**Credit Card Default Prediction using Machine Learning Techniques**

*Group 2:*

*Kevin Harmer, Chaithra K C*

**Table of Contents**

[**Abstract**](#_r2ouzsp2nsxq) **3**

[**1. Introduction**](#_tz48i5xdbg18) **4**

[**2. Data Overview**](#_d4yiinzcgjig) **4**

[2.1. Data Collection](#_mo7uq3btbmi2) 4

[2.2. Data Wrangling](#_je6dlj2ychwo) 5

[**3. Data Exploration**](#_lwempnfb1vwe) **5**

[3.1. Delinquency Distribution](#_b267qzkmwhhb) 5

[3.2. Factor Correlation](#_hu9r4i7z0gv7) 7

[3.3. Other Notable Observations](#_jz0rfphcnhu8) 8

[**4. Model Methodology**](#_d0fd27ln57je) **9**

[4.1. Handling Class Imbalance](#_k8sufvjiqkv8) 9

[4.2. Feature Engineering](#_wyfsnbar7bgz) 10

[4.3. Model Structuring](#_qs192n11ev9p) 11

[**5. Credit Card Payment Default Prediction Results**](#_rby4wuu8pyv5) **12**

[5.1. Logistic Regression (default params)](#_nyx9314zauh5) 12

[5.2. Logistic Regression (with Regularization)](#_4j0flwuvz3r) 12

[5.3. Random Forest](#_tcvtt5forkn7) 13

[5.4. Deep Learning](#_f700h6js7nlo) 14

[**6. Credit Card Default Analysis**](#_w18m0tjfd03t) **15**

[6.1. Logistic Regression](#_fxadcti8n183) 15

[6.2. Random Forest](#_olg87fgw1y50) 15

[6.3. Deep Learning](#_u4kev447et72) 16

[**7. Conclusion**](#_vs76m5mqa9qr) **17**

[**8. Appendix**](#_3ecegc6r4h05) **18**

[8.1. Data Dictionary](#_gl861z82v578) 18

[8.2. References](#_9o01xfvdhftz) 18

# Abstract

The project aims to use three classification machine learning models to conduct a quantitative analysis of credit card default risk. Despite the fact that the banking industry has incorporated machine learning and big data, the existing models are primarily dependent on credit scores. Relying too much on credit scores could result in banks missing out on key customers, such as recent immigrants with good repayment capacity but little to no credit history. This analysis is a machine learning application on default risk, but the predictor features do not include credit score or credit history. Due to the regulatory constraints that banks are facing, for example, The Fair Credit Reporting Act (FCRA), the algorithms used in this analysis are relatively simple and interpretable except for the deep learning model which is considered to be a black box.

This analysis used a Kaggle public dataset that consists of 30,000 credit card users and 3 machine learning models - Logistic Regression, Random Forest, and Deep Learning. There might be other classification models that could yield better performance, but due to the scope of this project, we did not cover other algorithms. Among the 3 models, Random Forest is the one with the best precision score of 0.80 and recall score of 0.65. It may appear that these scores are not satisfactory, however, predicting default risk is an inherently challenging task, and there is an inevitable trade-off between precision and recall. More importantly, this analysis is intended to be an aid to a human decision by flagging high default risk customers instead of automating the decision-making.

In this project, we discovered a few interesting insights which may or may not hold for other datasets. We learned the most important predictors of default are the most recent 2 months' payment status and users’ credit limit. The conventional thinking of younger people tend to have higher default risk seems to be somewhat inaccurate.

The machine learning models in this analysis can help credit card firms, loan lenders, and banks make educated decisions on creditworthiness based on easily accessible customer data. We recognize that credit card providers need to make decisions quickly while also adhering to rules. We propose that the model output probabilities rather than predictions to increase accuracy and provide human managers with more discretion over their decisions.

# Introduction

Economic transactions have evolved over recent years. The digital era has led to a completely new and innovative way of electronic transactions, which has heavily increased the convenience of purchases and made money much more accessible. Though the way people make transactions have changed over the years, the mode of payments has remained consistently dominated by credit.

Credit is defined as a designated amount, predetermined in advance, which a consumer can use to make payments with the included intension of repaying the amount in the future. There is a catch, however. Along with a predetermined limit, each credit account comes with a compounding interest rate, meaning that the consumer who used credit will have to repay a higher amount than what they borrowed. These interest rates make up a large portion of the revenue generated by the banking industry, giving credit the significance it holds in the economy today.

While interest payments from credit cards could likely fund the sector alone, credit remains a riskier investment due to the abundant amount of customers who fail to pay off their debt. Consequently, understanding who will fail to repay their credited amount, or *default on their payment*, and when is crucial. Predicting this phenomenon not only controls the financial damage caused by payment failures but can also be reimplemented into the system for new customer approval and their associated credit limits. This report will address possible avenues of exploration and predictions to be made considering credit card defaults, while also providing possible insight into what factors indicate default payments.

# Data Overview

## Data Collection

*Kaggle Dataset:* <https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset>

The original dataset can also be found at *UC Irvine Machine Learning Repository:* <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>

The dataset contains information on default payments, demographic information, credit limit, payment history, and bill statements of credit card holders in Taiwan from April 2005 to September 2005. It includes 30,000 observations (each representing a credit card user) and 25 columns. The payment amount and statement balance are NT$ (New Taiwan Dollars).

## Data Wrangling

Overall, the dataset is very clean, but there are several undocumented column values. As a result, most of the data wrangling effort was spent searching for information and interpreting the columns.

In the dataset, there are no missing values (NA/Null). Additionally, there are no records that are duplicates. We removed the “ID” column and retained the remaining 24 columns as predictors and the target variable for training the models.

A few columns (payment status, payment amount, and statement balance) have been renamed to better reflect the month they represent.

| **Existing Names** | **New Column Names** |
| --- | --- |
| PAY\_0, PAY\_2, … PAY6 | PAY\_STATUS\_SEP,  PAY\_STATUS\_AUG, … PAY\_STATUS\_APR |
| BILL\_AMT1, BILL\_AMT2…BILL\_AMT6 | BILL\_AMT\_SEP,  BILL\_AMT\_AUG,...BILL\_AMT\_APR |
| PAY\_AMT1, PAY\_AMT2…PAY\_AMT6 | PAY\_AMT\_SEP,  PAY\_AMT\_AUG,...PAY\_AMT\_APR |

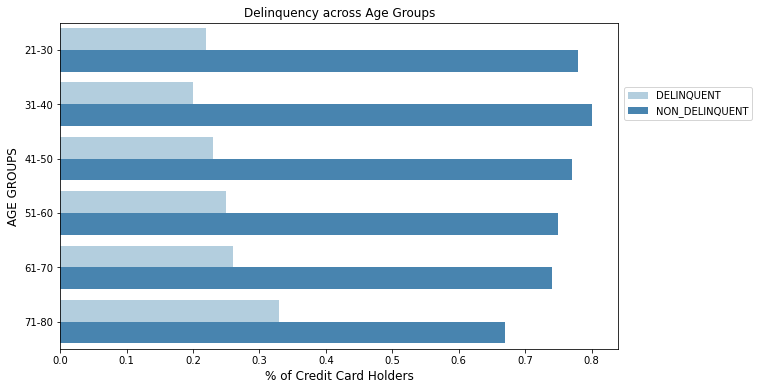
**Table 1: Factor Renaming**

# Data Exploration

Before moving into the prediction of credit card defaults, there are some notable observations to be made on the data set. Each of the following subsections provide an overview of notable observations, but avoid determining absolute relationships.

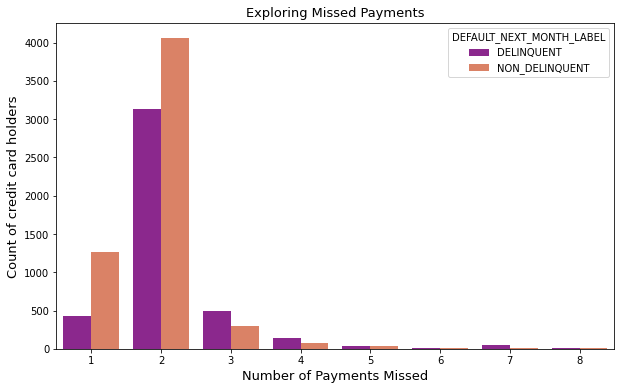
## Delinquency Distribution

The goal of this analysis is to analyze the demographic factors which could influence a default payment.

****

**Figure 4: Examining Delinquency across Different Age Groups**

The bar chart in figure 4 shows the percentage of defaulters in each age group from 20 to 80 years. Credit card users aged between 31 and 40 have the lowest percentage of defaults (about 20%), while users between 71 and 80 have the highest number of defaults (more than 30%). Additionally, we notice that as people get older, the percentage of defaulters gradually increased. However, this finding only applies to this dataset, and it should be further verified with statistical tests like Chi-Square, t-test, etc.

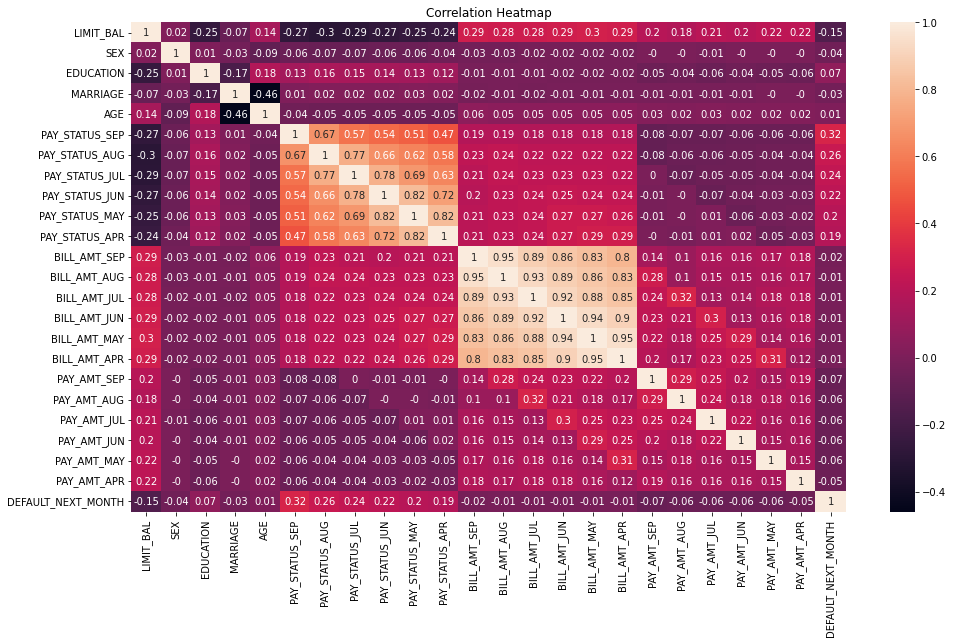
****

**Figure 5: Exploring the number of missed payments**

The bar graph in figure 5 shows the number of payments that users have missed. Although there is no apparent trend, it was surprising that some users have been late on payments for 7-8 months.

## Factor Correlation

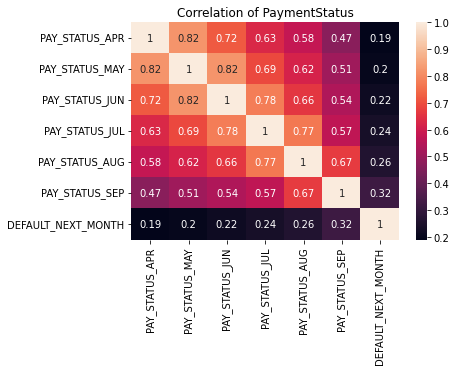
Before determining any modelistic insights, it was necessary to first analyze which factors would be most expected to indicate credit card default. Because there are 24 variables (including our default variable), exact correlation are difficult to identify in a matrix, so color shadings were used to identify which factors are most influential (which is shown in the heatmap below).

****

**Figure 1: Correlation Heatmap (all variables)**

After reviewing the correlation matrix, it is observed that there are few variables that heavily influence our default variable (or many variables that have strong correlations altogether). Still, it can be noted that there are clusters of variables which are heavily correlated with each other (and no other variables). This is seen most in the payment status and billing amount, where different months are often correlated with each other, but no other variables.

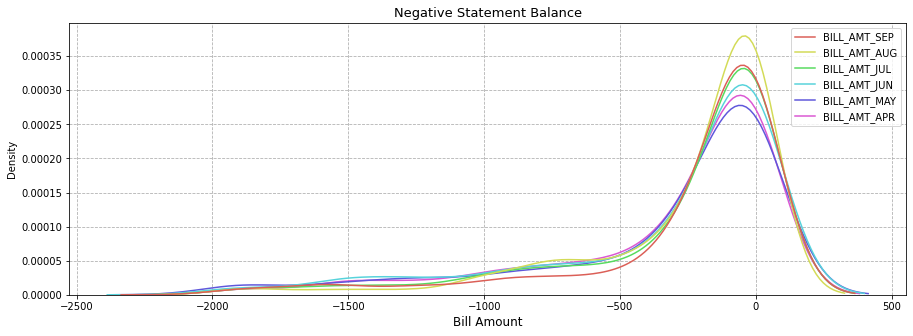
There are some other relationships, like bills and payments by month or age and marriage, but their relationships are reasonable and expected. Still, there is one other relationship worth noting: the correlation between our default variable and the payment status from september (the previous month). Reviewing our initial matrix, the payment status grouping appeared to influence our predicting variable more than any other variable. The relationship can be analyzed further in the correlation heatmap of only the default variable and the payment statuses.

****

**Figure 3: Variable-Filtered Correlation Heatmap (Comparing Default and Payment Status)**

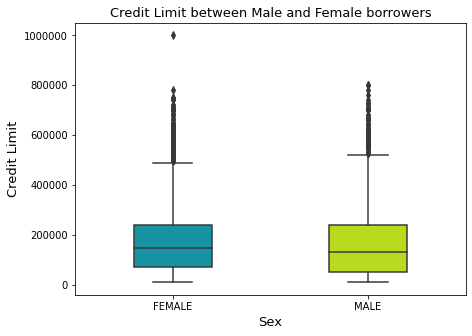
After observing the relationship further, the payments are still relatively weak when considering credit card default. Still, they do provide higher values than previously observed, which will be taken into account when generating our predictive models.

## Other Notable Observations

****

**Figure 6: Statement Balance Distribution**

A density plot of statement balance that focuses on the negative bill statements is shown in Figure X. We notice that some of the statement balances are negative. It suggests that a few users are overpaying their credit card bills.

****

**Figure 7: Examining the Credit Limit for any Gender Bias**

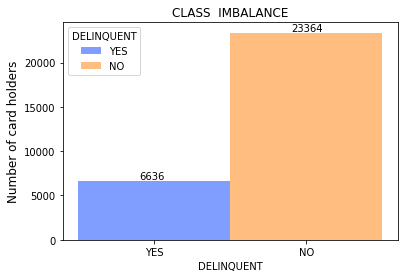
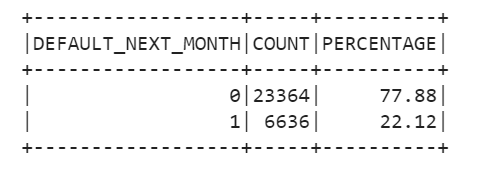
We used boxplots to understand the range of credit limits between male and female borrowers. The interquartile regions of both boxplots highly overlap. This indicates there is no significant difference between the credit limits of men and women.

# Model Methodology

Since the dataset that we have is labeled and the predicted outcome is the probability of customer default, we characterize this as supervised machine learning and it is a binary classification problem. We performed a few pre-processing steps before modeling to improve the model performance.

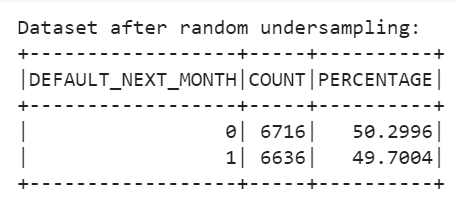
## Handling Class Imbalance

It is obvious that the majority of the users do not default. This dataset is probably going to be dominated by non-defaults, with occasional defaults. Machine learning algorithms will be misled and perform poorly with an unbalanced dataset.

****

**Figure 8: Imbalanced Dataset**

As seen in the above stats and the histogram, the number of defaulters is around 6,636 (22%) when compared to non-defaulters around 23,364 (78%). The class ratio is roughly 1: 3.5. We consider this dataset as imbalanced. The model's accuracy can be severely hampered by imbalanced data.   
 For this project, we have used the “random under-sampling” technique to balance the classes (default and non-default). Random undersampling aims to balance class distribution by randomly eliminating majority class observations (non-defaults). This is done until the majority and minority class instances are balanced out. The main drawback of this technique is that it can discard potentially useful information which could be important for building rule classifiers.

****

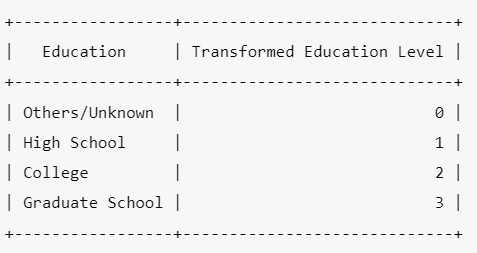
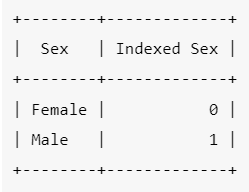
**Figure 9: Dataset after random undersampling**

Figure 9 shows the proportion of classes in the column “DEFAULT\_NEXT\_MONTH” after undersampling.

## Feature Engineering

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning. In our dataset, there are several categorical variables - Education, Sex, Marital Status, etc., which need to be transformed appropriately.

In the dataset, the “EDUCATION” column is an ordinal categorical variable. In the case of ordinal variables, retaining the order is recommended. Therefore, it is encoded according to an increasing level of education as shown in the figure on the left.

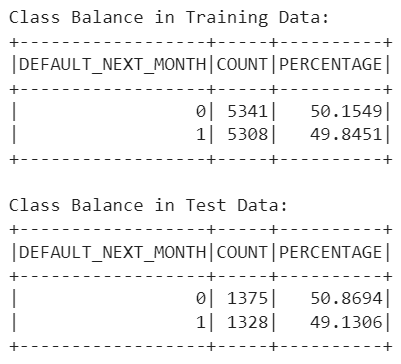
**Table 2: Encoding categorical variables**

The “MARRIAGE” column is an unordered, plain categorical variable. There is no intrinsic order to the values in this column, therefore it is one-hot encoded.

There are only two categories in the "SEX" column (male and female). For the encoding, just 0 and 1 were used as shown in the figure on the right.

We randomly split our datasets into training and test data with a ratio of 80:20. There are 2,703 observations in the test dataset compared to 10,649 in the training dataset. For each model, we have used the same ratio for training and test data to ensure consistency.

Additionally, we made sure the class balance was maintained in the train and test datasets. 50% of the default and non-default classes are present in each dataset (shown below).



**Table 3: Validating class balance in training and testing datasets**

## Model Structuring

Once the features have been prepped and balancing was completed, the data was ready for predictive models. The sequence used for model creation was to develop a standing model with baseline parameters, then perform a post hoc error analysis and hyperparameter tuning.

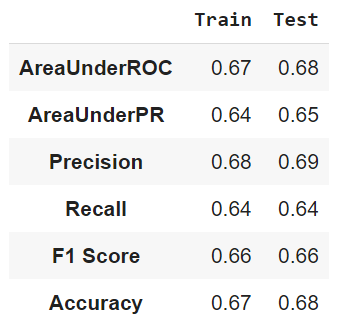
Each model experimented with a pyspark grid search, which had varying levels of success. Best parameters from the grid search were determined by the ROC AUC measures and some cross-fold validation. The deep learning model ran multiple grid searches, where each instance included a different amount of hidden layers or total hidden layer neurons. Once the best parameters of each model were determined, the parameters were hard coded into the program.

Once the best parameters were established, a preliminary error analysis was performed to test the overal performance of the model. Other metrics, including precision, recall, and f1 scores, were used to judge the models performance. Assuming the metrics were satisfactory compared to the base model, the model then moved into a more in depth error analysis where error reasoning was analyzed and feature importance was determined.

# Credit Card Payment Default Prediction Results

## Logistic Regression (default params)

The logistic regression model with default parameters was trained with 23 features. All the features were standardized for faster training and comparison of the coefficients. The error metrics of this model are shown below:



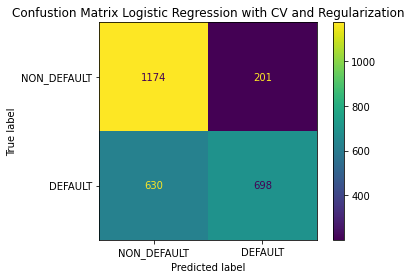
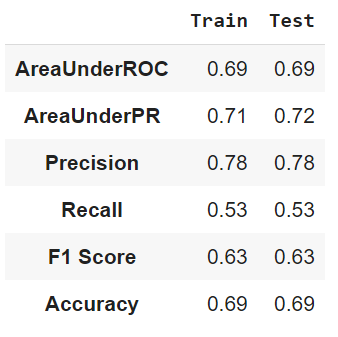
**Figure 10: Logistic Regression Error Metrics (left) and Confusion Matrix (right)**

Since the dataset is balanced, accuracy is a good measure to evaluate the model's performance. The accuracy of this model is 68%. The accuracy of the test data is slightly higher than the training data. This indicates the model is not overfitting.

## Logistic Regression (with Regularization)

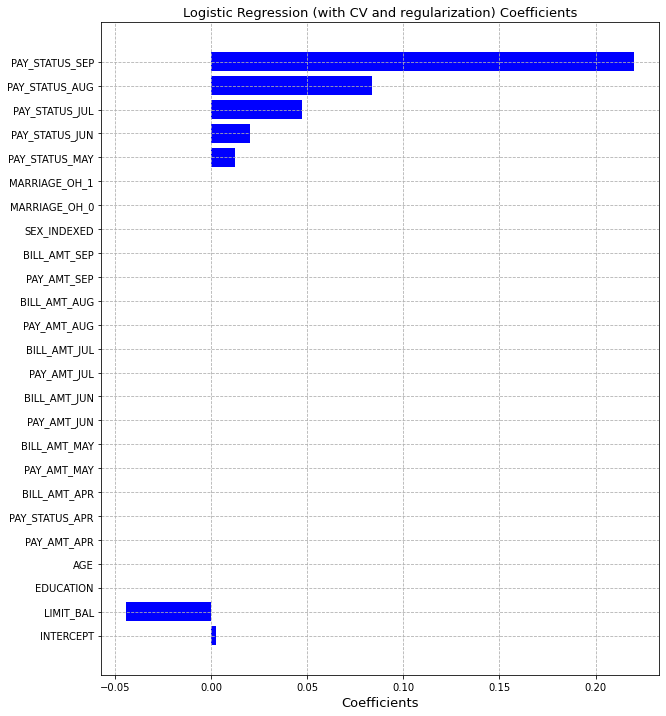
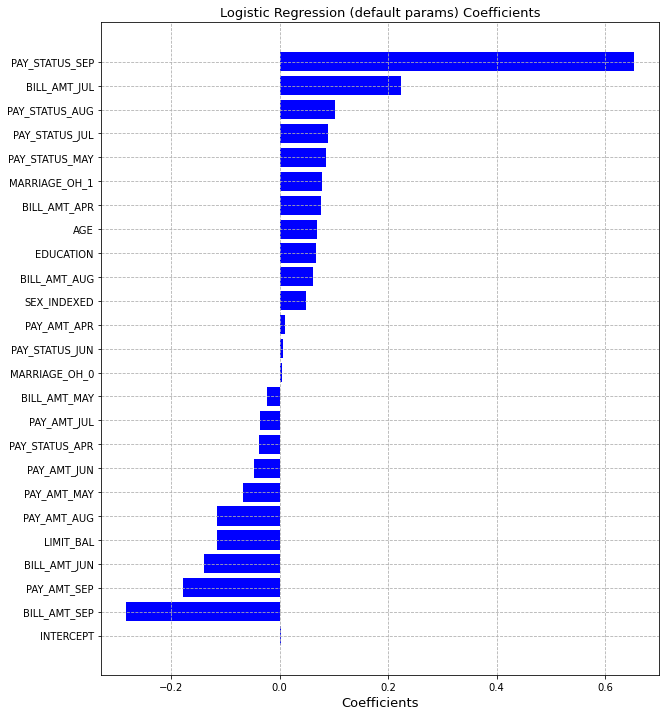
We used elastic net regularization when training this logistics regression model. The grid search with 3-fold cross-validation was used to find the optimal parameters (alpha and lambda) for regularization.

The performance of the logistic regression model does not appear to have improved significantly. The model’s precision has increased by 10% while its recall decreased by 10% compared to the base model (logistic regression with default parameters). The lower recall suggests that the model is categorizing more future defaulters as non-defaulters. It can be evidenced in the confusion matrix’s second row and first column (the base model has a lower value than the regularized model).



**Figure 12: Logistic Regression error metrics (left) and confusion matrix (right)**

Additionally, we observed that the coefficients of 17 features in the regularized model had been reduced to 0 (shown below).



**Figure 13: Coefficients of Logistic Regression (default and with regularization)**

## Random Forest

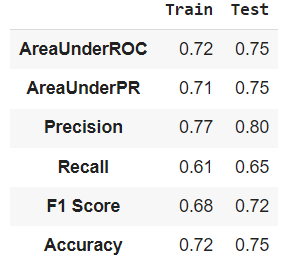
Similar to the logistic regression model, the random forest model was constructed to predict credit card payment defaults in October of 2005 based on customer payment history from the previous six months. Each engineered feature was used, along with a standardized layer for feature comparison. As for model parameter specifics, the details of the grid search are shown in the table below.

| **Parameter** | **Max Depth** | **Max Bins** | **Number of Trees** | **Subsampling Rate** | **Bootstrap** |
| --- | --- | --- | --- | --- | --- |
| **Grid Seach Inputs** | 4, 5, 6, 7, 8 | 25, 32, 40 | 15, 20, 25, 30 | 0.1, 0.25, 0.5, 1.0 | True, False |
| **Best Parameter** | 8 | 32 | 25 | 0.5 | False |

**Table 5: Random Forest Grid Search Parameters**

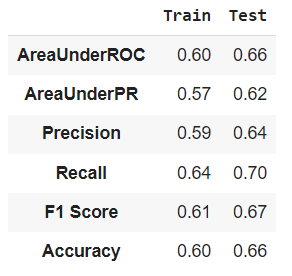
Running a grid search with these parameters required about three hours to run with available resources from Google Colab. A maximum depth larger than 8 may have led to stronger results, but was not tested due to performance restrictions. The best model’s error metrics are shown on the right.

Though the results yielded a lower recall score, these results are still relatively strong compared to the logistic regression and supporting documentation of the dataset.



**Table 6: Random Forest Error Metrics**

## Deep Learning

Similar to the previous models, the deep learning model was used to predict credit card delinquency. However, parameter tuning for the model was slightly different. The deep learning model’s runtime is heavily influenced by maximum iterations, number of hidden layers, and the number of neurons in the network layers. Consequently, a simple neural network was used in a grid search to determine the best block size (64) and step size (0.005). Then, variations of neural networks were tested individually with different combinations of hidden layers, hidden layer neurons, and max iterations. Because the training time increased very quickly as each of those parameters were increased, testing on this model was limited.

Of the few models that completed with in a 6 hour runtime, the best neural network was created using a maximum iteration of 25000 on a model with one hidden layer consisting of 500 hidden layers. The results of the model are shown in Table 7 on the right.:

**Table 7: Deep Learning Error Metrics**

While the model did produce some results with an even error spread, it struggled in overall accuracy compared to the previous models. More analysis will be conducted in the following section.

# Credit Card Default Analysis

## Logistic Regression

The recall for the logit regression model (with default parameters) is 64%, whereas the recall for the model with regularization is 53%. The precision of the logistic regression model with default parameters is 69% whereas the precision for the model with regularization is 78%.

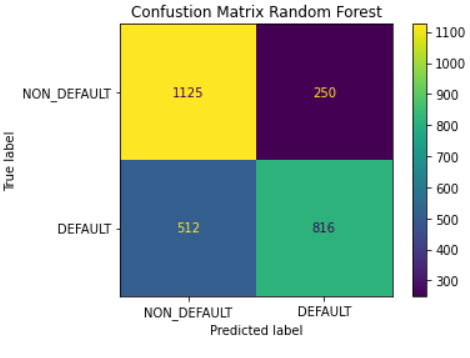
The regularized model's lower recall indicates that it is incorrectly identifying future defaulters as non-defaulters (higher false negatives). This leads us to the conclusion that the logistic regression model with default parameters gives better predictions despite the regularized model's higher precision.

The regularized model of logistic regression might be beneficial in the feature selection process. The coefficients were decreased from 23 features to just 7 features (payment status from April 2005 to September 2005 and credit limit). It was interesting to notice that the correlation matrix also showed similar correlations. The model's findings demonstrate,

1. The likelihood of default increases with the increase in payment delays
2. As credit limits are decreased, the probability of default increases

## Random Forest

As observed from the previous section, random forest predicted the largest portion of credit card defaults and non-defaults correctly. About 25% of the data, however, was not predicted accurately. To understand the model shortcomings, the error distribution will first be analyzed through the chart below.



**Figure 14: Random Forest Error Distribution**

As expected, the error distribution provides visual support for the higher precision and lower recall from the previous section. Precision, in the scope of this analysis, is desired much more than recall; of predicted defaults, the model produces a stronger proportion of correct predictions. Specifically, the model is capable of flagging customers as potential delinquets and pass them on to the organization to preemptively analyze there account and perhaps provide relief of repayment support where necessary.

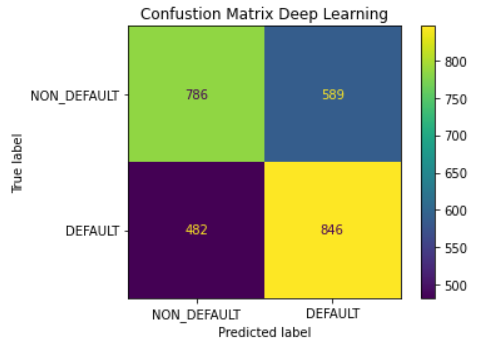
Still, even with the relatively strong precision, the model did show a large amount of error. Looking into the specific errors, the model struggled with a few different scenarios:

1. Model struggled to predict any non-default payments when one of the previous months had a defaulted payment.
2. Instances with high billing amounts and low payment amounts were consistently predicted incorrectly as defaulted
3. Instances with no default were predicted as default, likely due to no payments increasing the number of trees may help with these errors.

Before moving to the next model, it is important to note that feature importance analysis was conducted for the random forest model. Similar to the initial analysis on factor correlation, the high influencing factors of delinquency were the payment statuses from previous months and the credit limit.

## Deep Learning

Looking back at the results mentioned from the model, the deep learning model sought a relatively even error distribution. Recall, for instance, was much higher than the previous models, leaving it a decent addition to the generated models. The specific error distribution is shown below:

****

**Figure 15: Deep Learning Error Distribution**

Though the proportion of errors is more even in this model, error analysis revealed that the model still struggles with previously mentioned problems of high billing amounts and previous defaults. Perhaps a deeper model with more layers, neurons, and iterations would resolve this issue, but Google Colab does not provide necessary resources for the completion of this task. Furthermore, running these models would heavily increase the time to train the model, which was just not available given the timetable of the project.

So considering the performance, along with the model accuracy, the model does not meet the ability of the other models and should be disregarded despite the error distribution.

# Conclusion

Several factors contribute to a person’s credit card payments. Anything from purchasing a new streaming service to a mortgage on a new house can influence someone’s ability to pay credit car payments, but that data is not always available. Predicting defaulted payments from basic first party data and previous payments can be difficult, but any indication can help the industry's business decisions.

With only the basic amount of data available, implementing the random forest model is recommended. The model predicts the strongest proportion of defaults with high precision. Furthermore, the model addressed in this paper is not realistically overfitting, meaning that increasing the number of trees would likely improve the model. With the current resources available and associated time constraints, that was not available in this model, but would certainly be available for future exploration.

Incorporating the logistic regression models would also prove beneficial. The base logistic regression model produced a decent accuracy with a balanced error distribution, leaving it as a good comparison for future models and an option to observe recall predictions. The tuned logistic regression model also provides some value insight. The feature correlation extracted from the model shows which factors truly indicated credit card delinquency, leading to insight on how to prevent or prepare for it.

Much was extracted from the analysis conducted on credit card default payments, but more could be obtained. Extra resources or time focused on model tuning may yield even stronger results, whether it be from the expansion of a random forest model or going deeper into a neural network. Looking past model performance, the data is organized by singular points by account. Instead, the dataset contains 6 months of information compiled into one original point. Expanding that into a time series may yield some noticable insight, especially considering one of the strongest causes of error was one previous payment default at anytime. Yet even if the data was split into users with previous defaults and without previous defaults, rerunning the models may resolve that error entirely.

Though there is more to be done, the current models provide a strong starting point. Initial insight into credit card default has been obtained, and can now give a deeper understanding to those want to reduce the impact of credit card delinquency.

# Appendix

## Data Dictionary

There are 25 columns in this dataset:

| **Attribute Name** | **Attribute Description** |
| --- | --- |
| LIMIT\_BAL | Credit limit (in NT$) |
| SEX | Gender (1=Male; 2=Female) |
| EDUCATION | Highest level of education (1=Graduate School; 2=University; 3=High Scool; 4=Others) |
| MARRIAGE | Marital Status (1=Married; 2=Single; 3=Others) |
| AGE | Age (years) |
| PAY\_0 to PAY\_6 | History of payments (from April 2005 - September 2005)  (-1=pay duly; 1=payment delay for one month; 2=payment delay for two months, … 8=payment delay for eight months; 9=payment delay for nine months and above) |
| BILL\_AMT1 to BILL\_AMT6 | Amount of bill statement (in NT$)  BILL\_AMT1 = amount of bill statement in September 2005…  BILL\_AMT6=amount of bill statement in April 2005 |
| PAY\_AMT1 to PAY\_AMT6 | Amount of previous payment (in NT$)  PAY\_AMT1=amount paid in September 2005…  PAY\_AMT6= amount paid in April 2005 |
| default.payment.next.month | Default in October 2005 (1=yes, 0=no) |

## References

guest\_blog. (2020, July 23). *Imbalanced Classification | Handling Imbalanced Data using Python*. Analytics Vidhya. Retrieved December 11, 2022, from <https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/>

W3Schools. (2000). *Python Machine Learning - Confusion Matrix*. Python Machine Learning - Confusion Matrix. Retrieved December 11, 2022, from <https://www.w3schools.com/python/python_ml_confusion_matrix.asp>