

DEPARTMENT OF ELECTRONIC AND TELECOMMUNICATION ENGINEERING
UNIVERSITY OF MORATUWA


TEAM - SLOWMOSQUITONET (DS1046)

Data Storm V1.0

H.U.D.B. HAPUTHANTHRI
M.A.M. AFHAM
K.K. HERATH

Best F1 Score: 0.82466

This is submitted as a partial fulfillment for DataStorm v1.0

—
[Github repo] 

1 Introduction

We were given a dataset to predict the customers who might default credit card payments. A default can occur when a borrower is unable to make timely payments, misses payments, or avoids/stops making payments. Banks need help in predicting and preventing credit card default to improve their bottom line. To solve this problem we used various feature engineering techniques and classifiers to end up with the optimum features and classifier for this classification tasks. Finally we will be presenting our business insights for this problem.

2 Exploratory Data Analysis

2.1 Basic Analysis of the distribution

As the first step we tried to get most of the insights from the given distribution by roughly plotting various graphs using libraries. Below are some of the plots from which we got a brief idea of which column or data is going to play a major role in this classification.

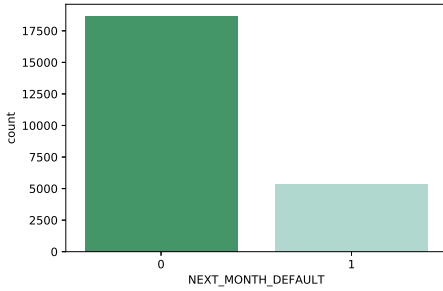


Figure 1: Target Counts

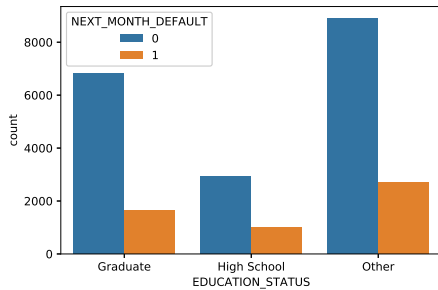
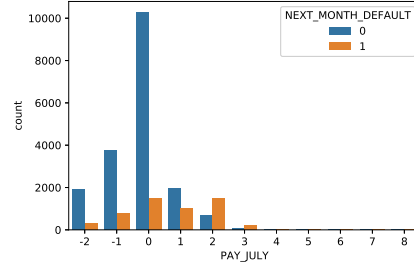


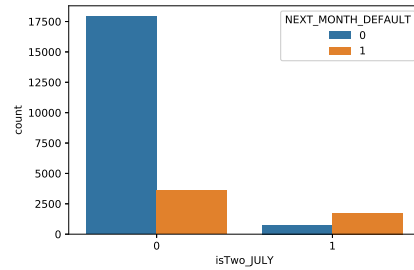
Figure 2: EDUCATION_STATUS

The first diagram shows the imbalance

nature of the data set which comprises of more 0's than 1's. The categorical variables such as 'AGE', 'EDUCATION_STATUS', 'MARITAL_STATUS' and 'Gender' shows a similar distribution of the second plot where not much insight is obtained.

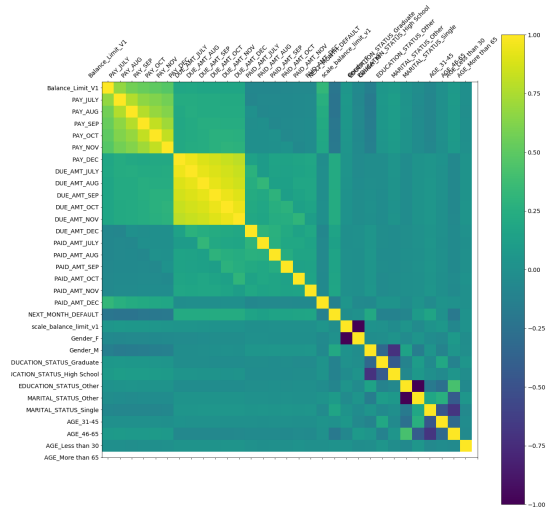


(a) PAY_JULY



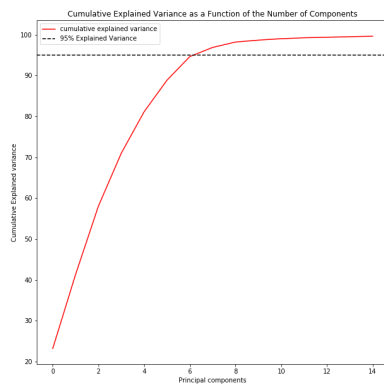
(b) PAY_JULY > 2

The column 'PAY_JULY' plays a major role since the probability of being credit card default is high when 'PAY_JULY' is greater than 2. (Identified from 2nd and 3rd plots)



(a) Correlation Matrix of given features

Figure 4: 2.2 section



(a) PCA- cumulative variance curve

Figure 5: 2.3 section

2.2 PCA-Cumulative Variance Curve(4)

By analyzing this curve, it can be observed that the most of the variance of the data will be due to the first 6 features of PCA (Principal Component Analysis). We used this prominent 6 features to train the models but there was not a significant improvement on the accuracy.

2.3 Correlation Matrix(5)

Correlation matrix can be used to analyze how the features are related to each other. By observing that it can be seen that DUE_AMT data and PAID data has some significant correlation.

2.4 Feature Importance Plots)

Further we approached to analyze how each feature is contributing towards the final model accuracy. We've used ExtraTreesClassifier in sci-kit learn library to get an idea of our feature engineering technique which we'll explain in the next section.

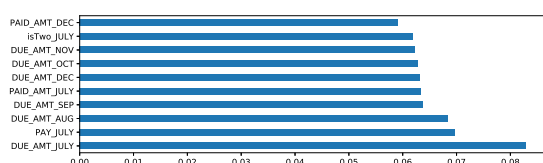


Figure 6: Feature Importance

3 Feature Engineering

3.1 Label Encoding & OneHot Encoding

Since from the Exploratory Data Analysis we saw that categorical features doesn't seem to have an impact in target variables, we implemented Simple Label Encoding Technique from Sci-Kit Learn to pass in the categorical features to the model. Once it was validated it gave us a lower accuracy than expected. So we switched to OneHot Encoding, the well-known technique to encode our categorical features.

Encoding Technique	Cross-Validated Accuracy for Random Forest Classifier
Label Encoding	0.813954
OneHot Encoding	0.8145

3.2 Feature Generation

The insights we got from Data Analysis made us generate new features which increased the F1 score considerably. considerably.

- PAY_values shows a significant correlation
- When the ratio between DUE values and PAID values increases the probability of an occurrence of a credit card default increases.

According to the above insights the following new feature columns were generated.

1. 'isTwo_JULY' - $\text{PAY_JULY} \geq 2 \implies \text{else } 0$
2. 'PAID_DUE_(month)' - The ratio between Due value and Paid value
3. 'PAY_TOT' - Sum of PAY_(month)

3.3 Scaling

To overcome the performance of the model due to data presented in various scales, we performed scaling techniques such as Min-MaxScaler and StandardScaler from Sci-kit Learn which helped us improve the validation accuracy.

3.4 Other

3.4.1 Oversampling

Due to the imbalance of the data, We tried oversampling using imbalance learn python lib. It didnt give higher accuracy.

3.4.2 Generate features using PCA and AutoEncoders

Autoencoders and PCA generated features are used for feed models. This method also could not help for a significant increment of the models' accuracies.

4 Model Selection & Evaluation

The following table describes the algorithms we used while cross-validating the data and the maximum accuracies given by them after a process of hyper parameter tuning. (The detailed code have been included in the GitHub repository)

Machine Learning Algorithm	Kaggle Submissions	Cross-Validated Accuracy
Support Vector Machines	No Submission	0.77
Random Forest Classifier	Day 01 - 1	0.8154
Deep Neural Networks	No Submission	0.7781
Ada Boost Classifier(Best)	Day - 01 - 2 & 3	0.81867
XGBoost Classifier	Day - 02 - 1	0.8198
LightGBM Classifier	Day - 02 - 2	0.8237

5 Final/Best Model

After doing our data analysis and feature engineering part, we were left with choosing the best model to feed our processed data and to get the maximum out of it. Since from the PCA plot, we got the idea that distance based classifier won't work well in our case, we focused in a Tree based Classifier. There are two methods of ensembling Tree based classifiers:

1. Bagging - Random Forest Classifier
2. Boosting - Ada Boost Classifier, XGBoost Classifier and LightGBM Classifier

The Bagging method is designed to reduce the variance such that overfitting doesn't happen while Boosting method is designed to reduce the bias such that underfitting doesn't happen.

Our initial approach was using Random Forest Classifier to ensure that overfitting doesn't happen but gradually we moved to Boosting techniques with proper hyper parameter tuning and optimization using Grid-SearchCV and with the background mathematical knowledge of the algorithms. So our goal was achieved by using a proper **Boosting Algorithm (XGBoost)** to reduce the **Bias (Under-fitting)** and with **Hyper-Parameter Tuning** to reduce the **Variance (Over-fitting)**

6 Business Insights

It's highly important to do such an analysis from the data a bank has, since then they will have an idea on the rate of defaults and approximately how much money they will lose in the coming months. Then from the trends in credit card defaults in the previous months by their customers, they will be able to take necessary action on targeted clients who have the highest risk in credit card defaults. Such as the bank can attract them with allowances for credit card payments such that they wouldn't make a default next time. However, if the situation continues for the predictions of next three months, the bank should consider discontinuing the credit purchase temporarily or permanently.