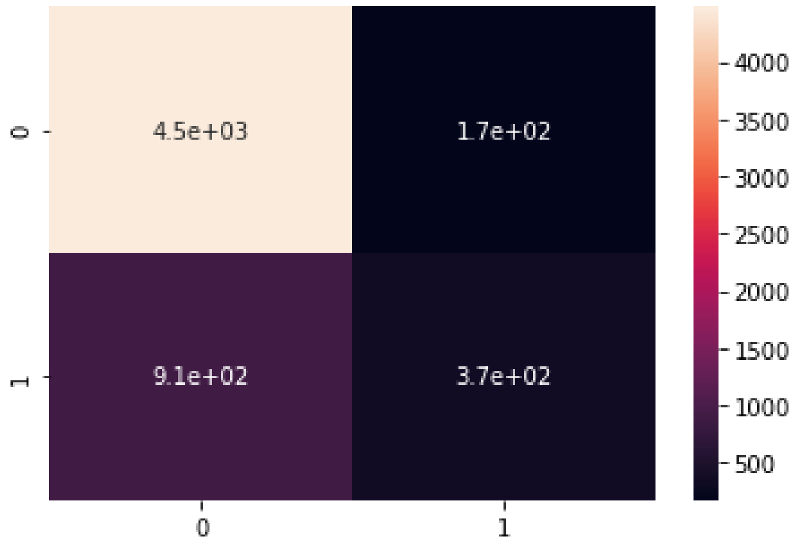
Once we decided that we are going to use Random Forest Classifier the very next question was “how many trees will be optimal? How many estimators we need?” these are the parameters that we do not train as a part of training process but select. These parameters affect models accuracy and computation time and hence it is very important to choose correct values. In this project GridSearchCV helped us determine these parameters. GridSearchCV helps us in running the model on all the possible combination of hyper parameters and then deciding the best one, in our case the best value was found to be 9 for n_estimators and 200 for number of decision trees.

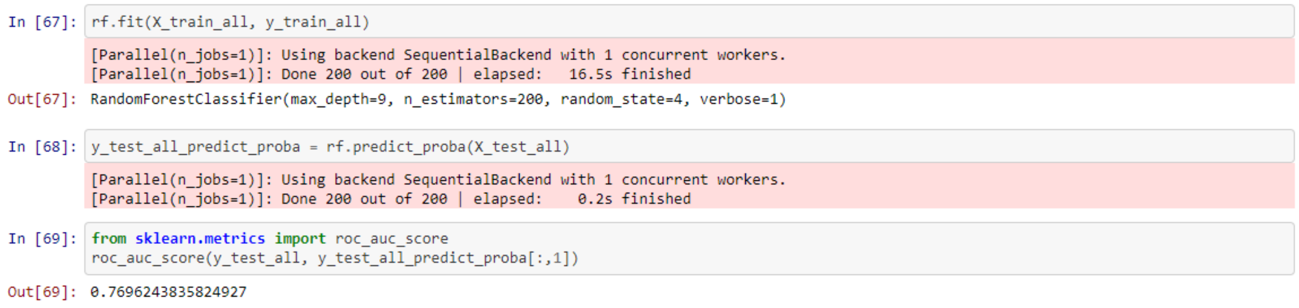
Model Accuracy:

The accuracy of the model was found to be **76.962%.**

The matrix below is the confusion matrix of the classifier we built.

```
Out[71]: <AxesSubplot:>
```





```
In [67]: rf.fit(X_train_all, y_train_all)

         [Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
         [Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:   16.5s finished
Out[67]: RandomForestClassifier(max_depth=9, n_estimators=200, random_state=4, verbose=1)

In [68]: y_test_all_predict_proba = rf.predict_proba(X_test_all)

         [Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
         [Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.2s finished

In [69]: from sklearn.metrics import roc_auc_score
         roc_auc_score(y_test_all, y_test_all_predict_proba[:,1])

Out[69]: 0.7696243835824927
```

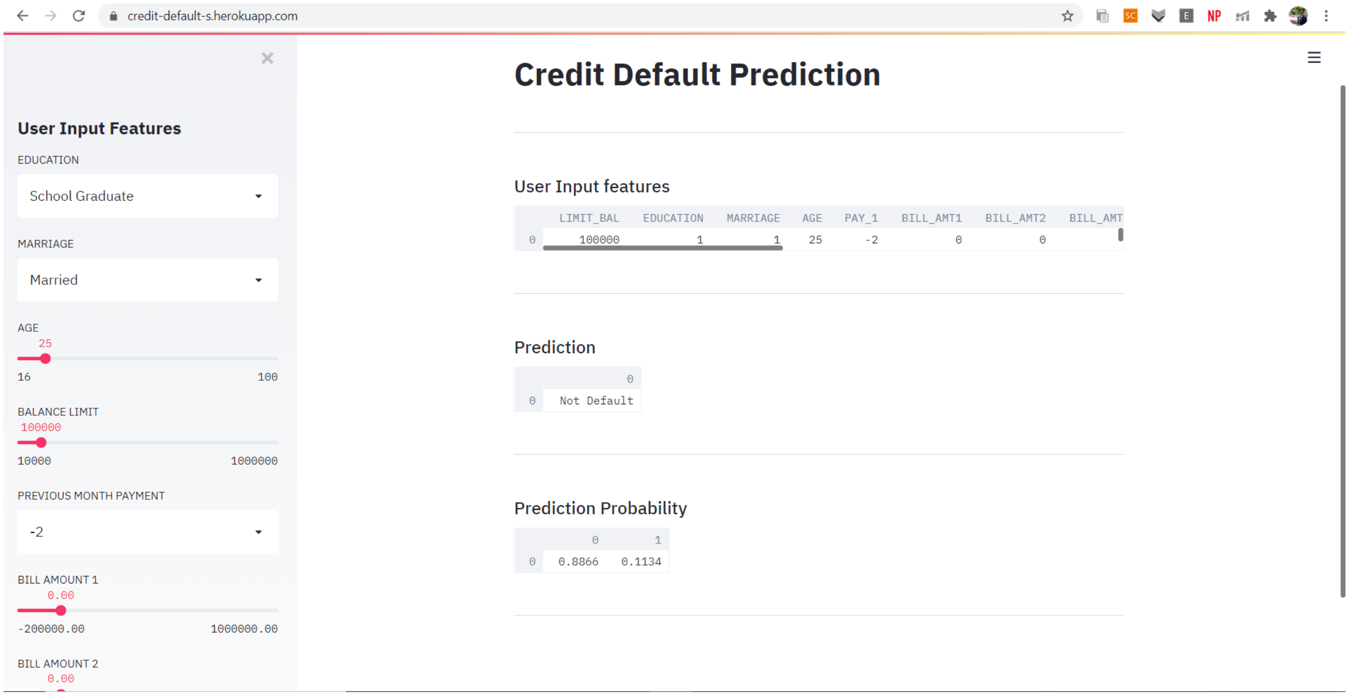
Deployment of the model:

The model built above was then saved as pickle file and used to deploy the model. The deployment as web application is done with the help of Streamlit library of python. The front end of the project is done with the help of Streamlit python library too.

The web application is easy to use; it not only helps our customer in determining the accounts that are expected to default but will also tell the customer which accounts should be considered for Credit Counseling.

Streamlit makes the front end very descent and easy to understand by providing scroll down options and sliders to input data. Application also prints the input values together in form of a row which can be used to recheck the entered values at a glance.

Cloud deployment of the model is done with the help of Heroku. Heroku is used as Platform as a service where we give our requirements and Heroku handles the server part of the application.



User Input Features

EDUCATION: School Graduate

MARRIAGE: Married

AGE: 25 (16 – 100)

BALANCE LIMIT: 100000 (10000 – 1000000)

PREVIOUS MONTH PAYMENT: -2

BILL AMOUNT 1: 0.00 (-200000.00 – 1000000.00)

BILL AMOUNT 2: 0.00

Credit Default Prediction

User Input features

	LIMIT_BAL	EDUCATION	MARRIAGE	AGE	PAY_1	BILL_AMT1	BILL_AMT2	BILL_AMT
0	100000	1	1	25	-2	0	0	

Prediction

	0
0	Not Default

Prediction Probability

	0	1
0	0.8866	0.1134

The link to Web Application: https://credit-default-s.herokuapp.com/