Better Default Prediction

Using cost-sensitive metrics and resampling techniques

Jogoo, Yanish Dan
2nd August 2019

[Word count: 5485]

Table of Contents

1. Abstract

This report aims to provide a better assessment of default risk by analysing, critiquing and expanding upon the study by Yeh and Lien (2009) where they compared multiple data mining methods' accuracy on predicting defaults in a dataset of Credit card clients in Taiwan.

Additional dimensions such as algorithm speed and ethical risk as well as methods and techniques to better tackle the problem of dataset imbalanced were introduced. The ROC curve and F_β-score as alternative metrics to the Lift-Chart and the SMOTE and more recent Borderline SMOTE algorithm as resampling methods are favoured.

The derivation of a reduced model of attributes is used to promote speed and minimise ethical risks. Results from the comparison of four Machine learning algorithms show that the Extreme Gradient Boosting algorithm trained on the reduced model provides the best combination of overall predictive performance and speed.

2. Introduction

Credit cards are an essential segment of the UK Banking industry with around £18.5 billion worth of transactions in January 2019 only (Lilly, 2019). In an environment where Central Banks are cutting interest rates, returns of Credit Card loans are ever more attractive to banks. This might potentially encourage some to lower their acceptance criteria.

In a gloomy Macroeconomic context with major economies experiencing slow or negative growth, wages are under pressure hindering client's ability to repay loans (Ft.com, 2019). Consequently, in 2019, the UK has seen the worst credit card default rate of the last two years and a total credit card debt of £73 billion which has sparked discussions about a credit card bubble bursting (Mail Online, 2019)

The combination of these two trends has reinforced the importance of better credit scoring and default prediction methods. A plethora of Machine Learning techniques including the increasingly popular Artificial Neural Networks (Jagielska and Jaworski, 1996) have been enlisted to attempt to model and predict default.

Yeh and Lien (2009) attempted to evaluate the performance of multiple Machine Learning algorithms on a dataset containing information about credit card clients in Taiwan to determine the best suited algorithm for the task. They compared 6 techniques and identified the Artificial Neural Network model as the best model. However, whilst their study acknowledges the presence of a class imbalance in the data, a common issue in default predictions (Zhou and Wang, 2012), they do little to address it. Furthermore, Yeh and Lien's (2009) definition of the best model ignores the associated costs, namely the cost of misclassification errors and the cost of computation (Turney, 2002).

Growing ethical concerns about credit scoring that have plagued the loan and microfinance sector (Sarker, 2013) extending to credit cards should also be addressed.

This report explores the same dataset and attempts to address the short comings of Yeh and Lien's (2009) study by introducing both well-known and newer techniques like Borderline SMOTE to overcome the class imbalance problem.

The report also argues that the Lift-Chart is ill-suited to the task and proposes alternative cost-sensitive metrics such as the ROC curve and the Fβ -score.

Ultimately, this report aims to compare four popular Machine Learning techniques and identify the optimal combination of techniques and algorithms to provide fast and accurate predictions which accommodate the needs of the bank while proposing strategies to minimise ethical risks.

3. Model evaluation

3.1 Measure of Performance

To be able to compare the multiple models, a measure of their performance must first be established.

3.1.1 Accuracy

Accuracy is a commonly used metric that measures how well the model correctly classifies the observations with respect to their true label. The equation for accuracy is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Where TP = True positive, observations correctly labelled as positive
TN = True negative, observations correctly labelled as negative
FP = False positive, negative observations wrongly labelled as positive
FN = False negative, positive observations wrongly labelled as negative

The more accurate a model is, the less it's predictions will differ from the true observations.

However, in the case of an imbalanced dataset such as the one discussed in this report, accuracy fails to provide an adequate metric. For example, a model that

simply categorises all observations as negatives will yield a high accuracy of 78% on the dataset since 78% of the data are negative observations. This relatively high accuracy score masks the poor results on the minority class. Optimising towards accuracy would only lead favouring correctly classifying the majority class at the expense of the others.

3.1.2 True Positive Rate

Accuracy's lack of penalty for incorrect predictions can be especially costly in cases where the minority class is of greater interest. For instance; defaults form the minority class of the discussed dataset yet, defaults might be more costly to the bank than incorrectly classifying a non-defaulter as a defaulter.

The accuracy of the model relative to the positive class is known as the True Positive Rate (TPR), also known as Recall score or Sensitivity. It is defined as the proportion of positive observations correctly classified as such:

$$Recall = TPR = \frac{TP}{TP + FN} \tag{3.2}$$

The TPR measures how good the classifier is at 'catching' all the positive observations. In this report's dataset hidden by a seemingly good accuracy of 78% lies an abysmal TPR of 0% (since no positive observation were correctly classified).

3.1.3 False Positive Rate

However, the TPR cannot be used in isolation because whilst correctly predicting defaults is important, a 'good' model should also be accurate in predicting non-defaulters. A measure of the accuracy relative to the negative class is the False Positive Rate (FPR), defined as:

$$FPR = \frac{FP}{TN + FP} \tag{3.3}$$

While the Precision score and FPR are similar, they behave differently, the FPR is to be minimised while the Precision score is to be maximised. Precision is defined

as the ratio of the observations correctly classified as positive to all the observations classified as positive:

$$Precision = \frac{TP}{TP + FP} \tag{3.4}$$

The classifiers operate by predicting the probabilities of an observation belonging to the positive class. Therefore, a cut-off threshold must be determined to classify probabilities above it as positive. The value of the threshold therefore determines the TPR and the FPR of a model

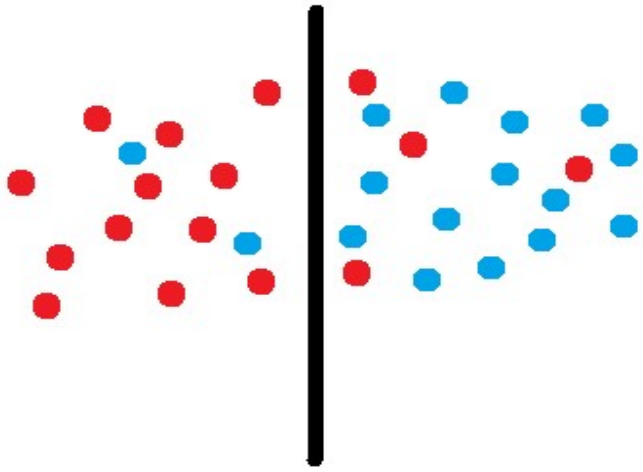

Figure 1: Illustration of a classification threshold (black line), two classes in red and blue

The better a model is at correctly predicting both classes, the clearer the difference between the predicted range of probabilities for each class and this produces a better trade-off between TPR and FPR.

3.1.4 Receiver Operating Characteristic

The Receiver Operating Characteristic (ROC) has been used in a wide variety of fields including comparing Machine Learning algorithms for many years (Fawcett,

2006). The ROC curve plots the TPR against the FPR for all thresholds, therefore representing the trade-off at different threshold values.

3.1.5. Fβ -score

Once a suitable threshold has been identified, threshold metrics can be useful to easily compare and evaluate model performances. Therefore, this report uses the F_β -Score, a cost-sensitive Threshold metric, as a complement to the ROC curve. The F_β -Score is defined as:

$$F_\beta = (1+\beta) \cdot \frac{precision \cdot recall}{(\beta \cdot precision) + recall} \tag{3.5}$$

$\beta =$ relative weights of precision and recall

In the F_β -Score, Recall is β times as important as Precision. Hence, F_β -Score can be made to be cost-sensitive. The F_β -Score can be used to provide a cost-sensitive metric both to compare the general performance of models and to analyse the cost-weighted performance at a specific one.

The F_1 -Score, a metric commonly used on imbalanced datasets (Sasaki, 2007) can be shown to be the F_β -Score where β=1 and is the harmonic mean of precision and recall. Chase Lipton, Elkan and Narayanaswamy (2014) argue that the value of the F_1 can be maximised by setting the threshold to approximately $0.5 \cdot F_1$. While the F_1 might not be useful here, it opens the possibility of the general F_β -Score exhibiting similar property, resulting in a threshold that is both close to optimal and cost-sensitive.

3.2 Advantages of ROC curves over Lift-Charts

A study by Jeni, Cohn and De La Torre (2013) evaluated the performance of both Threshold metrics and Rank metrics and found that only the ROC curve's area under the curve was not affected by skewed distributions. The area under the ROC curve (AUROC) is defined as:

(3.6) $$AUROC = \int_0^1 \frac{TPR}{FPR} \cdot dx$$

It can be shown that the AUROC is equal to the probability that the algorithm will correctly rank a randomly chosen positive observation higher than a randomly chosen negative one. The AUROC measures the model's ability to differentiate between the classes.

This report argues that in the case of the dataset used, the ROC curve and AUROC are better measures of the overall performance of predictive models than the Lift Chart approach used Yeh and Lien (2009). The advantages of the ROC curve over the Lift Chart are:

- The Lift Chart's, value depends on the chosen cut-off threshold and not solely on the predictive power of the measured model. Additionally, the trade-off between TPR and FPR is not constant for different thresholds, therefore, threshold metrics can yield misleading results (Powers, 2011). In contrast, the ROC curve is independent of a cut-off threshold.
- The Lift Chart is not cost-sensitive, it cannot take into consideration the perceived difference in cost between the misclassification of a positive sample and that of a negative sample.
  The ROC curve however, considers the TPR-FPR trade-off at multiple thresholds, allowing for the selection of a threshold that represents perceived costs. Similarly, since there is no single best threshold, the AUROC is preferred to the Youden's index which is calculated at a specific 'optimal' threshold (Yin and Tian, 2014).

4. Data Resampling

Imbalances are common among lending or credit datasets, where the number of defaulters is often a fraction of non-defaulters (Sadatrasoul, Gholamian and Shahanaghi, 2015). 'Normal' models trained on imbalanced datasets will be skewed towards the majority class because they are not cost-sensitive and operate by minimising their overall misclassification rate.

Resampling methods improve the classification on the minority class at the cost of poorer overall classification accuracy.

4.1 Undersampling

Undersampling consists of discarding observations from the majority class in an attempt to balance the ratio of classes incidentally reducing the cost of computation. However, while undersampling has been used with success in some cases (Drummond and Holte ,2003), the technique suffers from some critical drawbacks like:

- Discarding samples often implies losing potentially valuable information
- Less samples to train on leads to a classifier with a higher variance
- The removal of samples warps the distribution as the priori probability changes ($Priori\ probability = \frac{\#\ sample\ belonging\ to\ class}{Total\ \#\ of\ samples}$)

Undersampling must therefore be carried out carefully as its benefits are dependent on the initial number of samples and class distribution (Dal Pozzolo, Caelen and Bontempi, 2015).

4.2 Oversampling

Oversampling serves the same goal through the opposite way whereby samples from the minority class are duplicated. However, the duplication of existing samples leads to potential overfitting problems, this is the case even when random oversampling is used.

4.2.1. Synthetic Minority Oversampling Technique (SMOTE)

To alleviate the shortcomings of oversampling, the Synthetic Minority Oversampling Technique (SMOTE) was developed (Chawla et al., 2002). As opposed to duplicating existing samples, SMOTE aims to create new synthetic ones. This is achieved by using a K-nearest neighbours (KNN) algorithm to identify a set of K (generally set to 5) minority class neighbours of a minority class sample. Then the distance of the selected sample from N randomly chosen samples out of the K nearest identified neighbours is computed. The distances are subsequently

multiplied by a random number between 0 and 1 and added to the selected sample. This produces a new synthetic sample along the 'direction' of the samples between them.

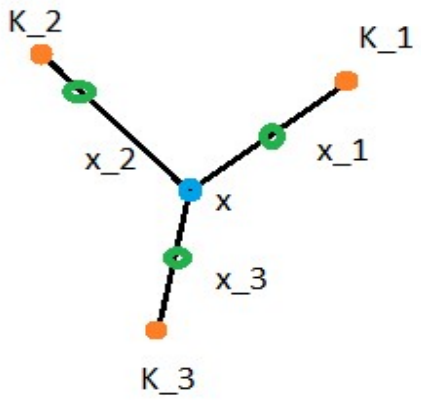

Figure 2: Creation of synthetic samples using KNN

In the figure above, x is the selected sample, K_1, K_2 and K_3 its nearest 3 neighbours of the same class and x_1, x_2 and x_3 are the synthetic point created along the directions.

4.2.2. Borderline SMOTE1

Han, Wang, and Mao (2005) based Borderline-SMOTE on the observation that most classification models try to perfectly learn the borderline of each class to be able to better differentiate them and improve predictions. Based on this observation, they argue that samples inside those borders contribute less to classification rates than the borderline samples that are more likely to be misclassified as the other class neighbouring the border. In their paper, they introduce two improved SMOTE techniques: Borderline-SMOTE1 and Borderline-SMOTE2, both of which operate by only resampling borderline samples. This report focuses on reviewing and later, implementing Borderline-SMOTE1, although Borderline-SMOTE2 is also briefly discussed.

Han, Wang, and Mao (2005) detail the operation of Borderline-SMOTE1 as follows:

$T = Training\ set$
$P = set\ of\ minority\ samples = \{p_1, p_2, \dots, p_{np}\}$
$N = set\ of\ majority\ samples = \{p_1, p_2, \dots, p_{nn}\}$

Where $np = \#\ of\ positive\ samples, nn = \#\ of\ negative\ samples$

Step 1. *Identify the nearest neighbours among both classes:*
For every minority sample p_i in the set P, a KNN algorithm is applied to identify, the m closest neighbours from the entire training set T. From those m neighbours the number of them belonging to the majority class N is labelled m', where $0 < m' < m$.

Step 2. *Identify borderline samples among neighbours:*
If $m' = m$, all the m neighbours of $p_i$belong to the majority class, therefore p_i is considered noise (a random minority sample outlier) and not brought further.

If $0 < m' < m/2$, then p_i is considered to lie inside of the border, relatively safe from misclassification and not brought further.

If $m2 < m' < m$, then p_i is said to be a borderline example, which is important to the model and at greater risk of misclassification. p_i is therefore added to a set labelled DANGER.

$$DANGER = \{p_1, p_2, \dots\}$$

Step 3. *Apply the SMOTE algorithm:*
Since DANGER contains borderline samples belonging to the minority class: $DANGER \subseteq P$. The K nearest minority sample neighbours from P of each element of DANGER are computed and the SMOTE

algorithm is used to create synthetic samples from them as detailed above.

4.2.3. Borderline SMOTE2

Borderline-SMOTE2 operates in an almost identical fashion, however at the last step the synthetic samples are created using neighbours from both P and N, that is, both classes. The distance from majority samples are multiplied by a random number between 0 and 0.5 to ensure that the synthetic sample produced remains closer to the minority class.

The figure below is presented in their paper and illustrates the mechanism of Borderline-SMOTE1:

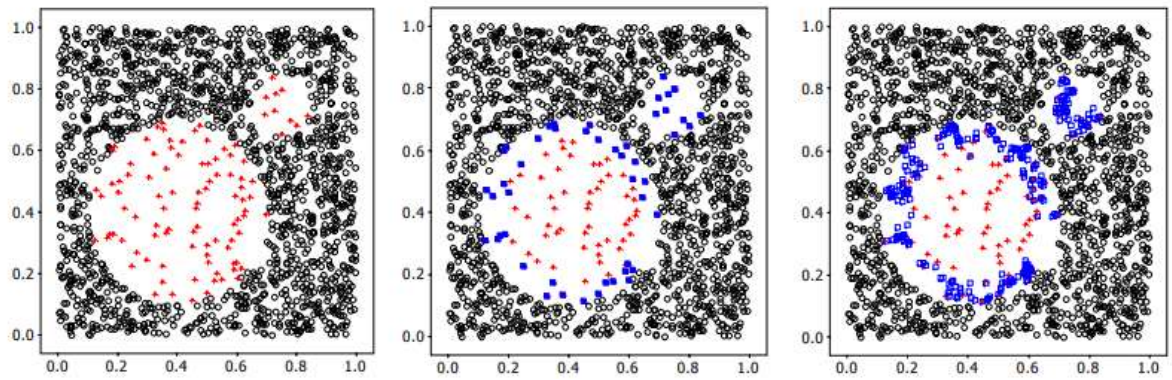

Figure 3: Example of Borderline SMOTE in 3 steps (Han, Wang, and Mao, 2005)

$$Red\ cross = Minority\ samples$$
$$Black\ circles = Majority\ samples$$
$$Solid\ Blue\ squares = Borderline\ minority\ samples$$
$$Hollow\ Blue\ Squares = Synthetic\ borderline\ minority\ samples$$

The performance of models trained on non-resampled data, data resampled using SMOTE and resampled using Borderline SMOTE1 will be evaluated and compared.

5. Exploratory Data Analysis

5.1 Data Overview

The dataset presented in this report contains information about credit card clients in Taiwan with the target variable being a binary variable where, a one (positive) represents that the client will default next month and zero (negative) representing the client not defaulting. This dataset was first presented by Yeh, I. C., & Lien, C. H. (2009) and was obtained from the UCI Machine Learning repository.
The working dataset contains 30000 observations and 24 variables presented below:

- **LIMIT_BAL**: The credit limit for the client (Integer)
- **SEX**: The gender of the client (1=male, 2=female)
- **EDUCATION:** The highest level of education achieved 1=graduate school, 2=university, 3=high school, 4=others)
- **MARRIAGE:** Marital status (1=married, 2=single, 3=others)
- **AGE**: The client's age in years (Integer)
- **PAY_1** to **PAY_6:** Client's repayment delay status from September 2005 (PAY_1) to April 2005(PAY_6) (0 = pay duly; 1 = payment delay for one month, …, 9 = payment delay for nine months and above.)
- **BILL_AMT1** to **BILL_AMT6:** The Bill Statement amount from September 2005 (BILL_AMT1) to April 2005(BILL_AMT6)
- **PAY_AMT1** to **PAY_AMT6:** Previous payment amount from September 2005 (PAY_AMT1) to April 2005(PAY_AMT6)

- **default_payment_next_month:** 1=Default, 0=No default

77.9% of the clients in the dataset did not default in the next month leading to a 1:4 ratio imbalance in the dataset. The techniques introduced previously will therefore be used to address this.

5.2 Data Exploration & Dataset Growth

Whilst all the variables were analysed, only the most pertinent are presented below:

5.2.1 Age

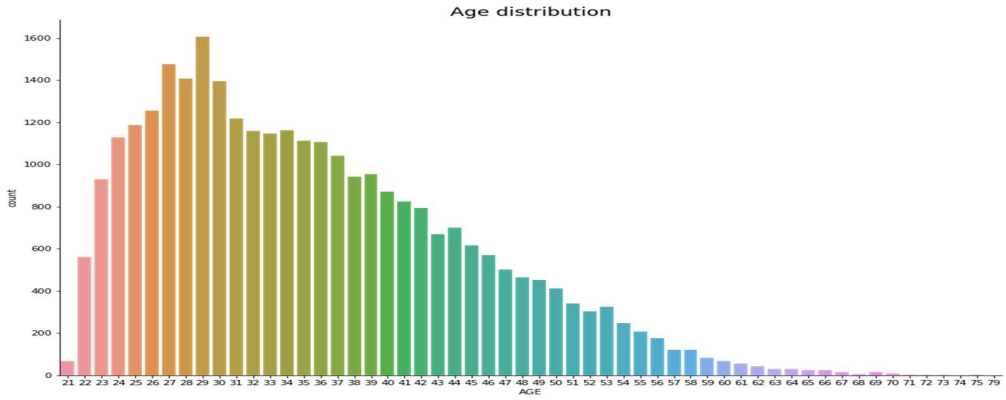

Figure 4: Age distribution

The age range of the clients in this dataset varies from 21 years old to 79 years old with a mean of 35.

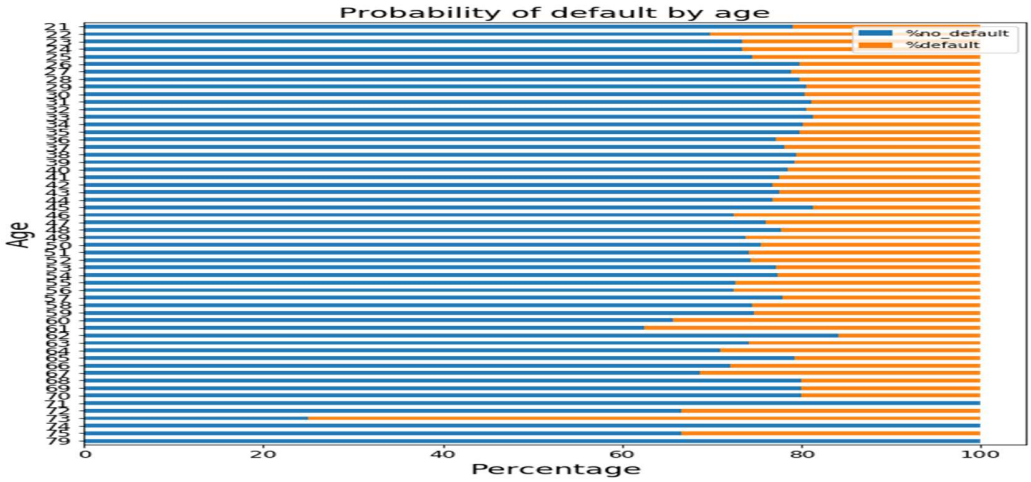

Figure 5: Probability of default by age

The probability of default seems to have a slight increasing trend with increasing age until around 65 years old.

5.2.2 Bill Statement ('BILL_AMT') and Payment Amount ('PAY_AMT')

The mean bill statement amount seems to be constant across most ages,

only varying for older clients. Simultaneously, the payment amount has a decreasing trend, again only starting to vary around 65 years old and older.

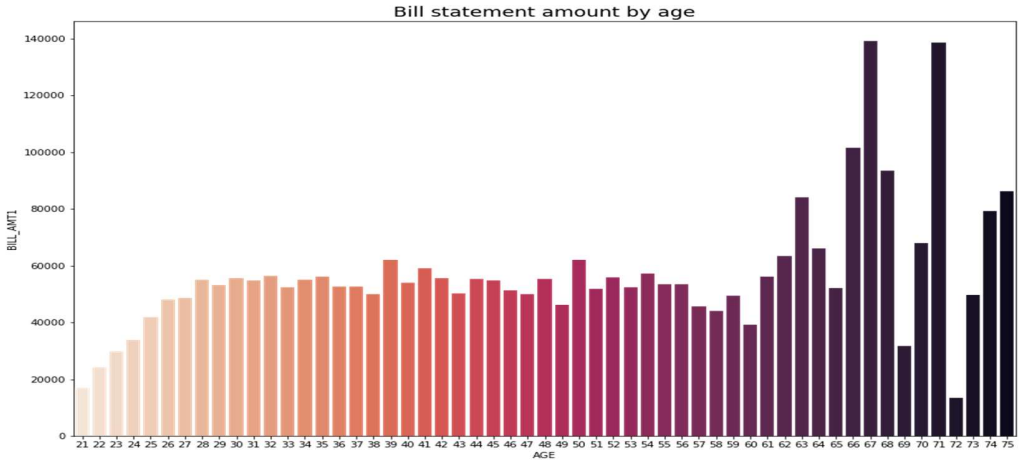

Figure 6: Bill statement amount by age

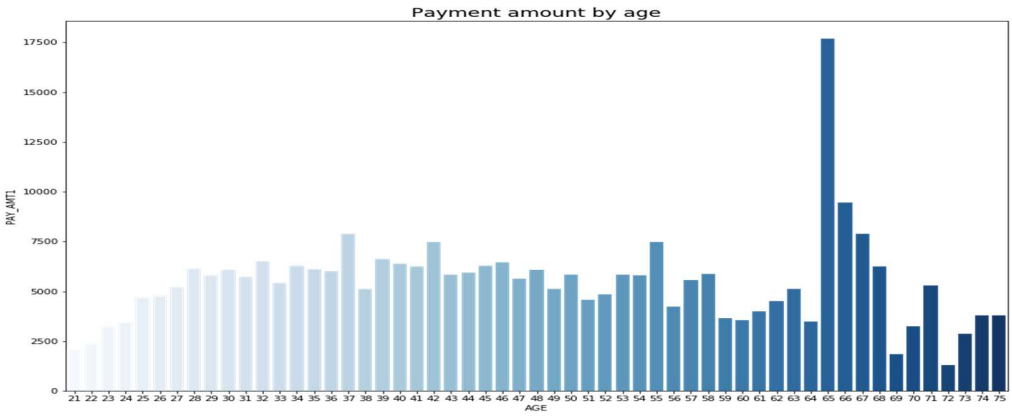

Figure 7: Payment amount by age

A possible explanation for this trend might be lower disposable income the older the clients become, due to family commitments (Economics Discussion, 2019). On the other hand, the variations observed around 65 years old and older might be due to retirement and less family commitments.

5.2.3 Gender ('SEX')

60.4% of the clients are female with 39.6% being males, however males seem to be more likely to default than female clients with a probability of default of ~5% higher. This is consistent with similar findings by Marrez, H., & Schmit, M. (2009) on loans who also found females to be less likely to default.

5.2.4 Marital Status ('MARRIAGE')

Visualising the variable denoting marital status provides some insight into the 'others' category, it has the oldest age average and a similar distribution to the 'married' category implying that it could represent the divorced category. Single clients have the lowest probability of default, followed by married clients and divorced clients respectively. This reinforces the assumption that the third category might indeed be divorced clients as findings from Lyons and Fisher (2006) indicated that divorced couples were more likely to default.

5.2.5 Repayment delay Status (PAY_)

The PAY_ variables are the most correlated (Pearson) to the probability of a client defaulting next month. The probability of default by number of months of delay is displayed below:

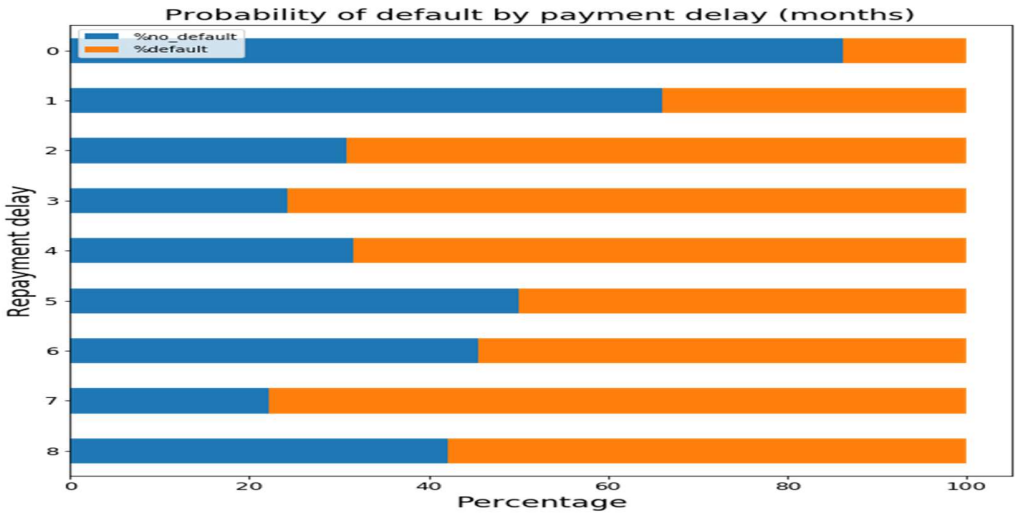

Figure 8: Probability of default by months of payment delay

As seen from the figure above, the probability of default quickly increases after a

month of delay, and after two month a client is defaults 70% of the time. The PAY_ variables are also highly correlated to each other, this is explained by each PAY_ variable containing information about the previous months' PAY_ variable by nature.

5.3 Created variable

To illustrate the dependency of a client on their credit card a new variable is created using the credit limit and bill amount (% of credit limit remaining '%REMAINING_BAL'). This variable denotes the proportion of unused or remaining credit limit, the greater it is, the less the client made use of their available credit. Analysis of this variable reveals that non-defaulters tend to use lower proportion of their limit. This is consistent with findings from Dunn and Kim (1999) who found the percentage of total credit limit used to be a significant factor affecting credit card defaults.

5.4 Data inconsistencies

Exploring the dataset revealed a set of undocumented categories and inconsistencies:

Through the newly created variable denoting unused credit limit it was revealed that for some clients (2115 observations) it can be negative and sometimes as low as -5.5, in other words a client used 650% of their credit limit in September. Even more puzzling, for some, the value can exceed 1, which upon further analysis show that 590 clients have negative bill statement amounts.

Surprisingly, despite exceeding their credit limit, 1479 of the 2115 did not default (~70%). Furthermore, 37 clients had used more than 200% of their limit. In September alone close to 10% of all observations followed this case and that those observations have a probability of 0.3 of defaulting. This implies that, while it might not truly represent exceeding the credit limit, it is probably not a random error either and represents useful information when it comes to predictive value. However, no explanation was found regarding this phenomenon and further investigation in the source of the data is required.

Deeper analysis in the negative bill statement amounts reveals that they are a product of the client overpaying in the previous month. However, analysing those overpayments revealed that while all negative bill amounts result from an overpayment, not all overpayment result in a negative bill amount. Only 17% of overpayment lead to a negative bill the next month. Furthermore, 12% of clients that overpaid the previous month still defaulted next month including 18% of those that have a negative bill amount in September. Unfortunately, the data doesn't provide enough information and further investigation at the source is required.

6. Models & Predictions

This section covers the model used for prediction/classification and supplementary models used to assist the predictive models. Most models used in this report are tuned to find the optimal hyperparameters for classification. This is achieved using a Grid Search algorithm and Cross-validation. The Pipeline function of the imblearn Python package (Imbalanced-learn.readthedocs.io, 2019) is used to apply resampling algorithms during cross validation, this is done to ensure only the training folds are resampled. Upon completion of the grid search, the model will be trained on the entire training set (with resampling if applicable) using the determined optimal hyperparameters.

6.1. Supplementary Models

6.1.1 Classification Tree (CF)

Decision trees are among the simplest and most well-known models. Starting from a set of data, the CF 'splits' the sets based on decision criterion relating to a predictor variable/feature. The feature to split on at every split is based on a principle of increasing the purity of the resulting subset/node, this is measured by the entropy criterion, or in the case of this report, the Gini index, defined as:

$$G(N) = \sum_{i=1}^{m} p_i(1 - p_i) \tag{6.1}$$

$$Where\ p_i = fraction\ of\ data\ belonging\ to\ category\ i$$

The process is repeated at each node for other features until no more features are left, all the data is correctly classified or gains from subsequent splits do not meet a specified threshold.
Classification trees are prone to overfitting, even when the tree depth is limited.

6.1.2 Lasso Regression (Lasso)

The Lasso regression is a variant of the ordinary least squares regression (OLS) that adds the inclusion of a regularisation parameter. Regularisation is a process by which constraints are added to minimise overfitting. The Lasso estimates parameters by minimising:

(6.2) $$min\left(\sum_{i=1}^{N}(Y_i - \beta_0 - \sum_{k=1}^{K} X_{ik}\beta_k)^2 + \lambda \sum_{k=1}^{K}|\beta_k|\right)$$
$$= min\left(RSS + \lambda \sum_{k=1}^{K}|\beta_k|\right)$$

$\lambda = regularisation\ parameter$
$\beta_0 = intercept$
$\beta_k = coeffecient\ of\ explanatory\ variables$
$Y = target\ variable$

As observed the term $\lambda \sum_{k=1}^{K}|\beta_k|$ represents the constraint imposed by the Lasso and λ is the regularisation parameter. The use of the of the 1-norm forces the model to promote sparsity while simultaneously attempting to minimise the residual Sum of Squares (RSS). λ determines the strength of regularisation, larger values of λ, promote more sparse models. (Tibshirani,2011).

6.1.3 K-Nearest Neighbour (KNN)

The K-Nearest neighbour algorithm is used in this report through the implementation of SMOTE and Borderline-SMOTE1 resampling methods. The K-Nearest Neighbour algorithm's mechanism involve computing the distance of a sample, generally using the Euclidean distance, to other samples in the dataset. A set of K closet neighbours (a hyperparameter to be tuned) is obtained. The probability of the sample belonging to a class is dependent on the class ratio among its K closest neighbours. (Khamis, Cheruiyot and Kimani, 2014).

6.2 Predictive Models

6.2.1 Logistic Regression (Logit)

Logistic Regression models operate on representing the probability of a sample belonging to a class by a linear function of its parameters. The regression coefficients of the logit model are the log-odds representing effect of a predictor variable on the outcome probability. Additionally, logistic regressions are relatively easy to implement and require relatively less computational power. The Logit relies on the assumptions of a linear relationship between variables and that they are independently and identically distributed (i.i.d).

6.2.2 Random Forest (RF)

Random Forest algorithms improve upon the concept of Bagging and Decision trees. Bagging algorithms (Breiman, 1996) were developed to reduce the high variance that plagues Decision tree algorithms. Bagging algorithms operate by randomly sampling data from the training set with replacement, growing decision trees on each sample and averaging the results, reducing the variance.
This relies on the property that for n independent observations with mean μ and variance σ^2, the average will be unbiased with mean μ and variance $\frac{\sigma^2}{n}$.
Unfortunately, Bagging often results in highly correlated trees for which the above property of independent observations no longer holds true.
Breiman (2001) proposes the Random Forest algorithm as an improvement to Bagging algorithms. Random Forests are different from Bagging in that at each split the trees can only choose from a random subset of variables to split on usually set to $\sqrt{p}$, where p is the number of available variables. This procedure decorrelates the trees, resulting in more accurate predictions through averaging.

6.2.2 XGBoost

Extreme Gradient Boosting (XGBoost) is a gradient tree boosting algorithm (Boosting) developed by Tianqi and Carlos (2016), which is an alternative improvement on the bagging algorithms.

Boosting operates by randomly sampling data from the training set and growing trees from them, limiting the depth of the trees to 1 or 2 splits.

The main difference is that Boosting operates iteratively, whereby initially all samples are given the same weights, however, after every iteration the weights of misclassified samples are increased and those of correctly classified ones are decreased. Subsequent samples will thus be weighted and include more of the previously misclassified samples, allowing Boosting algorithms to iteratively improve on their performance. Predictions are made by averaging the results of each tree, however, the contribution of each tree is weighted by its misclassification rate, resulting in increased accuracy.

6.2.3 Artificial Neural Network (ANN)

Artificial Neural Networks have been evolving quickly over recent years and have found great success in a variety of fields from image recognition to medical research (Khan, Wei, Ringner, Saal, Ladanyi, Westermann and Meltzer, 2001). The type of ANN used in this report is a Multilayer Perceptron Classifier (MLPC), which is a type of feedforward ANN, that is information moves in a single direction from the input nodes to the output nodes. ANNs consist of nodes which act as data processors carrying out computation on the input data and passing it on to the next layer. The MLPC fits to the training data through backpropagation and the Adam optimiser, a stochastic gradient booster. Akin to the tree boosting of the XGBoost algorithm, the result of every classification made by the MLPC is fed back through backpropagation. There the weights of the nodes in the model, which influence the node output's contribution will have on the input on the next node, are adjusted iteratively to improve the next predictions. The MLPC excels at learning the functional form of the relationship between the features without prior specification.

7. The Reduced Model

The Classification tree, Random Forest and Lasso classifier in combination with insights derived from exploratory analysis are used to produce a reduced model. The reduced model uses a subset of the explanatory variables to train predictive models on as opposed to using all the features.

The motivations to produce a reduced model are that it is less computationally intensive and that it might address possible ethical concerns by not considering

variables on which an individual could potentially be discriminated against.
The chosen features are:

1. **PAY_1:** The predictive power of PAY_1 is detailed in the analysis section, furthermore it ranks the highest in the CT, RF and Lasso selection. Only PAY_1 is selected since the PAY_ variables are highly correlated.
2. **% of Remaining credit limit 1-3:** The CT and RF rank the %REMAINING_BAL variables highly. Additionally, they encompass and condense information about the credit limit and bill amount variables.
3. **PAY_AMT 1-3:** The ability of the client to repay forms an integral part of their probability of default. This is backed by results from the RF and CT.

8. Analysis Results

8.1 ROC curve & AUROC score

The best performing model purely in terms of AUROC is the non-resampled Random Forest trained on the full model with a score of 0.788.
The second-best performing model is the RF full model trained on data resampled using Borderline SMOTE1 with an AUROC score of 0.782.
Resampling tends to improve classification on the minority class at the expense of the overall misclassification rate. However, whilst resampled models underperform slightly with regards to their non-resampled peers, the difference is negligible as illustrated by the AUROC and the ROC curve comparison between the two aforementioned RF models:

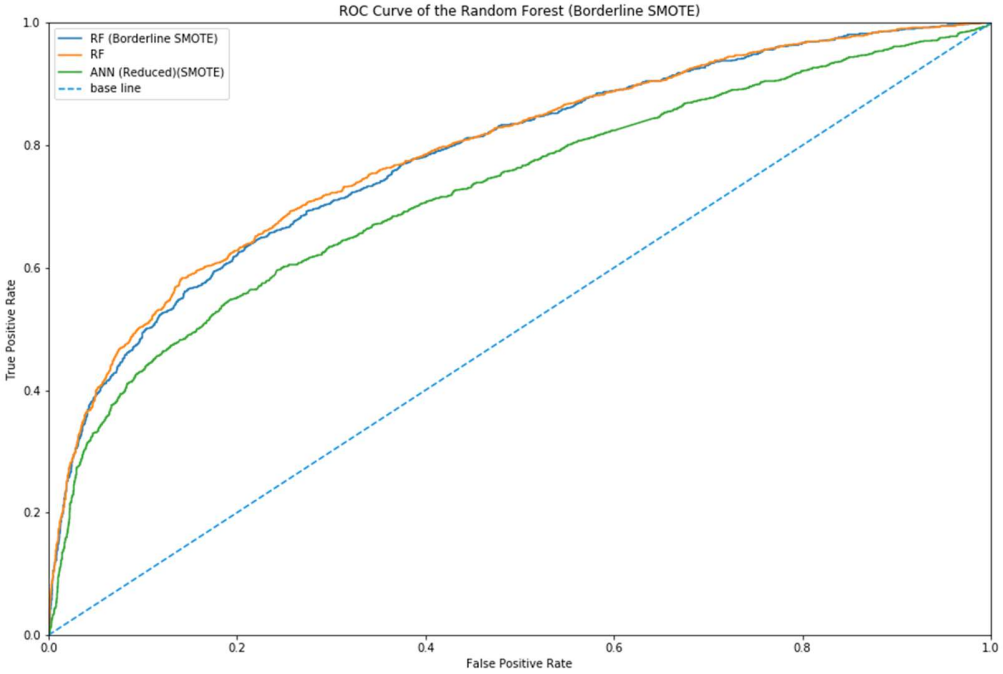

Figure 9: ROC curve of the RF, Borderline SMOTE RF and Reduced SMOTE ANN

Overall with respect to the AUROC score, there tends to be little to no difference between algorithms resampled using SMOTE and those resampled using Borderline SMOTE1.

While Yeh, I. C., & Lien, C. H. (2009) found that the Neural Network was superior in terms of accuracy, the Neural Networks presented in this report are outperformed by other models in terms of their AUROC scores.

In general, ANN models are slower, as well as models that use resampling especially Borderline SMOTE1. Consequently, the ANN trained using Borderline SMOTE1 is the slowest model by far. Conversely, XGBoost algorithms are faster and using the reduced model also promotes speed. Hence, the reduced model XGBoost is the fastest, taking mere seconds. The time taken to train the models mentioned above on 24000 observations and the AUROC result of testing them on the remaining 6000 are:

- XGBoost (Reduced): 10 s (AUROC score = 0.764)
- Random Forest (Borderline SMOTE1): 13 min (AUROC score = 0.782)
- Random Forest (Full): 14 min (AUROC score = 0.788)
- Multilayer Perceptron (Reduced) (Borderline SMOTE1): 98 min (AUROC score = 0.743)
- Multilayer Perceptron (Borderline SMOTE1): 167 min (AUROC score = 0.615)

In the context of this report the XGBoost reduced model is therefore the best overall model, boasting both rapid and high performance with a score close to that of the RF.

8.2 Scenario Analysis: Fβ – score

The F_β – score is used to analyse a scenario where the misclassification of defaulters is thrice (β=3) that of misclassifying non-defaulters. Here resampled models tend to outperform their non-resampled peers by a significant amount. Moreover, Borderline SMOTE1 tends to edge out models using the regular SMOTE algorithm overall. In this scenario the Neural Network using Borderline SMOTE1 outperforms the competition by a visible margin with an F_3-score of 0.72.

Whilst different from the findings of Yeh, I. C., & Lien, C. H. (2009) who used the Lift-Chart as a metric, these results give weight to the claim that Neural Networks are the best models for this task when the cost of default is higher than that of misclassifying non-defaulters. Higher βs favour the Borderline SMOTE1 Neural Network further.

8.3 Limitations

The results produced in this report need to be considered in the context of the present limitations. Different results can be achieved under different conditions. The limitations are listed as follows:

1. The tuning of the models' hyperparameters is computationally intensive. For this reason, not all hyperparameters were tuned and those that were, were tuned from a relatively small pool of options. The time taken to satisfactorily tune every model is dependent on the underlying hardware available.
2. The Neural Networks presented in this report are Multilayer Perceptron Classifiers. More advanced Neural Networks have been shown to achieve better results (Akkoç, 2012). However, such models are much more computationally intensive and could not be afforded in this report.
3. As discussed previously, the interest rate and other socioeconomic factors prevalent at the time have a significant impact on the default risk of clients. This dataset has limited information regarding socioeconomic factors and could be enriched to derive better results and deeper insights.

8.4 Future Research

Artificial Immune Systems (AIS) are a type of biology inspired Machine Learning algorithm and like the more popular ANNs fall under the field of Artificial intelligence. AIS aims to abstract the operation of the vertebrate immune system through mathematics, developed many years ago, like ANNs it has benefitted form advances in hardware technology.
One example of an AIS model is the negative selection model, it operates by defining a feature space where attributes from training samples are mapped. 'Detectors' are randomly placed across this feature space, then the regions containing positive observations are defined as 'normal' and detectors in those regions are removed. Consequently, the detectors left after training the model will detect samples that lie outside of the normal region when they are mapped on the feature space, classifying them as negatives. In a two-dimensional feature space, a simplified diagram would look like the figure below, where the circle are detectors and the normal region is shaded:

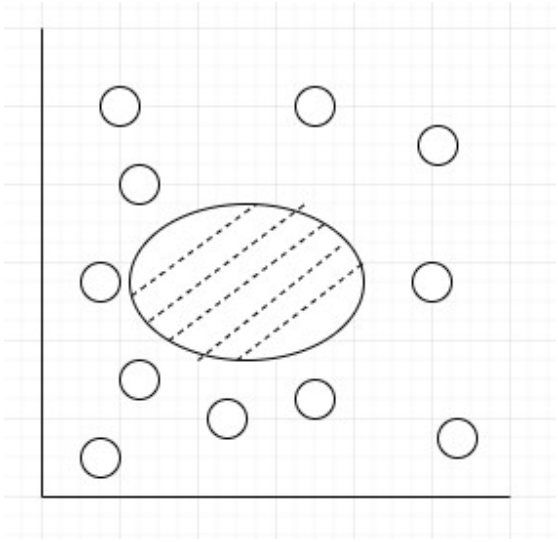

Figure 10: Negative selection, two-dimensional example

Gadi, Wang and do Lago (2008) used a Brazilian dataset to compare the performance of a few models, including AIS and ANNs in terms of predicting credit card fraud. They found that when parameters were tuned, the AIS outperformed the other algorithms. Given the similarities between credit card fraud and credit card default prediction, there is a strong potential for AIS to be applied to default prediction, which warrants future research in the topic.

9. Conclusion

In conclusion, this report sets the importance of Card default prediction in a difficult economic context. Factors affecting the default probability of clients were analysed within the confines of the presented dataset. The report subsequently presented tools and methods to improve the correct classification of defaulters, including the novel Borderline SMOTE1 technique, using appropriate metrics and the use of reduced models to quicken operations.

From a managerial point of view, the recommendations of this report are to use an XGBoost algorithm trained on a reduced model to obtain the best combination of speed and performance while minimizing ethical risks. The Borderline SMOTE1 technique especially in combination with a Neural Network algorithm could be used where timeliness is less important, for a better cost-sensitive approach.

10. References

1. Yeh, I. C., & Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.
2. Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), pp.861-874.
3. Archive.ics.uci.edu. (2019). UCI Machine Learning Repository: default of credit card clients Data Set. [online] Available at: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients [Accessed 26 Aug. 2019].
4. Gardner, M. and Mills, D. (1989). Evaluating the Likelihood of Default on Delinquent Loans. Financial Management, 18(4), p.55.
5. Marrez, H., & Schmit, M. (2009). Credit risk analysis in microcredit: How does gender matter. Université Libre de Bruxelles CEB-WP, 09-053.
6. LYONS, A. and FISHER, J. (2006). Gender Differences in Debt Repayment Problems after Divorce. Journal of Consumer Affairs, 40(2), pp.324-346.
7. Allgood, S. and Walstad, W. (2013). Financial Literacy and Credit Card Behaviors: A Cross-Sectional Analysis by Age. Numeracy, 6(2).
8. Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informe
9. Khokhlova, A. (2019). The Impact of AUC and Gini on Credit Risk Models. [online] Blog.instantor.com. Available at: https://blog.instantor.com/the-impact-of-gini-on-modern-credit-risk-models [Accessed 27 Aug. 2019].
10. Yin, J. and Tian, L. (2014). Joint confidence region estimation for area under ROC curve and Youden index. Statistics in medicine, 33(6), pp.985-1000.
11. Bahnsen, A.C., Stojanovic, A., Aouada, D. and Ottersten, B. (2013), December. Cost sensitive credit card fraud detection using Bayes minimum risk. In 2013 12th international conference on Machine Learning and applications (Vol. 1, pp. 333-338). IEEE.
12. Sasaki, Y., 2007. The truth of the F-measure. Teach Tutor mater, 1(5), pp.1-5.
13. Chase Lipton, Z., Elkan, C. and Narayanaswamy, B. (2014). Thresholding Classifiers to Maximize F1 Score. arXiv preprint arXiv:1402.1892.

14. Jeni, L.A., Cohn, J.F. and De La Torre, F. (2013), September. Facing imbalanced data--recommendations for the use of performance metrics. In 2013 Humaine association conference on affective computing and intelligent interaction (pp. 245-251). IEEE.

15. Sadatrasoul, S.M., Gholamian, M.R. and Shahanaghi, K. (2015). Extracting Rules from Imbalanced Data: The Case of Credit Scoring.

16. Drummond, C., and Holte, R. C. (2003). C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In Workshop on learning from imbalanced datasets II (Vol. 11, pp. 1-8). Washington, DC: Citeseer.

17. Turney, P. D. (2002). Types of cost in inductive concept learning. arXiv preprint cs/0212034.

18. Dal Pozzolo, A., Caelen, O., & Bontempi, G. (2015). When is undersampling effective in unbalanced classification tasks?. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 200-215). Springer, Cham.

19. Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. In 2015 IEEE Symposium Series on Computational Intelligence (pp. 159-166). IEEE.

20. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over–sampling technique. Journal of Artificial Intelligent Research,16, 321–357.

21. Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In International conference on intelligent computing (pp. 878-887). Springer, Berlin, Heidelberg.

22. Imbalanced-learn.readthedocs.io. (2019). imblearn.pipeline.Pipeline — imbalanced-learn 0.5.0 documentation. [online] Available at: https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.pipeline.Pipeline.html [Accessed 30 Aug. 2019].

23. Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73(3), 273-282.

24. Khamis, H. S., Cheruiyot, K. W., & Kimani, S. (2014). Application of k-nearest neighbour classification in medical data mining. International Journal of Information and Communication Technology Research, 4(4).
25. Shahinfar, S., Guenther, J. N., Page, C. D., Kalantari, A. S., Cabrera, V. E., Fricke, P. M., & Weigel, K. A. (2015). Optimization of reproductive management programs using Machine Learning analysis and cost-sensitive evaluation of classification errors. Journal of dairy science, 98(6), 3717-3728.
26. Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123-140.
27. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.
28. Chen, Tianqi; Guestrin, Carlos (2016). "XGBoost: A Scalable Tree Boosting System". In Krishnapuram, Balaji; Shah, Mohak; Smola, Alexander J.; Aggarwal, Charu C.; Shen, Dou; Rastogi, Rajeev (eds.). Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. ACM. pp. 785–794
29. Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., ... & Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial Neural Networks. Nature medicine, 7(6), 673.
30. Zurada, J. M. (1992). Introduction to artificial neural systems (Vol. 8). St. Paul: West publishing company.
31. Dunn, L. F. & Kim, T. (1999). An Empirical Investigation of Credit Card Default. Working Paper, Ohio State University, Department of Economics, 99-13.
32. Lilly, C. (2019). Statistics on UK household credit card usage | May 2019 | finder UK. [online] Finder UK. Available at: https://www.finder.com/uk/credit-card-statistics [Accessed 1 Sep. 2019].
33. Calem, P. S., & Mester, L. J. (1995). Consumer behavior and the stickiness of credit-card interest rates. The American Economic Review, 85(5), 1327-1336.
34. Nytimes.com. (2019). What Is Austerity and How Has It Affected British Society?. [online] Available at: https://www.nytimes.com/2019/02/24/world/europe/britain-austerity-may-budget.html [Accessed 1 Sep. 2019].

35. Mail Online. (2019). Is Britain's £73billion credit card bubble about to burst?. [online] Available at: https://www.dailymail.co.uk/news/article-6937907/Is-Britains-73billion-credit-card-bubble-burst.html [Accessed 1 Sep. 2019].

36. The Money Charity. (2019). The Money Statistics May 2019 - Quarter Century Credit Card. [online] Available at: https://themoneycharity.org.uk/money-stats-may-2019-average-uk-credit-card-debt-take-quarter-century-repay/ [Accessed 1 Sep. 2019].

37. Ft.com. (2019). Credit card interest rates hit UK consumers | Financial Times. [online] Available at: https://www.ft.com/content/97493ef4-bf25-11e7-b8a3-38a6e068f464 [Accessed 1 Sep. 2019].

38. Jagielska, I., & Jaworski, J. (1996). Neural Network for predicting the performance of credit card accounts. Computational Economics, 9(1), 77-82.

39. Zhou, L., & Wang, H. (2012). Loan default prediction on large imbalanced data using Random Forests. TELKOMNIKA Indonesian Journal of Electrical Engineering, 10(6), 1519-1525.

40. Sarker, D. (2013). Pressure on loan officers in microfinance institutions: An ethical perspective. Journal of Economics and Sustainable Development, 4(12), 84-88.

41. Gadi, M. F. A., Wang, X., & do Lago, A. P. (2008). Credit card fraud detection with artificial immune system. In International Conference on Artificial Immune Systems (pp. 119-131). Springer, Berlin, Heidelberg.

42. BBC News. (2019). Economy in double-dip recession. [online] Available at: https://www.bbc.co.uk/news/business-17836624#targetText=Shoppers%20say%20they%20are%20struggling%20to%20buy%20goods%20and%20pay%20bills&targetText=The%20UK%20economy%20has%20returned,first%20three%20months%20of%202012.&targetText=A%20recession%20is%20defined%20as,the%20fourth%20quarter%20of%202011. [Accessed 2 Sep. 2019].

43. Economics Discussion. (2019). The Life-Cycle Theory of Consumption (With Diagram). [online] Available at: http://www.economicsdiscussion.net/consumption-function/the-life-cycle-theory-of-consumption-with-diagram/14495 [Accessed 2 Sep. 2019].

44. Akkoç, S. (2012). An empirical comparison of conventional techniques, neural networks and the three stage hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) model for credit scoring analysis: The case of Turkish credit card data. European Journal of Operational Research, 222(1), 168-178.