University College Dublin

School Of Mathematics and Statistics

COMPARISON AND EVALUATION OF DIFFERENT MACHINE LEARNING METHODS AT PREDICTING CREDIT CARD DEFAULT

By

Sidney Harshman-Earley
Denis O'Riordan

in partial fulfilment of ACM40960 -
Projects in Maths Modelling required
for the completion of the degree
MSc in Data and Computational Sciences

Principal Supervisor: Dr Sarp Akcay
Programme Coordinator: Assoc. Professor Lennon Ó Náraigh

July 2022

We hereby certify that the submitted work is our own work, was completed while registered as candidates for the degree stated on the Title Page, and we have not obtained a degree elsewhere on the basis of the research presented in this submitted work.

—

Abstract

no more than 250 words for the abstract - a description of the research question/knowledge gap what we know and what we don't know
- how your research has attempted to fill this gap
- a brief description of the methods
- brief results
- key conclusions that put the research into a larger context

Research into utilising ML methods to predict credit card default has largely been limited to one historic data set. As part of an applied data science competition organised by the American Express credit card company, an industrial-sized dataset was released which in turn provided the foundation to further research in this field. A subset of the data was used and feature engineered to create a dataset of 458'913 individual customer observations and 129 features, with masked feature labels. Exploratory data analysis found the P_2 variable, in particular, displayed a strong positive correlation with the target variable - defaulting on credit card payment. Four ML classification models were trained, validated and tested on independent data to predict the target - logistic regression, logistic regression (P_2 variable only), Random Forrest and Artificial Neural Networks.

Contents

List of Figures

List of Tables

1 Introduction

A credit card is a financial instrument issued by banks and other financial institutions with a pre-set credit limit allowing customers to make cashless transactions. Each month or at different intervals, the credit card provider issues a statement with details of spending history, the interest charged, balance, and payment deadlines. As of May 2022, there were 1.5 million active credit cards in circulation in Ireland, with daily spending exceeding €1 million.[1]. Further to that, according to research conducted by the Central Statistics Office in 2018, 12.7% of all households have credit card debt, this figure reaches 21.0% for households with two adults and one to three children under 18.[2]

A default event concerning a credit card occurs if a customer does not pay the amount due within 120 days of their latest statement date. However, the deadline can depend on the issuing institution. The most significant risk to lenders is a large and unexpected number of customers failing to meet their credit repayment obligations. This is known as credit default risk. Credit default prediction is central to managing risk in a consumer lending business. Financial institutions are legally obligated to create models that predict credit defaults. Credit card default prediction is based on historical and underlying information of credit card customers. The use of models to predict and analyse credit card customer default behaviour is a typical classification problem. These models advise the bank's capital requirements, which ensure that the bank can absorb losses brought about by a significant increase in credit defaults. Credit default prediction also allows lenders to optimise lending decisions, which leads to a better customer experience and sound business economics. Current models exist to help manage risk, but is it possible to create better models that outperform those currently in use? Given that banking portfolios can be worth billions, even marginal improvements in preventing and reducing defaults can be considered significant.

The big data paradigm has revolutionised the banking industry, changing how financial institutions operate. Developments in technology have revolutionized the customer experience and granted access to an ever-growing volume of structured and unstructured information. Institutions can leverage this new information in many ways, including predicting credit default. There have been advances in Machine Learning (ML) techniques such as representation learning methods e.g. Neural Networks, ensemble learning methods such as Random Forest. These models have yet to be widely deployed because of their opaque nature i.e. Black-Box methods. From a regulation perspective, transparency and explainability are necessary when deploying credit default prediction models in practice. However, they may still provide value to financial institutions and may in the future be accepted as valid, clear and sound methods for predicting credit card default.

The aim of this project is to apply a range of machine learning techniques to predict credit card default using the historical data of credit card customers. The following report describes the process undertaken to compare a number of models. Beginning with an industrial size dataset, cleansing and formatting the data before using it to train, tune and evaluate the chosen models based on the historical data of credit card customers.

[1] https://www.centralbank.ie/statistics/data-and-analysis/credit-and-debit-card-statistics/daily-credit-and-debit-card-statistics

[2] https://www.cso.ie/en/releasesandpublications/ep/p-hfcs/householdfinanceandconsumptionsurvey2018/debtandcredit/

2 Review of Literature

There have been numerous articles and papers within the scope of using ML methods to predict credit default, this includes the prediction for credit card default.

Studies examining credit card default have been concentrated, mainly using data originally used as part of Yeh & hui Lien (2009). This data is currently freely available as the Default credit card clients Data Set on the UC Irvine Machine Learning Repository.[3] This data, collected in October 2005 is from a cash and credit card issuing bank in Taiwan. Among the 30,000 observations, 22.12% are the cardholders who defaulted on payment. This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This data uses 23 explanatory variables, including a mix of personal information(age, gender, marital status and education level), amount of credit given, historical bill statements, and historical payment information.

Yeh & hui Lien (2009), compared six classification algorithms - K-nearest neighbour, Logistic regression, Discriminant analysis Naïve Bayesian classifier, Artificial Neural Networks, Classification Trees. In the classification accuracy, the results show that there were few differences in error amongst the six methods. The generated probability of default by the Artificial Neural Network most closely resembled the actual probability of default which was estimated using a novel "Sorting Smoothing Method".

Other research on predicting credit card default in subsequent years utilised this data for training and evaluating models. The following is a sample of articles available, applying a wide variety of Machine Learning methods to this classification problem. Neema & Soibam (2017), took a similar approach in the choice of methods but attempted to predict the best possible cost-effective outcome from the risk management perspective. Naive Bayes Estimator and Random Forest classifiers were introduced. This was evaluated using a cost function which gave a higher cost to defaulters classified not correctly as they are the minority in the data. It was concluded that original data with the Random Forest algorithm is the best in terms of a good balance on cost versus accuracy.

Yang & Zhang (2018) introduces two new methods used to predict credit card default. Support Vector Machine (SVM) involves using a kernel function to map the predictor data into a high-dimensional feature space where the outcome classes are easily separable. XGBoost and LightGBM, forms of gradient boosted trees algorithm were used. LightGBM and XGBoost were both deemed to have the best performance in the prediction of categorical response variables.

While other studies have used Neural Networks in predicting credit card defaults, the models used have been vague, and little detail has been given on the architecture or tuning of the model. Chou & Lo (2018) trialled a range of Networks, experimenting with two to five layers with the number of processing units of 64, 32 and 16 units. Neural Networks with three layers and 64 units recorded the highest accuracy of all configurations.

Due to a lack of credit card-specific data pertaining to defaulting on payment, all available studies which predict credit card default utilise the Taiwan data, which is both region specific, over 15 years old and obtained at a time when credit card issuers in Taiwan faced a credit card debt crisis.[4]

[3] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[4] https://sevenpillarsinstitute.org/case-studies/taiwans-credit-card-crisis/

3 Data

3.1 Underlying Data

Data used for this project was obtained from the Kaggle website, an online platform and community for data scientists.[5] One of Kaggles main features is it's competition platform.[6] Kaggle allows users to organise and host competitions, these range from commercially-purposed prediction problems to more experimental research competitions. They give entrants the opportunity to test and grow their data science skills while competing for prizes and gives hosts an outlet to tackle tough business problems by turning them into a competition and allowing Kaggles user base of 10 million to provide potential solutions.

Access to the data required registering with Kaggle and joining the American Express - Default Prediction competition.[7] American Express is a globally integrated payments company and the largest payment card issuer in the world.[8] This competition provides an industrial scale data set to build a machine learning model to predict credit card default using time-series behavioural data and anonymised customer profile information over the period of March 2017 to March 2018. The target binary variable i.e. default is calculated by observing 18 months performance window after the latest credit card statement, and if the customer does not pay due amount in 120 days after their latest statement date it is considered a default event.

3.2 Data Structure

train_labels.csv
A list of unique customer identifiers `customer_ID` with the target label `target` indicating a default event with `target = 1` indicating a default, `target = 0` indicating no default. There are 458913 observations in this data meaning 458913 unique AMEX customers. 74.1% of customers in the data did not default on payment while 25.9% defaulted.

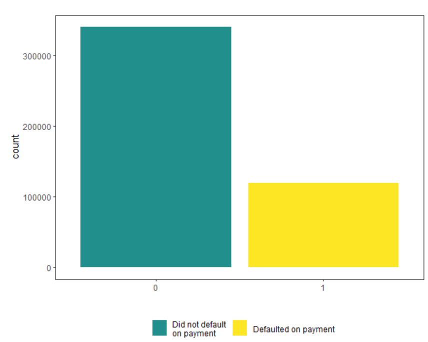

Figure 1: Breakdown of defaults in training data

train_data.csv

[5] https://www.kaggle.com/getting-started/44916
[6] https://www.kaggle.com/docs/competitions
[7] https://www.kaggle.com/competitions/amex-default-prediction/data
[8] https://about.americanexpress.com/our-company/our-business/our-business/default.aspx

This is a tabular data set corresponding to the training data, consisting of 190 features. It contains multiple statement dates per customer_ID, each observation corresponds to a monthly statement for a customer. The data is ordered firstly by `customer_ID`, corresponding with the train_labels.csv file and secondly ordered chronologically by statement date. The dataset contains aggregated profile features for each customer at each statement date. There are 177 numeric features which are anonymised and normalized, and fall into the following general categories:

Table 1: Variable Categories

Variable Name	Category
D_*	Delinquency variables
S_*	Spend variables
P_*	Payment variables
B_*	Balance variables
R_*	Risk variables

There are 11 categorical variables - B_30, B_38, D_114, D_116, D_117, D_120, D_126, D_63, D_64, D_66 and D_68.

The remaining two unaccounted for features are the variable S_2 which contains the date of the statement and `customer_ID` which can used to match to the target label.

Table 2: First five rows and columns from train_data.csv

customer_ID	S_2	P_2	D_39
0000099d6bd597052cdcda90ffabf56573fe9d7c79be5fbac11a8ed792feb62a	2017-03-09	0.9384687	0.001733339
0000099d6bd597052cdcda90ffabf56573fe9d7c79be5fbac11a8ed792feb62a	2017-04-07	0.9366646	0.005775443
0000099d6bd597052cdcda90ffabf56573fe9d7c79be5fbac11a8ed792feb62a	2017-05-28	0.9541803	0.091505397
0000099d6bd597052cdcda90ffabf56573fe9d7c79be5fbac11a8ed792feb62a	2017-06-13	0.9603836	0.002455224
0000099d6bd597052cdcda90ffabf56573fe9d7c79be5fbac11a8ed792feb62a	2017-07-16	0.9472484	0.002483014

Other Data

The competition data also includes two other files but are not within the scope of this project. test_data.csv is test data used to evaluate entries in the competition. However, labels for this data are not included an the evaluation for the models performance on this data is limited to the competitions own evaluation metric.[9] sample_submission.csv is a sample submission file for the competition in the correct format.

[9] https://www.kaggle.com/competitions/amex-default-prediction/overview/evaluation

3.3 Data Engineering

Brief bit on load, chunking and caching process?

3.4 Data Cleaning

- Remove NA columns,
- remove low variance columns
- remove highly correlated columns
- noise removal
- merge label and data
- save as parquet Anything else?

4 Methods

4.1 Models

4.1.1 Logistic Regression (LR)

Logistic regression is one of the most simple classification algorithms and is used to model a binary categorical variable given a collection of numerical and/or categorical predictors, Harrell (2015). The model applies the logistic function to the linear regression model which is used to model the probability of a binary event occurring.

$$p_i = Pr(Y_i = 1|x_i) = \frac{exp(w_0 + x_i^T w)}{1 + exp(w_0 + x_i^T w)}$$

4.1.2 Random Forest (RF)

Random Forests, Breiman (2001), are a very powerful ensemble classification method. Ensemble learning is an aggregation of predictions made by multiple classifiers with the goal of improving accuracy. The method uses Classification Trees and bootstrapping extensively. A random forest is, effectively, a random collection of Classification Trees estimated on random subsets of the data.
A Classification Tree is an iterative process of splitting the data into partitions based on values of the observations, and then splitting it up further on each of the branches. The classifier is trained in order to produce pure groups of observations or 'buckets', by minimising the entropy or spread of the target variable in each bucket. The majority value of the target variable in a bucket is then used for predictions. The main disadvantage is that Classification Trees suffer from over-fitting and bias and using a single Classification Tree would present an over simplistic model.
In a Random Forest, each Classification Tree is grown independently, and each individual Classification Tree has an equal vote as to what the outcome is. Random forests provide an improvement by means of a small tweak that reduces the dependence among the trees. The main idea is to use only a random subset of the predictor variables at each split of the classification tree fitting step, this is generally taken as $\sqrt{N}$ where N is the number of features but can be specified by the user. If we build classifiers on subsets of the variables, then they will behave more independently than if we build them on all of the data. This increases diversity and averaging results across independent classifiers will be more stable than averaging results on dependent ones. The main parameter of the Random Forests to be tuned is the number of variables used at each split and is tuned to prevent overfitting and improve efficiency.

4.1.3 Deep Neural Network (DNN)

Neural networks are machine learning methods that simulate the biological learning mechanism of the brain (Goodfellow et al. (2016), Zhang et al. (2021)). Neural networks allow to learn representation features encompassing complex relations between inputs and output. Representation learning is the use of Machine Learning to not only learn the mapping from representation features to output, but also to learn the representation itself. Deep Learning methods introduce representations that are expressed in terms of other simpler representations, thus enabling to derive complex concepts out of simpler concepts. In practice, a neural network is a series of layers, made up of nodes with subsequent layers separated by activation functions.

4.2 Training/Validation

The data is first randomly split 80:20 into training/validation and test sets. For the LR and RF models, optimally tuned models are obtained using a three-fold cross-validation method over ten replications.

4.3 Evaluation

The models are compared on a number of metrics based on the correspondence of predicted labels for the test data versus observed labels of the test data.

		Observed Value	
		0	1
Predicted Value	0	True Negative (TN)	False Negative (FN)
	1	False Positive (FP)	True Positive (TP)

Figure 2: Cross tabulations between predicted and actual outcomes

Sensitivity, otherwise known as the True Positive Rate (TPR), quantifies the proportion those who defaulted were predicted to have defaulted. It is calculated by TP/(TP + FN). Specificity, otherwise known as the True Negative Rate (TNR) quantifies the proportion those who did not default were were predicted to not have defaulted. It is calculated by TN/(TN + FP). Accuracy is the overall classification accuracy of a results set and is given by (TP + TN)/(TP + TN + FP + FN).

As the binary classification models can produce their predictions in the form of probabilities, varying the classification threshold - probability threshold to which default label is applied - affects the predictions and therefore the performance of each model. Further evaluation and comparison can be performed by comparing the performance of a classification model at all classification thresholds. The receiver operating characteristic - ROC curve plots Sensitivity versus 1 - Specificity over. The curve illustrates the diagnostic ability of a binary classifier as the classification threshold is varied. The ROC determines how often will a randomly chosen 1 outcome have a higher probability of being predicted to be a 1 outcome than a randomly chosen true 0. The larger the area under the curve - AUC, the better is the discrimination. A perfect classifier would have AUC = 1. A classifier not better than random guessing would have AUC = 0.5 and the corresponding ROC curve would resemble a plot of the identity function ($y = x$).[10]

A fourth metric was also used to compare models. The evaluation metric used for the Kaggle competition [11] is the mean of two measures of rank ordering - Normalized Gini Coefficient,G, and default rate captured at 4% , D. $M = 0.5 \times (G + D)$. G is an extension of the AUC and is

[10] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
[11] https://www.kaggle.com/competitions/amex-default-prediction/overview/evaluation

calculated as $G = 2 \times AUC - 1$. The Default rate captured at 4% is the Sensitivity for a classification threshold set at 4% of weighted total sample count.

5 Results

5.1 EDA

5.2 Final Models

The training and validation process yielded the following four models which were used to predict the test observations.

- Logistic regression model of all features using a classification threshold of 0.223. This will be referred to as the "full LR" model.
- Logistic regression model using P__2 feature as the only predictor variable using a threshold of 0.208. This will be referred to as the "reduced LR" or "simple LR" model.
- The optimal Random Forest model best on repeated cross-validation was found to use a random sample of 20 variables at each split and generating 100 trees.
- Brief format of Final NN

5.3 Model Comparrison

A full table of result metrics can be seen in Appendix A.

In terms of accuracy, NN and RF both slightly outperform LR and achieve an accuracy that is 0.07 greater than the accuracy produced by the reduced LR model. That pattern of results is repeated in the Sensitivity. Of all the customers who defaulted in the test set, the Neural Network and Random Forest correctly identified over 90% of them. The full LR model correctly identified 85% while the simple LR identified under 80%. The full LR model achieves the highest specificity on the test data out of the four models with over 0.91 recorded. The simple LR model records the second highest specificity with 0.89 while both the Neural and Random Forrest recorded specificities of 0.78.

As the Sensitivity has a positive effect in the calculation of the AMEX metric, it is only logical that NN and RF would fare better than the two LR models. With a value fo 0.71, the NN model is the only one to record a AMEX metric greater than 0.7. The RF model records an metric value of 0.69 while the full and simple LR models achieved values of 0.61 and 0.58 respectively.

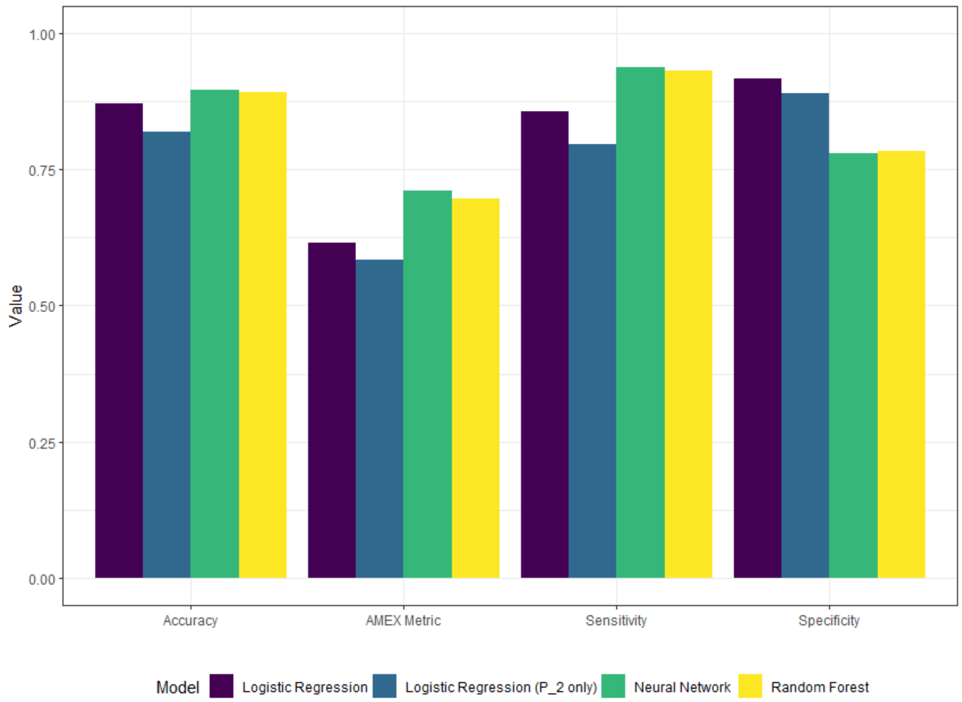

Figure 3: Bar Chart of Results by Model

On initial inspection, the four ROC curves produced by the model appear quite, however the ROC curves for the simple LR model and the RF model don't extend as close to the top-left of the plot as much as the full LR and the NN models. This is confirmed when comparing the Areas Under the Curve (AUC). The full LR and NN models recorded an AUC of over 0.95 while the other two models only recorded AUC's of 0.92.

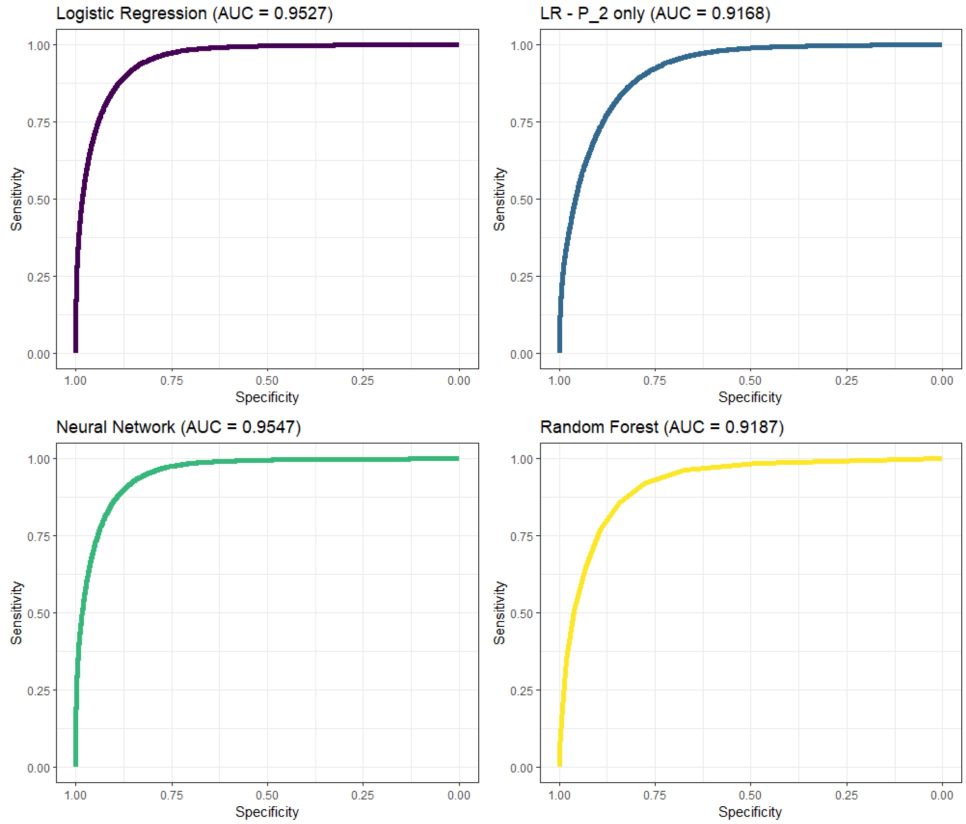

Figure 4: Receiver Operating Characteristic (ROC) curve for each Model.

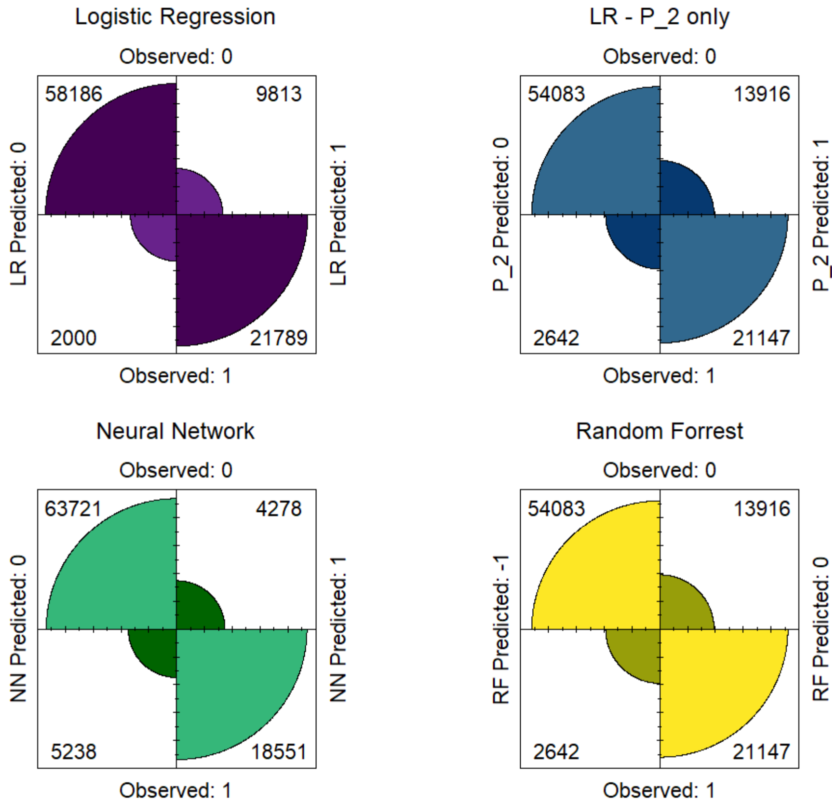

Figure 5: FourFold Plot of Test Data Results

6 Discussion

6.1 Context of Results

Compare to previous studies

Only slight improvements gained by advanced ML methods over simple LR. A significant takeaway from the results is the closeness in results between a relatively basic logistic regression model and the more advanced and intensive Neural Networks and Random Forest models. Accuracy was only improved by 0.08

Arguably, a more interesting revelation is the closeness in results between the simple logistic regression model, utilising only one variable and the full regression model which had an extra 128 predictor variables.

Only slight improvements gained by full LR methods over reduced LR.

6.2 Limitations Future Research

Computational limitations Only able to run computationally heavy algorithms such as RF on very small portions of the data Unable to harness the benefits of the time series take RNN paragraph from lit review lag features

More advanced forms of Neural Networks have also been applied to the problem, Hsu et al. (2019) proposed using a Recurrent Neural Network (RNN) feature extractor with a Gated Recurrent Unit (GRU) on credit card payment history to leverage the time dependencies embedded in these dynamic features of historical credit card data. Recurrent Neural Networks (RNN) are specifically designed to use recursive architecture to extract patterns from input sequences (Goodfellow et al. 2016). They have been proven helpful in applications that heavily rely on time-variant features. As a result, it is natural to consider RNN models as feature extractors for customer behaviour that often appear as sequences in financial data.

One of the main obstacles for contextualising the results is the lack of information of the variables included as part of the data. The only detail of the features was the category to which it belonged.[12] Due to this, little inference can be made on the variables and the main outcome of this study is the performance of the models. Particularly it would be difficult to comment on the Delinquency and Risk variables as there is no further information to what they constitute and is impossible to infer any further information from them.

The choice of models employed for this project was relatively arbitrary. A selection of three well known models which were seen to perform this task in previous studies using the Taiwan data were chosen. One simple method, one ensemble learning and one deep learning model were comparatively evaluated. This unfortunately does not allow us to benchmark the performance against internal models used by AMEX or other banks. This benchmark would allow analysis on whether any of the models employed in this study could provide value and risk mitigation. A small sample of potential models were used in this project. Models used in studies on other data that may improve performance independently or in tandem to other

[12] Variable Categories table

models. include K-Nearest Neighbour,Linear Discriminant Analysis,Logistic Regression, XGBoost and LightGBM.

Other methods of evaluating and comparing the models could be employed in future. Neema & Soibam (2017) used a cost function when comparing methods to predict customers credit card default. A higher penalty or cost was given to defaulters classified incorrectly. A range of cost factors were used to identify a model with the best accuracy in predicting a defaulter in a cost- effective manner.

7 Conclusion

This extends model choice to include black-box models that are better able to identify the non-linear relationships between the target variable and the predictor variables. This means that banks and lenders have the potential to use models which could help them reduce expenses brought about by loan defaults and/or identify and avoid lending opportunities that are too risky.

The conclusion is intended to help the reader understand why your research should matter to them after they have finished reading the paper. A conclusion is not merely a summary of your points or a re-statement of your research problem but a synthesis of key points. For most essays, one well-developed paragraph is sufficient for a conclusion, although in some cases, a two-or-three paragraph conclusion may be required.

- *Presenting the last word on the issues you raised in your paper. Just as the introduction gives a first impression to your reader, the conclusion offers a chance to leave a lasting impression. Do this, for example, by highlighting key points in your analysis or findings.*
- *Summarizing your thoughts and conveying the larger implications of your study. The conclusion is an opportunity to succinctly answer the "so what?" question by placing the study within the context of past research about the topic you've investigated.*
- *Demonstrating the importance of your ideas. Don't be shy. The conclusion offers you a chance to elaborate on the significance of your findings.*

8 Bibliography

Breiman, L. (2001), 'Random forests', *Mach. Learn.* **45**(1), 5–32.
URL: *https://doi.org/10.1023/A:1010933404324*

Chou, T. & Lo, M. (2018), 'Predicting credit card defaults with deep learning and other machine learning models', *International Journal of Computer Theory and Engineering* **10**, 105–110.

Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep Learning*, MIT Press. http://www.deeplearningbook.org.

Harrell, F. E. (2015), *Regression modeling strategies : with applications to linear models, logistic and ordinal regression, and survival analysis / Frank E. Harrell, Jr.*, Springer series in statistics, second edition. edn, Springer, Cham.

Hsu, T.-C., Liou, S.-T., Wang, Y.-P., Huang, Y.-S. & Che-Lin (2019), Enhanced recurrent neural network for combining static and dynamic features for credit card default prediction, *in* 'ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', pp. 1572–1576.

Neema, S. & Soibam, B. (2017), The comparison of machine learning methods to achieve most cost-effective prediction for credit card default.

Yang, S. & Zhang, H. (2018), 'Comparison of several data mining methods in credit card default prediction', *Intell. Inf. Manag.* **10**(05), 115–122.

Yeh, I.-C. & hui Lien, C. (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, Part 1), 2473–2480.
URL: *https://www.sciencedirect.com/science/article/pii/S0957417407006719*

Zhang, A., Lipton, Z. C., Li, M. & Smola, A. J. (2021), 'Dive into deep learning', *arXiv preprint arXiv:2106.11342* .

9 Appendices

9.1 Appendix A: additional tables

Table 3: Test data results

Model	Measure	Value
Neural Network	Specificity	0.7798142
Logistic Regression	Specificity	0.9159275
Logistic Regression (P_2 only)	Specificity	0.8889403
Random Forest	Specificity	0.7843961
Neural Network	Sensitivity	0.9370873
Logistic Regression	Sensitivity	0.8556891
Logistic Regression (P_2 only)	Sensitivity	0.7953499
Random Forest	Sensitivity	0.9304990
Neural Network	Accuracy	0.8963263
Logistic Regression	Accuracy	0.8713013
Logistic Regression (P_2 only)	Accuracy	0.8196060
Random Forest	Accuracy	0.8926330
Neural Network	AMEX Metric	0.7118359
Logistic Regression	AMEX Metric	0.6160811
Logistic Regression (P_2 only)	AMEX Metric	0.5847342
Random Forest	AMEX Metric	0.6964640
Logistic Regression	AUC	0.9527000
Logistic Regression (P_2 only)	AUC	0.9168000
Neural Network	AUC	0.9547000
Random Forest	AUC	0.9187000

9.2 Appendix B: additional figures

Insert content for additional figures here.

9.3 Appendix C: code

```
amex_metric = function(target, pred){

  df = data.frame(target, pred)
    # Default rate captured at 4%
    top_four_percent_captured = function(target, pred){
    df = data.frame(target, pred)
    # Descending predictions (pred == 1 first)
    df = df  %>% arrange(-pred)
    # define weight (negative labels are given a weight of 20 to adjust for down sampling)
    df[,'weight'] = ifelse(df[,'target'] == 0, 20, 1)
    # get rows under 4% cutoff
    pctCut = as.integer(sum(0.04*df[,'weight']))
    pctCut = df[cumsum(df[,'weight']) <= pctCut,]
    # return
    sum(pctCut['target'] == 1)/sum(df['target'] == 1)
  }

  weighted_gini = function(target, pred){
    df = data.frame(target, pred)
    df = df  %>% arrange(-pred)
    # define weight (negative labels are given a weight of 20 to adjust for down sampling)
    df[,'weight'] = ifelse(df[,'target'] == 0, 20, 1)
    df[,'random'] = cumsum(df[,'weight'] / sum(df[,'weight']))
    total_pos = sum(df[,'target'] * df[,'weight'])
    df[,'cum_pos_found'] = cumsum(df[,'target'] * df[,'weight'])
    #Define lorentz and Gini variables
    df[,'lorentz'] = df[,'cum_pos_found'] / total_pos
    df[,'gini'] = df[,'weight'] * (df[,'lorentz'] - df[,'random'])
    # return
    sum(df[,'gini'])
  }

  normalized_weighted_gini = function(target, pred){
    weighted_gini(target, pred) / weighted_gini(target, target)
  }

  G = normalized_weighted_gini(target, pred)
  D = top_four_percent_captured(target, pred)

  # Return val
  0.5 * (G + D)

}
```

```r
pred_full <- readRDS("pred_full.rds")
# completely right - all rows in agreement 1 or 0
sum(rowSums(pred_full) == 5 | rowSums(pred_full) == 0 )/nrow(pred_full)

# completely wrong - four predictions opposite to observed
comp_wrong1 <-(pred_full[,1] == 0) & (rowSums(pred_full[,c(2,3,4,5)]) == 4)
comp_wrong2 <-(pred_full[,1] == 1) & (rowSums(pred_full[,c(2,3,4,5)]) == 0)

(as.numeric(summary(comp_wrong1)[3])+as.numeric(summary(comp_wrong2)[3]))/nrow(pred_full)

# majority vote - 2 vs vote for default
maj <- ifelse(rowSums(pred_full[,c(2,3,4,5)]) < 2, 0, 1)

table(maj, pred_full[,1])
#sensitivity specificity and accuracy
tabfunc <- function(pred, obs){
  tab <-table(obs, pred)
  acc <- sum(diag(tab))/sum(tab)
  sens <- tab[2,2]/sum(tab[2,c(1,2)])
  spec <- tab[1,1]/sum(tab[1,c(1,2)])
  return(c(sens, spec, acc))
}
# sens, spec and acc
tabfunc(maj,pred_full[,1])
```

Insert code (if any) used during your dissertation work here.

10 Abbreviations

Table 4: List of Abbrevivations

Abbreviation	Meaning
AI	Artificial Intelligence
AMEX	American Express
AUC	Area Under the Curve
DNN	Deep Neural Network
GRU	Gated Recurrent Unit
LR	Logistic Regression
ML	Machine Learning
NN	Neural Network
RF	Random Forest
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SVM	Support Vector Machines
TNR	True Negative Rate
TPR	True Positive Rate