# Credit Card Default Prediction

Adrian Sandoval-Vargas

Department of Computer Science

Oakland University

## Abstract

This study report was aimed to encompass *"The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients"* by I-Cheng Yeh and Che-hui Lien [1] using statistical analysis, data mining techniques, and limiting the classification algorithm to Random Forest. The data used was 30,000 credit card holders from Taiwan in a period of 6 months [2]. In this study we will investigate different which data segmentation and data preprocessing approach yields the best accuracy.

## Introduction

This report focuses of the use of the Random Forest Classifier [3] for the primary reason that our data is structured in such a way that it fits the supervised schema for the ensembled Random Forest Classifier. The data was obtained from the University of California – Irvine in the form of an excel spread sheet. The classifier should classify if credit card holders will default or not based on this data. I will first preprocess the data and clean the data with any outliers in specific area of interest and provide some visuals on this cleaning process. I will keep the original cleaned data and create segmentations of the data into different groups that best depicts a boundary. Once data is segmented, I will be discussing the different approaches to deal with data imbalance. After dealing with the data imbalance, I will conduct a series of validations of F1, Recall, and Precision scores to determine which data segments performed the best. In addition, I will experiment with Normalization and Principle Component Analysis to specific segments and discuss the effects on the recall and accuracy.

The paper will be laid out in 4 sections. In section 1, I will introduce the data, discuss and visualize key features, and perform statistical analysis to get a clean dataset. In section 2, I will discuss on random forest classifiers and my parameters as well as how to deal with data imbalance. Section 3 will be my experimentation and the tests conducted on the data. Section 4 will the final result and thoughts on what I should have done differently.

## 1. Dataset Analysis

The dataset was obtained in the form of an excel spread sheet from UCI. It was collected by a group of researchers in Taiwan over a period of 6 months from credit card holders. The dataset contains index id, 23 variables, and its target (1 = default, 0 = did not default). The 23 variables are a mixture categorical and numerical discrete values. The variables are as follows:

LIMIT_BAL

SEX (1 = male, 2 = female)

EDUCATION (1 = grad school, 2 = university, 3 = high school, 4 = other)

MARRIAGE (1 = married, 2 = single, 3 = other)

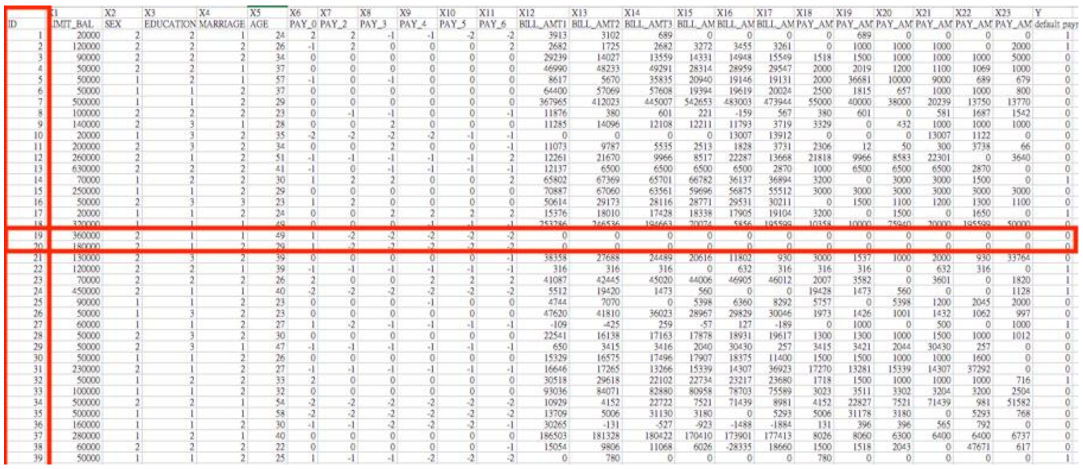
AGE

PAY_0, PAY_2, …, PAY_6

BILL_AMT1, BILL_AMT2, … , BILL_AMT6

PAY_AMT1, PAY_AMT2, … , PAYAMT_6

On a visual inspection we do not need the 'ID' column as it only marks the index of the record. We also see records in our dataset that contain -2 for the "Pay_X" (where X means 1,…,6 ) columns that have a default value of 0 and 1 which do not make any sense to keep in our dataset.



| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | IMIT_BAL | SEX | EDUCATION | MARRIAGE | AGE | PAY_0 | PAY_2 | PAY_3 | PAY_4 | PAY_5 | PAY_6 | BILL_AMT1 | BILL_AMT2 | BILL_AMT3 | BILL_AM | BILL_AM | BILL_AM | PAY_AM | PAY_AM | PAY_AM | PAY_AM | PAY_AM | PAY_AM | default pay |
| 1 | 20000 | 2 | 2 | 1 | 24 | 2 | 2 | -1 | -1 | -2 | -2 | 3913 | 3102 | 689 | 0 | 0 | 0 | 0 | 689 | 0 | 0 | 0 | 0 | 1 |
| 2 | 120000 | 2 | 2 | 2 | 26 | -1 | 2 | 0 | 0 | 0 | 2 | 2682 | 1725 | 2682 | 3272 | 3455 | 3261 | 0 | 1000 | 1000 | 1000 | 0 | 2000 | 1 |
| 3 | 90000 | 2 | 2 | 2 | 34 | 0 | 0 | 0 | 0 | 0 | 0 | 29239 | 14027 | 13559 | 14331 | 14948 | 15549 | 1518 | 1500 | 1000 | 1000 | 1000 | 5000 | 0 |
| 4 | 50000 | 2 | 2 | 1 | 37 | 0 | 0 | 0 | 0 | 0 | 0 | 46990 | 48233 | 49291 | 28314 | 28959 | 29547 | 2000 | 2019 | 1200 | 1100 | 1069 | 1000 | 0 |
| 5 | 50000 | 1 | 2 | 1 | 57 | -1 | 0 | -1 | 0 | 0 | 0 | 8617 | 5670 | 35835 | 20940 | 19146 | 19131 | 2000 | 36681 | 10000 | 9000 | 689 | 679 | 0 |
| 6 | 50000 | 1 | 1 | 2 | 37 | 0 | 0 | 0 | 0 | 0 | 0 | 64400 | 57069 | 57608 | 19394 | 19619 | 20024 | 2500 | 1815 | 657 | 1000 | 1000 | 800 | 0 |
| 7 | 500000 | 1 | 1 | 2 | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 367965 | 412023 | 445007 | 542653 | 483003 | 473944 | 55000 | 40000 | 38000 | 20239 | 13750 | 13770 | 0 |
| 8 | 100000 | 2 | 2 | 2 | 23 | 0 | -1 | -1 | 0 | 0 | -1 | 11876 | 380 | 601 | 221 | -159 | 567 | 380 | 601 | 0 | 581 | 1687 | 1542 | 0 |
| 9 | 140000 | 2 | 3 | 1 | 28 | 0 | 0 | 2 | 0 | 0 | 0 | 11285 | 14096 | 12108 | 12211 | 11793 | 3719 | 3329 | 0 | 432 | 1000 | 1000 | 1000 | 0 |
| 10 | 20000 | 1 | 3 | 2 | 35 | -2 | -2 | -2 | -2 | -1 | -1 | 0 | 0 | 0 | 0 | 13007 | 13912 | 0 | 0 | 0 | 13007 | 1122 | 0 | 0 |
| 11 | 200000 | 2 | 3 | 2 | 34 | 0 | 0 | 2 | 0 | 0 | -1 | 11073 | 9787 | 5535 | 2513 | 1828 | 3731 | 2306 | 12 | 50 | 300 | 3738 | 66 | 0 |
| 12 | 260000 | 2 | 1 | 2 | 51 | -1 | -1 | -1 | -1 | -1 | 2 | 12261 | 21670 | 9966 | 8517 | 22287 | 13668 | 21818 | 9966 | 8583 | 22301 | 0 | 3640 | 0 |
| 13 | 630000 | 2 | 2 | 2 | 41 | -1 | 0 | -1 | -1 | -1 | -1 | 12137 | 6500 | 6500 | 6500 | 6500 | 2870 | 1000 | 6500 | 6500 | 6500 | 2870 | 0 | 0 |
| 14 | 70000 | 1 | 2 | 2 | 30 | 1 | 2 | 2 | 0 | 0 | 2 | 65802 | 67369 | 65701 | 66782 | 36137 | 36894 | 3200 | 0 | 3000 | 3000 | 1500 | 0 | 1 |
| 15 | 250000 | 1 | 1 | 2 | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 70887 | 67060 | 63561 | 59696 | 56875 | 55512 | 3000 | 3000 | 3000 | 3000 | 3000 | 3000 | 0 |
| 16 | 50000 | 2 | 3 | 3 | 23 | 1 | 2 | 0 | 0 | 0 | 0 | 50614 | 29173 | 28116 | 28771 | 29531 | 30211 | 0 | 1500 | 1100 | 1200 | 1300 | 1100 | 0 |
| 17 | 20000 | 1 | 1 | 2 | 24 | 0 | 0 | 2 | 2 | 2 | 2 | 15376 | 18010 | 17428 | 18338 | 17905 | 19104 | 3200 | 0 | 1500 | 0 | 1650 | 0 | 1 |
| 18 | 320000 | 1 | 1 | 1 | 49 | 0 | 0 | 0 | -1 | -1 | -1 | 253286 | 246536 | 194663 | 70074 | 5856 | 195599 | 10358 | 10000 | 75940 | 20000 | 195599 | 50000 | 0 |
| 19 | 360000 | 2 | 1 | 1 | 49 | 1 | -2 | -2 | -2 | -2 | -2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 180000 | 2 | 1 | 2 | 29 | 1 | -2 | -2 | -2 | -2 | -2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 130000 | 2 | 3 | 2 | 39 | 0 | 0 | 0 | 0 | 0 | -1 | 38358 | 27688 | 24489 | 20616 | 11802 | 930 | 3000 | 1537 | 1000 | 2000 | 930 | 33764 | 0 |
| 22 | 120000 | 2 | 2 | 1 | 39 | -1 | -1 | -1 | -1 | -1 | -1 | 316 | 316 | 316 | 0 | 632 | 316 | 316 | 316 | 0 | 632 | 316 | 0 | 1 |
| 23 | 70000 | 2 | 2 | 2 | 26 | 2 | 0 | 0 | 2 | 2 | 2 | 41087 | 42445 | 45020 | 44006 | 46905 | 46012 | 2007 | 3582 | 0 | 3601 | 0 | 1820 | 1 |
| 24 | 450000 | 2 | 1 | 1 | 40 | -2 | -2 | -2 | -2 | -2 | -2 | 5512 | 19420 | 1473 | 560 | 0 | 0 | 19428 | 1473 | 560 | 0 | 0 | 1128 | 1 |
| 25 | 90000 | 1 | 1 | 2 | 23 | 0 | 0 | 0 | -1 | 0 | 0 | 4744 | 7070 | 0 | 5398 | 6360 | 8292 | 5757 | 0 | 5398 | 1200 | 2045 | 2000 | 0 |
| 26 | 50000 | 1 | 3 | 2 | 23 | 0 | 0 | 0 | 0 | 0 | 0 | 47620 | 41810 | 36023 | 28967 | 29829 | 30046 | 1973 | 1426 | 1001 | 1432 | 1062 | 997 | 0 |
| 27 | 60000 | 1 | 1 | 2 | 27 | 1 | -2 | -1 | -1 | -1 | -1 | -109 | -425 | 259 | -57 | 127 | -189 | 0 | 1000 | 0 | 500 | 0 | 1000 | 1 |
| 28 | 50000 | 2 | 3 | 2 | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 22541 | 16138 | 17163 | 17878 | 18931 | 19617 | 1300 | 1300 | 1000 | 1500 | 1000 | 1012 | 0 |
| 29 | 50000 | 2 | 3 | 1 | 47 | -1 | -1 | -1 | -1 | -1 | -1 | 650 | 3415 | 3416 | 2040 | 30430 | 257 | 3415 | 3421 | 2044 | 30430 | 257 | 0 | 0 |
| 30 | 50000 | 1 | 1 | 2 | 26 | 0 | 0 | 0 | 0 | 0 | 0 | 15329 | 16575 | 17496 | 17907 | 18375 | 11400 | 1500 | 1500 | 1000 | 1000 | 1600 | 0 | 0 |
| 31 | 230000 | 2 | 1 | 2 | 27 | -1 | -1 | -1 | -1 | -1 | -1 | 16646 | 17265 | 13266 | 15339 | 14307 | 36923 | 17270 | 13281 | 15339 | 14307 | 37292 | 0 | 0 |
| 32 | 50000 | 1 | 2 | 2 | 33 | 2 | 0 | 0 | 0 | 0 | 0 | 30518 | 29618 | 22102 | 22734 | 23217 | 23680 | 1718 | 1500 | 1000 | 1000 | 1000 | 716 | 1 |
| 33 | 100000 | 1 | 1 | 2 | 32 | 0 | 0 | 0 | 0 | 0 | 0 | 93036 | 84071 | 82880 | 80958 | 78703 | 75589 | 3023 | 3511 | 3302 | 3204 | 3200 | 2504 | 0 |
| 34 | 500000 | 2 | 2 | 1 | 54 | -2 | -2 | -2 | -2 | -2 | -2 | 10929 | 4152 | 22722 | 7521 | 71439 | 8981 | 4152 | 22827 | 7521 | 71439 | 981 | 51582 | 0 |
| 35 | 500000 | 1 | 1 | 1 | 58 | -2 | -2 | -2 | -2 | -2 | -2 | 13709 | 5006 | 31130 | 3180 | 0 | 5293 | 5006 | 31178 | 3180 | 0 | 5293 | 768 | 0 |
| 36 | 160000 | 1 | 1 | 2 | 30 | -1 | -1 | -2 | -2 | -2 | -1 | 30265 | -131 | -527 | -923 | -1488 | -1884 | 131 | 396 | 396 | 565 | 792 | 0 | 0 |
| 37 | 280000 | 1 | 2 | 1 | 40 | 0 | 0 | 0 | 0 | 0 | 0 | 186503 | 181328 | 180422 | 170410 | 173901 | 177413 | 8026 | 8060 | 6300 | 6400 | 6400 | 6737 | 0 |
| 38 | 60000 | 2 | 2 | 2 | 22 | 0 | 0 | 0 | 0 | 0 | -1 | 15054 | 9806 | 11068 | 6026 | -28335 | 18660 | 1500 | 1518 | 2043 | 0 | 47671 | 617 | 0 |
| 39 | 50000 | 1 | 1 | 2 | 25 | 1 | -1 | -1 | -2 | -2 | -2 | 0 | 780 | 0 | 0 | 0 | 0 | 780 | 0 | 0 | 0 | 0 | 0 | 1 |

Figure 1: the column ‘ID’ is irrelevant to the dataset. Scenario where the data contains -2’s and 0’s and has a default value of 0.

Upon writing python code to observe all incidents where all these values occur, we get

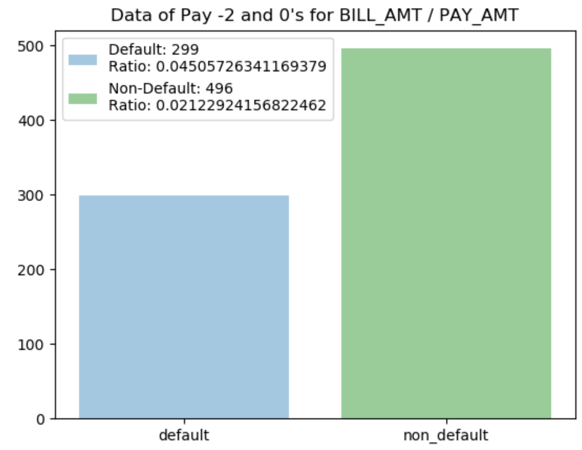


Figure 2: the amount of Default and Non-Default records what contain -2 in the PAY and 0 in BILL_AMT and PAY_AMT and its ratio to it’s respective target in the overall dataset.

Since these values are not going to constitute any value to our classifier it is best to remove them.

There are also values in our dataset where they were not specified by the dataset description that we must drop.

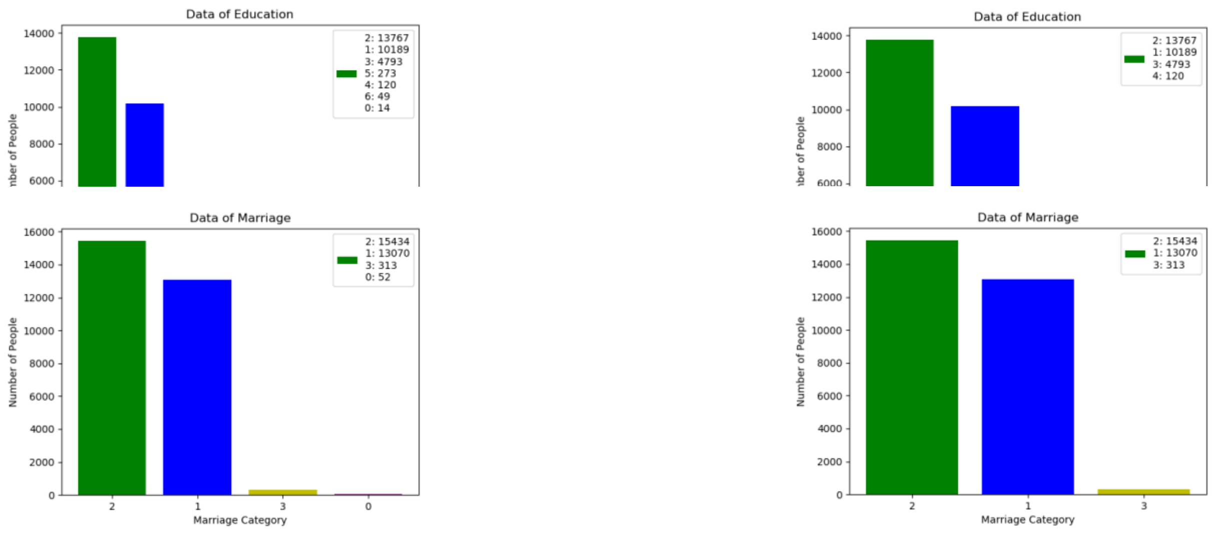


Figure 3: removal of the unspecified values for EDUCATION and MARRIAGE

In general, for any supervised ensemble classifier it is best to check if there is any imbalance in our dataset.

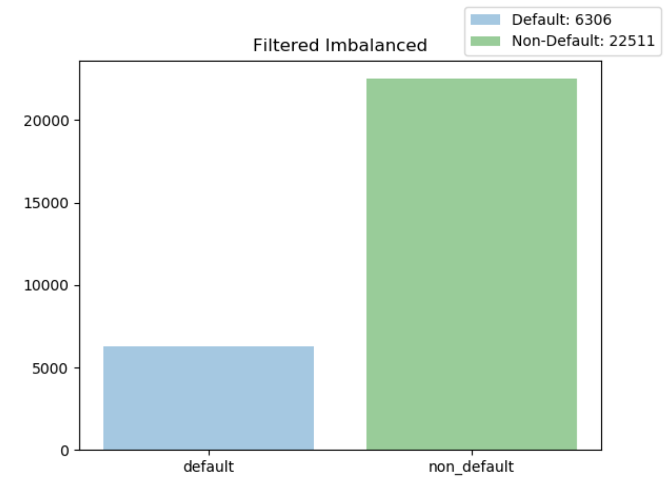


Figure 4: the amount of Default’s vs Non-Default’s in our dataset

Due to this imbalance of approximately 28% to 72% if we were to train our Random Forest Classifier it will automatically assume Non-Default due to the amount of data that is Non-Default. In order to have a good dataset to train our classifier, we must do further analysis on the dataset. Since the domain is regarding humans it is crucial that we focus statistical analysis on AGE and LMIT_BAL due to its realistic limits. For example, while it is possible that a 78-year-old will have a credit card account, they are in the minority which could indicate a possible outlier. That is what we will analyze in AGE vs LIMIT_BAL.

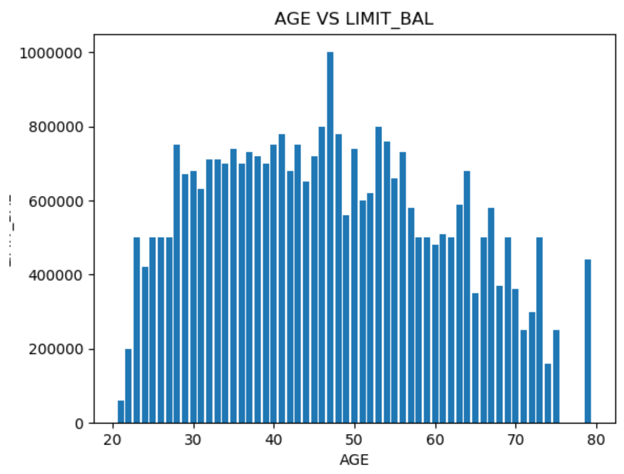


Figure 5: histogram of AGE vs LIMIT_BAL

We can see that there is slight skew to the right. So, we must clean up the data by using statistical analysis. Processing the AGE vs LIMIT_BAL into a standard normal curve we get:

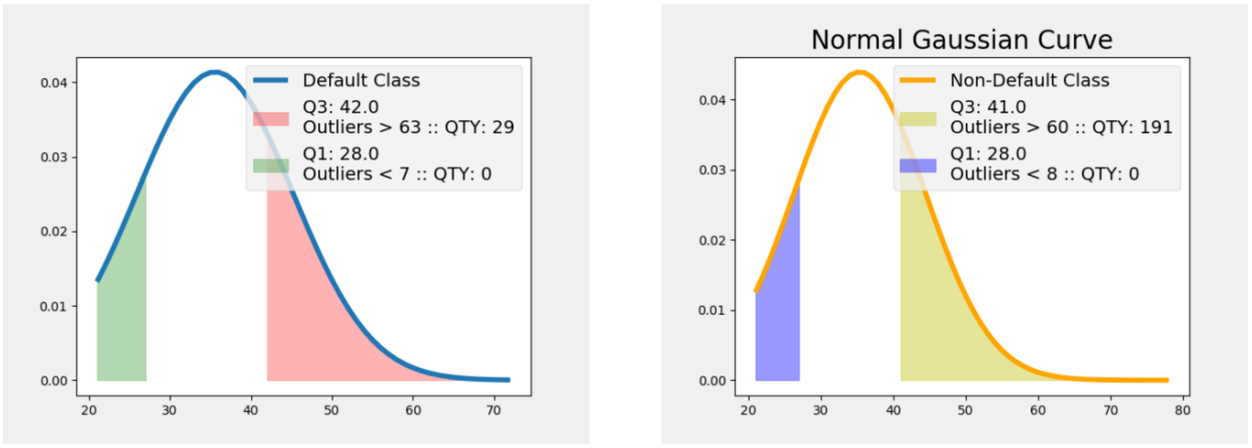
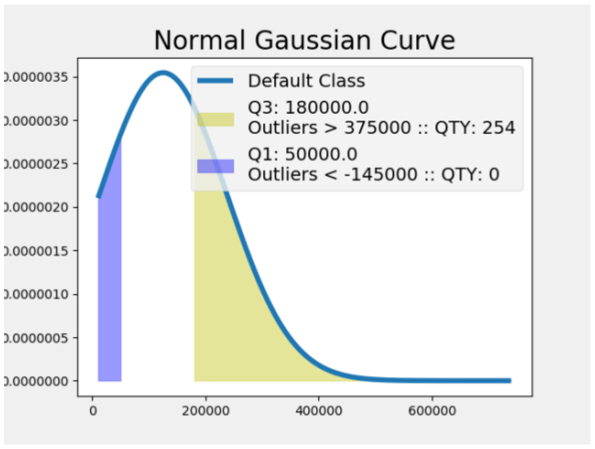


Figure 6: Normal Guassian Curve of default and non-default targets.

Given q3 and q1 we can determine the IQR and then apply the standard q1 – (1.5*IQR) and q3 + (1.5*IQR) to obtain the outliers (shown in figure 6). The exact process was repeated for

LIMIT_BAL and the normal distribution curve is as follows:



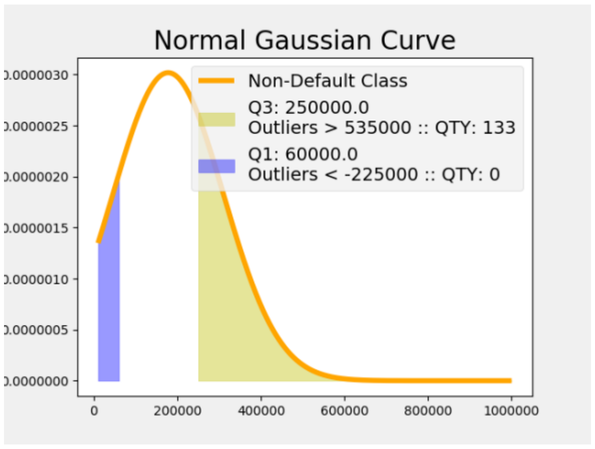


Figure 7: LIMTI_BAL Normal Guassian Curve of default and non-default targets.

This analysis results us in a more equally distributed of AGE vs LIMIT_BAL

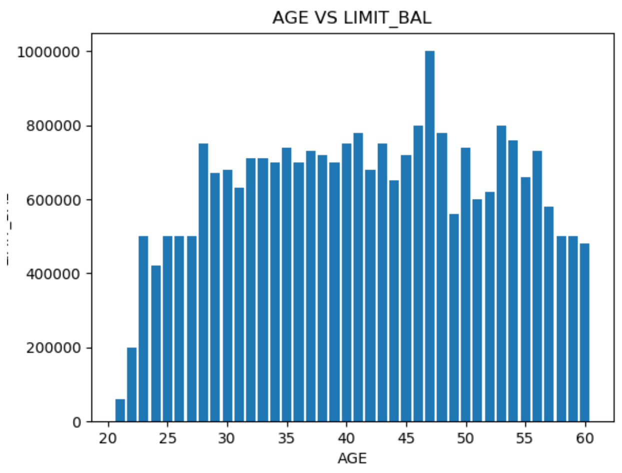


Figure 8: removed outliers in AGE and LIMIT_BAL

With this restructuring of our data, we will obtain an imbalance of our data of:

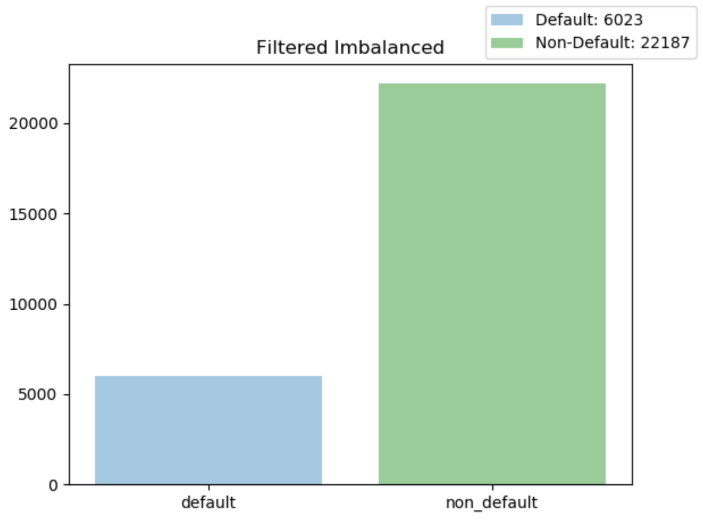


Figure 9: resulting imbalanced data

This statistical analyis gives us a better distribution of AGE vs LIMIT_BAL. I tried to do the same for EDUCATION and other features, but I concluded that these outliers scenrios will not help with decreasing the imbalance and will affect the classifier in validation. For example,

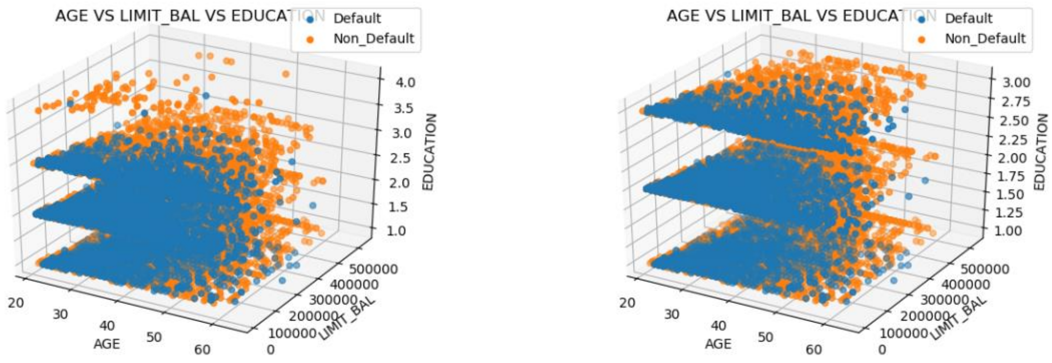


Figure 10: visualization of education outliers and no outliers

Altough the data on the right seems statistcally clean, our classifier works on a boolean model which considers the education level in its tree generation. Thus, having an extra level to classify will ultimately result in a greater accuracy.

Now that we have our data as clean as possible we have to randomly split our data into two sections. I have splitted the data is train_features with train_labels and test_features with test_labels. My test size is ¼ of the total dataset. This was generated randomly with a random_state of 42. Let this be our ‘original data’.

## 2. Classification Process

I haved decided to use the Random Forest Classifier based on a few reasons. First reason was that in the original research, they used 5 methods to find the best prediction algorithm which was a nueral network with linear regression. Next, upon testing multiple classifiers (KNN, Percptron, NN, etc.) with dummy data, the classifier that showed promissing results was Random Forest. I wondered if I could find a way to improve the accuracy using imbalanced handling techniques. Lastly, Random Forest Classifiers are ensemble classifiers meaning it uses a GINI index when validating input [3].

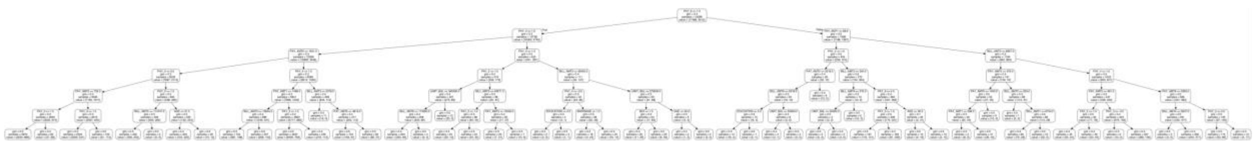


Figure 11: an example of Random Forest Tree with a depth of 6

I did not have enough time to test with the max_depth so I left the max_depth in my classifier empty which according to sk-learn’s documentation [4], means that the nodes nodes of the tree will be expanded untill all leaves are pure or until all leaves contain less than min_samples_split which was also left to none so it will make it until the last leaf is constructed. Leaving this parameter empty will also benefit the model as it will prevent any form of overfitting.

Before we start training our classifier, we must deal with the data imbalance. I used three of the most common approaches when encountering data imbalance which are Upsampling the miniority data, Downsampling the majority data, and using Synthetic Minority Over-sampling

Technique  (SMOTE) [5]. These techniques essentially get the targets to be equal to each other. My assumption would be that it works best if both targets have the same amount of data since it is binary classifier.

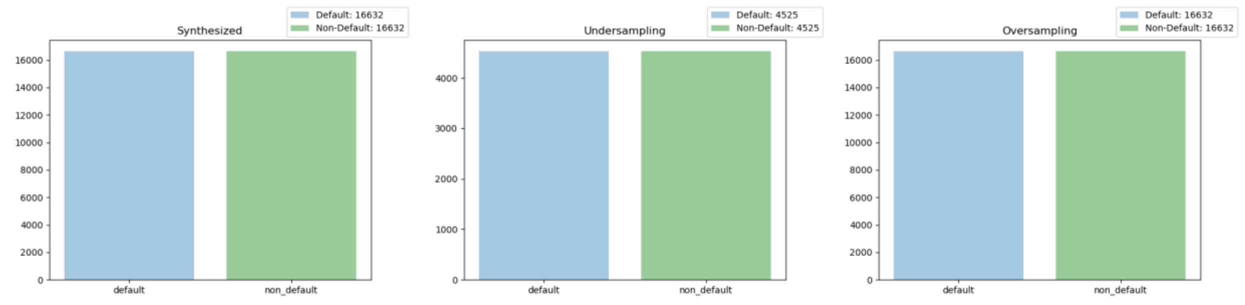


Figure 12: synthesized, undersampling, oversampling. (16632, 4525, 16632) respectively

Synthensizing data is in essense ‘Oversampling’. The main difference is that it generates new records by taking random values from the minority data and adds it to the original dataset. By using this approach we will negate the random forest classifier’s temptation to automatically assume that the test data will be non-default (because of the imbalance ~30% default ~70% non-default).

## 3. Experimentation

To conduct all my tests I decided to use python as the langauge with common libraries suchas pandas, numpy, sklearn, matplotlib, scipy, etc. It is important to note that $precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$ where TP = True Positive results, FP = False Positive results, FN = False Negative and $F1 = 2\frac{Precision*Recall}{Precision+Recall}$. Before continuing I would like to mention that for each experiment, the first test is without data imbalance techniques, hence the high percision.

### 3.1 Experiment with LIMIT_BAL, EDUCATION, and AGE

The training and test data has only 3 features. The classifer had 100 estimators and a random state of 42 to match the random_state of the imbalanced techniques. In order to see the difference between unbalanced data and balanced data I also ran the same classifier with the data pre-imbalanced techniques.

| F1 | Recall | Precision |
|---|---|---|
| 0.147 | 0.096 | 0.763 |

To no surprise we are given a high precision score but with really low recall and f1 score. What this means is that total true positive actually retrieved was very low (since the majority of the samples are ‘non-default’ ~70% the random forest tree will automatically assume ‘non-default’. This is why we do the imbalanced techniques as mentioned in the previous section).

Running the classifier with upsampled data:

| F1 | Recall | Precision |
|---|---|---|
| 0.352 | 0.533 | 0.583 |

Running the classifier with downsampled data:

| F1 | Recall | Precision |
|---|---|---|
| 0.361 | .604 | 0.546 |

Running the classifier with synthesized data:

| F1 | Recall | Precision |
|---|---|---|
| 0.342 | 0.419 | 0.657 |

As we can see upsampling and downsampling are great options when it comes to obtain a good set of data. Synthesiszing has a similar affect but with an increase in precision. This experiment shows that we can get above 50% accuracy with just taking in consideration 3 out of the 23 features. The goal is to achieve the best possible accuracy without sacrificing our recall and f1 scores to much.

### 3.2 Experiment with Normalizing LIMIT_BAL, EDUCATION, and AGE

This expirement followed the same procedure as before. The only difference is that I used the StandardScaler() function to normalize the features. What this does is puts all the values in the domain of [-1, 1]. I tested this method because of the numerical and categorical mixture of data.

Running the classifier:

Normalized data:

| F1 | Recall | Precision |
|---|---|---|
| 0.149 | 0.098 | 0.762 |

Normalized upsampled data:

| F1 | Recall | Precision |
|---|---|---|
| 0.351 | 0.532 | 0.582 |

Normalized downsampled data:

| F1 | Recall | Precision |
|---|---|---|
| 0.361 | 0.606 | 0.544 |

Normalized synthesized data:

| F1 | Recall | Precision |
|---|---|---|
| 0.373 | 0.795 | 0.434 |

Besides the Sythensized Data test, we can see similar results as the non normalized data. This indicates that normalizing the data increased the quality of data that is used to train the model, but decreased our accuracy by ~20%.

**3.3 Experiment with Normalizing and PCA of PAY, BILL_AMT, and PAY_AMT**

For this test, PCA was applied to reduce to 3 features for each segment (Pay, BILL_AMT, and PAY_AMT) aswell as normalizing them.

Running the classifier:

Normalized/PCA data:

| F1 | Recall | Precision |
|---|---|---|
| 0.153 | 0.166 | 0.611 |

Normalized/PCA Upsampled data:

| F1 | Recall | Precision |
|---|---|---|
| 0.148 | 0.143 | 0.650 |

Normalized/PCA Downsampled data:

| F1 | Recall | Precision |
|---|---|---|
| 0.189 | 0.302 | 0.450 |

Normalized/PCA Synthesized data:

| F1 | Recall | Precision |
|---|---|---|
| 0.166 | 0.182 | 0.611 |

This test is the most under performing, but we can clearly see how dimensionality reduction has an affect on recall and percision. Note that the test data used for this was normalized and pca in the same way the training data was.

**3.4 Experiment combining 3.2 and 3.3 normalized data**

I wanted to run this test to see if by combining the normalized data it would increase the percision of our classifier. I also combined the test data from 3.1 normalize and 3.2

Running the classifier:

Normalized/PCA:

| F1 | Recall | Precision |
|---|---|---|
| 0.159 | 0.155 | 0.652 |

Upsampled data:

| F1 | Recall | Precision |
|---|---|---|
| 0.142 | 0.128 | 0.671 |

Downsampled data:

| F1 | Recall | Precision |
|---|---|---|
| 0.272 | 0.408 | 0.537 |

Synthesized data:

| F1 | Recall | Precision |
|---|---|---|
| 0.224 | 0.332 | 0.526 |

This combination does not show any impressive signs. So this test failed like 3.3.

So segmentating the dataset and running the random forest classifier only guarentees at it's best 60% recall and 54% percision. This isn't a model we want use.

**3.5 Experiment upsampling, downsampling, synthesizing with 'original data'**

This test was conducted with the 'original data'. Which means that is data post statistical analysis and pre segmentation.

Running the classifier:

Original data:

| F1 | Recall | Precision |
|---|---|---|
| 0.489 | 0.386 | 0.828 |

Upsampled data:

| F1 | Recall | Precision |
|---|---|---|
| 0.518 | 0.444 | 0.824 |

Downsampled data:

| F1 | Recall | Precision |
|---|---|---|
| 0.527 | 0.654 | 0.751 |

Synthesized data:

| F1 | Recall | Precision |
|---|---|---|
| 0.506 | 0.416 | 0.827 |

## 4. Conclusion

The result of this expirement varies on which key performance indicator you are willing to accept. For the purpose of obtaining the best accuracy, then experiment 3.5's synthesized data results in 82.7% with a decent F1 and Recall percentage. If I had to conduct this study again, I would look into duplicate values and find a relation between the payment status (pay duly, 1 month over, etc), the bill amount, and how much the credit card holder paid.

## References

[1] Original research paper on this dataset

*The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients* – https://www.sciencedirect.com/science/article/pii/S0957417407006719

[2] Credit Card Default Dataset
https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[3] Random Forest Classification https://www.kdnuggets.com/2017/10/random-forests-explained.html

[4] Random Forest Classifier Python Documentation https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[5] Synthetic Minority Over-Sampling Technique (SMOTE)
https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume16/chawla02a-html/chawla2002.html