The Impact Of Machine Learning In Credit Scoring Of Off-Credit Data

Johannes Tebalo Kokozela

UNIVERSITY OF THE WITWATERSRAND

WITS
UNIVERSITY

School of Computer Science and Applied Mathematics
Faculty of Science
HONOURS RESEARCH REPORT

Supervisor(s): Prof. Celik Turgay and Dr. Terrence Van Zyl

November 2017, Johannesburg

DECLARATION
UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG
SCHOOL OF COMPUTER SCIENCE AND APPLIED MATHEMATICS
SENATE PLAGIARISM POLICY

I, Johannes Tebalo Kokozela, (Student number: 714188) am a student registered for COMS4051 in the year 2017.
I hereby declare the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that ALL the work submitted for assessment for the above course is my own unaided work except where I have explicitly indicated otherwise.
- I have followed the required conventions in referencing the thoughts and ideas of others.
- I understand that the University of the Witwatersrand may take disciplinary action against me if there is a belief that this in not my own unaided work or that I have failed to acknowledge the source of the ideas or words in my writing.

Signature:

Signed on _____ day of _____ , 2017 in Johannesburg.

The Impact Of Machine Learning In Credit Scoring Of Off-Credit Data

Johannes Tebalo Kokozela, 714188

ABSTRACT

Credit scoring is a method used by financial institutions to evaluate the credit worthiness of a customer. Traditionally logistic regression and fisher's linear discriminant analysis are the most used algorithms for credit scoring. We compare these two algorithms on a large dataset from Amalgamated the Bank of South Africa (ABSA) with four machine learning algorithms: support vector machines, artificial neural network, naïve bayes and two variations of the *k*-nearest neighbors. We find that linear support vector machines performs better than all other algorithms. We also found that logistic regression and *k*-nearest neighbor are among the high performing algorithms when the performance measure is the accuracy and receiver operating characteristic's area under the curve.

I. INTRODUCTION

Credit Scoring is a method used by financial institutions to determine the credit worthiness of a customer i.e. whether to lend a potential client capital or not. Clients are classified as either good or bad depending on their credit worthiness or score. It is very important for banks to have the most accurate models of determining if a client is worth lending money. The objective of credit scoring methods is to help the bank to find good credit applications who are likely to observe obligation according to their age, credit limit, income and marital status [1]

A. Importance of study

According to [2] credit scoring models provides a decreased cost of credit analysis, assessment of credits with an effective and rapid decision making process, higher probability of credit repayments and lower possible risk. Although most studies focus on the benefits of the lender alone, credit scoring is also becoming increasingly important for clients as well, with the increasing amount of debt accumulated each year by individuals and institutions, accurate knowledge of ones own financial status could be very helpful in reducing the debt problem. Credit risk evaluation are inherently difficult due the the forms of risk associate with errors in classifications [3]. In the past this was a very difficult problem for institutions, since they could only depend exclusively on the information provided by the client on the application form and data from the credit bureau. With the introduction of numerical score cards in the 1930's by mail-order companies, a range of different data mining and statistical techniques were then utilized [4].

The models or pattern recognition techniques used in credit scoring can be roughly divided into two groups: statistical approach and machine learning (ML) methods [5]. The commonly used statistical models for credit scoring in banks are Linear Discriminant Analysis (LDA) and Logistic Regression (LR) . This is mostly due to their ease of implementation and the interpretability of the results, they also have high accuracy. The drawback of both these models is the assumption of linear relationship between input and output variables, which firstly is difficulty to verify as the dimensionality of the dataset increases and seconds the assumption seldom holds. In addition to the the linearity assumption, the LDA also has an additional multivariate normality assumption. Within the ML models, the generally used model for credit scoring is the Artificial Neural Network (ANN) [6]–[8]. The ANN algorithm presents several advantages over the statistical methods, firstly ANN makes no assumptions about the underlying distribution of the data, secondly unlike other learning methods, real-valued inputs can be used, the neural network can be implemented without specifying the model up-front and finally, it is excellent in generalisation [3]. ANN also has drawbacks, firstly, ANN has a long training process in finding the optimal network parameters, secondly, it is not easy to identify the relative importance of each input variable as would with the logistic regression where we would use the Wald statistics and lastly, ANN presents some interpretation difficulties [1].

Other Methods in ML have been explored, which includes Support Vector Machine (SVM), Naïve Bayes and *k*-nearest neighbour. SVM projects the data into a high dimension and fits a boundary that separates the classes using the support vectors [4], [9]. *k*-Nearest neighbor (*k*NN) has also been explored in credit scoring with the adjustment in the distance matrix. Naïve Bayes (NB) are build on Bayesian classification methods, it assumes that the data from different classes are normally distributed [10]. Most of the research on credit scoring is done using either simulated data or a small dataset [4]. This paper compares the models presented above using huge dataset or big data from a bank [11]. The models will be compared using Precision, Recall, Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) and accuracy. The data is on policies (off-credit).

The rest of the paper is structured as follows, Section II explores the models used in credit scoring and reviews the previous literature related to the models studied in this paper. Section III describes the data, experimental design and model development. The experimental results with performance evaluation and comparison is discussed in Section IV. Section V concludes the paper with suggestions for further research.

II. Background and Related Work

In this section, six commonly used models for credit scoring are discussed with their optimisation equations. Currently, a lot of institutions are devoting their resources in understanding their customer behaviour through data. According to [6] banks must use the capital at their disposal in a better way. The capital is not only money but also also the data stored in the databases. Four of the five models presented in this section work well with huge amounts of data but, the standard *k*-NN does not, this is due to the fact that distance has to be computed every time a customer or client is to be classified.

A. Fisher's LDA

LDA is among the commonly used models for credit scoring in industry and in literature for benchmarking other models [1], [4]. LDA was proposed in 1936 by Fisher as a classification technique. LDA tries to maximize the distance between the means of the two classes in relation to their variance [1] .The objective is to maximize the Fishers discriminant criterion i.e.

$$\begin{aligned}\boldsymbol{J}(\boldsymbol{w}(\boldsymbol{x_1}, y_1, ..., \boldsymbol{x_n}, y_n)) =& (\boldsymbol{\mu_1} - \boldsymbol{\mu_0})^T (\boldsymbol{\mu_1} - \boldsymbol{\mu_0}) \\ & \left[\boldsymbol{S_1}^2 + \boldsymbol{S_0}^2\right]^{-1},\end{aligned} \tag{1}$$

where $y_i \in \{-1, 1\}, \forall i \in [1, n]$ is class label, $\boldsymbol{\mu_y}$ and $\boldsymbol{S_y}^2$ are respectively, mean vector and variance matrix of each class i.e.

$$\boldsymbol{\mu_y} = \frac{1}{|C_y|} \sum_{i \in C_y} \boldsymbol{w} \cdot \boldsymbol{x_i}$$

$$\boldsymbol{S_y}^2 = \frac{1}{|C_y|} \sum_{i \in C_y} (\boldsymbol{w} \cdot \boldsymbol{x_i} - \boldsymbol{m_y})^T (\boldsymbol{w} \cdot \boldsymbol{x_i} - \boldsymbol{\mu_y})$$

where $C_y = \{i = 1, ..., n | y_i = y\}$. [1] Maximizing (1) yields $0 = \boldsymbol{w} \cdot \boldsymbol{x}$ which maximizes the distance between the means in relation to the variance. New examples $\hat{\boldsymbol{x}}$ can then be classified by computing $\hat{y}$ =sign$(\boldsymbol{w} \cdot \hat{\boldsymbol{x}})$.

[4] used LDA as benchmark for the SVM on a dataset of size $25,000$ records and the results were not very significant in comparison, with the mean AUC[12] of the LDA being $0.781(0.0058)$, Linear SVM $0.783(0.0055)$, polynomial SVM $0.755(0.0068)$ and Gaussian RBF $0.783(0.0053)$. [1] also used LDA testing against Multivariate Adaptive Regression Spline (MARS), which had an accuracy of 82.5% and the LDA had an accuracy of 76% and finally LR had an accuracy of 76.5% .

B. LR

LR is also a commonly used model for credit scoring and thus also used in literature for benchmarking newly introduced models for credit scoring models that may improve the accuracy [1], [4], [5], [9]. The cost function to be minimized for credit scoring is:

$$\boldsymbol{J}(\boldsymbol{w}) = -\frac{1}{N}\left[\sum_{i=1}^{N} y_i \ln(\pi_i) + (1 - y_i) \ln(1 - \pi_i)\right], \tag{2}$$

where the π_i in (2) is sigmoid function of the linear combination of $\boldsymbol{w}$ and $\boldsymbol{x}$ given by :

$$\pi_i = h_{\boldsymbol{w}}(\boldsymbol{x}) = \frac{1}{1 - \exp(-\boldsymbol{w}^T \cdot \boldsymbol{x})},$$

where π_i can be interpreted as the probability of $y_i = 1$ given $\boldsymbol{x}$ or $P(y_i = 1|\boldsymbol{x})$ and $y_i \in \{0, 1\}$ [2].

[4] used LR to compare the accuracy of different kernels for the SVM model, they found that SVM with linear and Gaussian RBF both perform better than LR, with both having an average AUC of $0.783(0.0054)$ while LR had average AUC of $0.773(0.0063)$. [9] compared LR with *k*-NN constructed by adjusted Euclidean matrix, they found that *k*-NN performed better in terms of bad (classification) rates, *k*-NN having a rate of 43.09 and LR had 43.30. [13] compared different Neural models to LR and found that LR out performed all neural models on both datasets tested on .

C. k-NN

k-NN is a ML non-parametric classification algorithm. The *k*-NN algorithm classifies a new test example $\hat{\boldsymbol{x}}$ by finding the k nearest samples in the training data by calculating the distance between it and the features in the feature matrix $\boldsymbol{X}$, then classify $\hat{\boldsymbol{x}}$ according to the k [3]majority labels within the neighborhood (smallest distance) of $\hat{\boldsymbol{x}}$ as the value of $\hat{\boldsymbol{y}}$ the class label of $\hat{\boldsymbol{x}}$.

k-NN is also used in some literature as a benchmark for other algorithms [13] and [4] used *k*-NN with other algorithms to test variations of neural networks and SVM's. [9] studied *k*-NN with an adjusted distance matrix for credit scoring.

D. SVM

SVM fits a hyperplane onto the data such that the plane separates the classes. This is an optimization problem such that the objective is to maximize the margin width given by $\frac{2}{\|\boldsymbol{w}\|}$, which is the distance from the hyperplane to the data points. The cost function is given by

$$\boldsymbol{J}(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{l}(\boldsymbol{x}_i, y_i|\boldsymbol{w}),$$

where $\lambda = \frac{2}{NC}$, $\boldsymbol{l}(\boldsymbol{x}_i, y_i|\boldsymbol{w}) = \max(0, 1 - y_i h(\boldsymbol{x_i}))$, $h(\boldsymbol{x_i}) = \sum_i \alpha_i y_i (\bar{\boldsymbol{x}}_i^T \boldsymbol{x_i})$ and $\bar{\boldsymbol{x}}_i$'s are support vectors. [4] study the three kernels for SVM and the interpretation of the parameters $\boldsymbol{w}$ in the credit scoring domain.[2] study different models with their ensembles for credit scoring including SVM and SVM-bagging. [1] try three SVM based models for credit scoring: 1) grid search to optimize the models parameters 2) Use the grid

[1] see Bellotti *et al* 2009 [4] for all LDA formulations

[2] for LR we need to change the labels from {-1,1} to {0,1} since these will be interpreted as probabilities

[3] k is always odd

search results with the F-score to select significant features, 3) Use SVM with genetic algorithm (GA) to simultaneously optimize model parameters and input features.

E. ANN

The ANN unlike the above mentioned statistical methods, assumes no structure in the data and it is inspired by the biological model of the brain. The only assumption made is that of the architecture of the neural network. In the multi-layer ANN model has a minimum of 3 layers, the first and last layer being the input and output layer, the number of middle layers can vary and is called the hidden layers as depicted in Fig. 2. The number of nodes of the input layer depends on the the number of features of the problem and the output layer for credit scoring is typically one, which is commonly interpreted as the probability of delinquency [14]. The weights of the ANN are found using back propagation such that they minimize the following cost function :

$$\begin{aligned}\boldsymbol{J}(\boldsymbol{w}) = &- \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\left[y_k^n \ln(\pi_k^n) + (1-y_k^n)\ln(1-\pi_k^n)\right] \\ &+ \frac{\lambda}{2N}\sum_{l=1}^{L-1}\sum_{i}^{l_i}\sum_{j=1}^{S_{l+1}}\left(\boldsymbol{w}_{j,i}{}^{(l)}\right)^2,\end{aligned} \tag{3}$$

where all the variable are defined as in logistic regression and L is the total number of layers and l_i is the ith layer.

[8] explored the ability of the neural networks such as multilayer perceptron (MLP) and modular neural network (MNN) in building credit scoring models at the credit scoring union environment. These models were tested against the traditional models i.e. LR and LDA. They found that neural networks offer a promising avenue if the performance measure is the percentage of bad loans correctly classified. [13] investigated the credit scoring accuracy of five neural network models namely MLP, mixture-of- experts, radial basis function (RBF), learning vector quantization and fuzzy adaptive resonance. [13] found that the LR is most accurate model (compared to the five ANNs). [13] also suggest that Mixture-of-experts and RBF be considered for credit scoring applications, since their performance was closer to that of LR. [6] suggest an enhancement on the ANN classification accuracy using GA for feature selection (FS). They found that the hybrid of ANN with GA performed much better than ANN alone for credit assessment. [2] provided a comparison of four types of FS methods which are the Genetic Algorithm, relief method, principal component analysis (PCA) and information gain. The selected features were further used to compare the following classification algorithms: Classification and regression trees (CART), ANN, GA, NB and SVM, Random forest, stacking with their adaptive boosted versions and bagging. They found that the adaptively boosted ANN performed much better than others with higher accuracy and AUC score.

F. NB

Naïve Bayes tries to reduce the complexity of the intractable sample in order the learn the Baysian Classifiers. It does this by making a conditional independence assumption that dramatically reduces the number of parameters to be estimated when modeling $P(X|Y)$, from our original $2(2^n - 1)$ to just $2n$ [15]. A new test example is then classified by using Bayes rule to compute the posterior probability of each class y given the vector of observed attribute values:

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})}$$

[10]. [10] study different state of the art classifiers for credit scoring including the NB, they found that the SVM and ANN are among the best perfomoring algorithms using accuracy and ROC-AUC.

III. Research Methodology

A. Data

A dataset of more than $195,000$ records of life policy holders of the Amalgamated Bank of South Africa (ABSA), one of the big five banks of South Africa. The duration of the policies range from 1 to 9 years. A customer is said to have defaulted (lapsed) if he or she is at least 3 months behind his/her payment of the policy, this definition of default is given by [4]. The data contained 35 attributes that were attained from the customers original application and on going changes in the policy.

B. Feature selection

To select the most significant attributes into the feature space, we used the PCA algorithm, which retains the most variation in the data. The choice of PCA was motivated by the results obtained by [2], who showed that PCA outputs features that retains the most variation and give more accurate features for credit scoring. Thus we had 19 features selected for the classification models. The variables were standardized to have mean zero (0) and variance (1). 62.5% of the data were good classification and 37.5% were bad classifications. The data was split into 60% for training and 40% for testing.

C. Research hypothesis

ANN, SVM, NB, *k*-NN and adjusted *k*-NN (ML) algorithms perform better then LDA and LN (statistical models), in credit scoring of large datasets that do not entail credit history of the applicants.

D. Implementation

We used Python 3.6.0 programming language on Anaconda version 2 throughout this research. The Python packages used are Sklearn [16] for preprocessing and performance metrics, Numpy (numerical python) [17] for optimized functionality in matrix algebra, Pandas (Panel data) [18] for handling dataframes and matplotlib for plotting [19]. The operating system of the computer used is macOS Sierra version 10.12.6 and the hardware on the computer was an intel i7 processor and 16 GB of memory (RAM). For three of the eight models (the linear SVM, radial basis function SVM and Naïve Bayes)

we used the Sklearn package this is because the models are more robust and optimized.

For the LDA we used equation (1) to calculate the weights vector $\boldsymbol{w}$. The classification were calculated different from those given in II, we used a threshold α calculated using $(\boldsymbol{\mu_0} \cdot \boldsymbol{w} + \boldsymbol{\mu_1} \cdot \boldsymbol{w})/2$ where $\boldsymbol{\mu_0}$ and $\boldsymbol{\mu_1}$ are the mean vector calculated from the two classes. This approach is much better than the one presented in section II where the data is inseparable as illustrated in Fig. 1 .

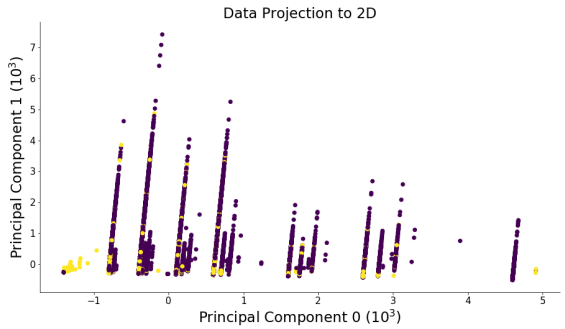

Fig. 1. PCA projection into two 2 dimensional space

For the LR and ANN we used stochastic gradient descent algorithm (SGB) [20] to obtain the weights $\boldsymbol{w}$ in (2) and (3). For LR the lowest value for the cost function was obtained after $894,200$ iterations with the learning rate (σ) set to 0.1.

For the SVM we used a linear kernel (SVM) so as to obtain the weights $\boldsymbol{w}$ easily for future predictions. The parameters C and N where set to 1.0 and 117360 (size of the training set) by default from Sklearn. We also used SVM with radial basis function (RBF).

For k-NN we used the adjusted distance matrix given by [9] which calculates the distance matrix as $d_3(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \left[(\boldsymbol{x} - \hat{\boldsymbol{x}})^T (\boldsymbol{I} + \boldsymbol{D}\boldsymbol{\omega}\boldsymbol{\omega}^T)(\boldsymbol{x} - \hat{\boldsymbol{x}})\right]^{\frac{1}{2}}$ where we set $D = 1.40$ and $\boldsymbol{\omega}$ is calculated as $\boldsymbol{\omega} = \ln\left(\frac{P_{ij}}{Q_{ij}}\right)$ where P_{ij} and Q_{ij} are to be interpreted as the proportion of those classified good in attribute i of characteristic j and Q_{ij} is the proportion of those classified bad in attribute i of characteristic j We chose a value of $k = 3$.

For the ANN we used had 3 layers as illustrated in Fig. 2 below. The architecture was as follows 19 nodes on the input layer, 9 nodes on the hidden layer and 1 for the output layer. The 1 on the output layer is a node outputting probabilities of the output being 0 (not lapsed) and 1 (Lapsed). To obtain this architecture different variations of the number of hidden layers and nodes on the hidden layers were tried and this gave faster and better results after more than 1000 epochs.

E. Performance Evaluation

In order to measure and compare the performance of our models, we used two performance evaluation measures. The accuracy which is as given in equation (4) [2].

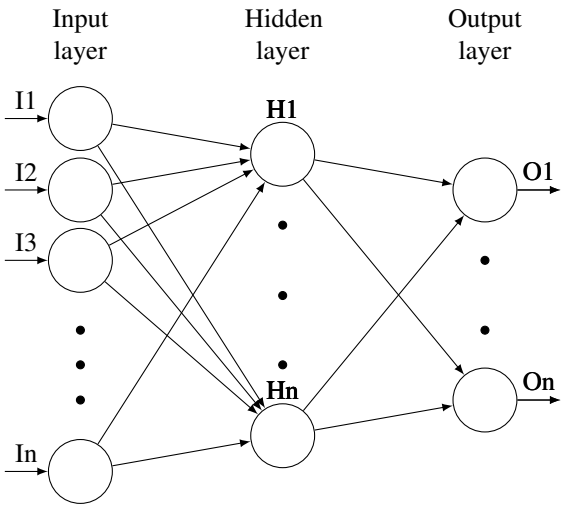

Fig. 2. Generalized structure of ANN.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

Where P and N are Positive and Negative, and F and T are False and True. We also use ROC (Receiver Operating Characteristic) curve, which plots true positive rate (sensitivity) against negative rate (1-specificity) for all possible threshold values. We use both accuracy and area under the curve of the ROC as a single summary statistics [4].

IV. Experiments

In this section the results of our experiment are presented. Table (I) below gives the results in terms of accuracy and ROC AUC measures.

TABLE I
Accuracy and AUC

Algorithm	**Accuracy**(%)	**AUC**(%)
LDA	51	61
LR	80	76
SVM	85	87
k-NN	76	74
ANN	59	60
Adjusted kNN	50	54
RBF-SVM	66	55
NB	63	67

To get a clearer comparison, the results of the classification algorithms were plotted on a bar graphs, In Fig. 3 we set the y range to $y \in \{0.20, 0.90\}$ to highlight the differences on the algorithms performance. Fig. 3 summarises all the information in Table I in a bar graph which compares algorithms performances in terms of accuracy and ROC-AUC of each algorithm.

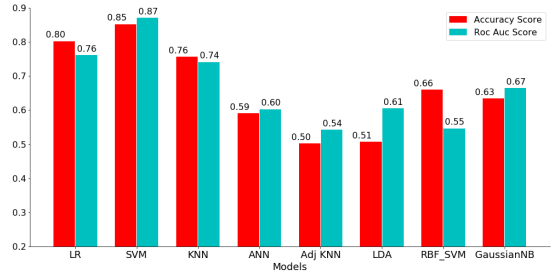

Fig. 3. The results of classification algorithms by accuracy and AUC

A more precise representation of the results of the AUC scores in Table I and Fig. 3 is the ROC curve which gives a measure of how far the algorithm is from randomly guessing the values (area between baseline). Fig. 4 below gives the ROC curve for each algorithm.

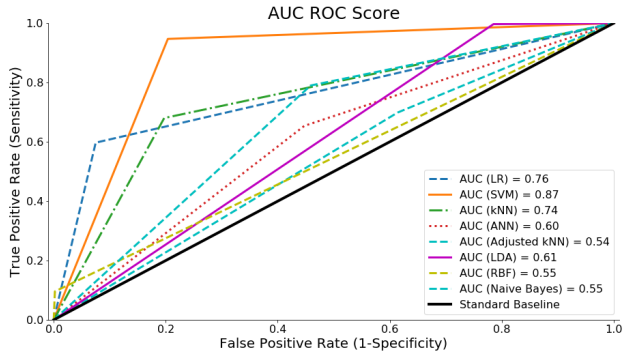

Fig. 4. ROC AUC curve

The baseline indicates a $45°$ line, which is a line the plot would follow if it was randomly guessing the results. From the plot we observe that none of the algorithms are randomly guessing the results.

Other measures of performance used are precision and recall of the algorithm. Precision and recall both measure how well an algorithm is able to reconstruct the the given classes or labels. The two tables below give the precision and recall of the algorithms.

TABLE II
PRECISION - RECALL TABLE

Algorithms	Precision	Recall	f1-Score
LDA	0.43	0.99	0.60
LR	0.83	0.60	0.69
SVM	0.74	0.43	0.82
*k*NN	0.67	0.68	0.68
ANN	0.47	0.65	0.54
Adjusted *k*NN	0.40	0.70	0.51
NB	0.51	0.78	0.61
RBF-SVM	0.96	0.1	0.17

Fig. 5 gives a summary of Tables II which is a step function of the intersection between precision and recall. The highest point in the plot represent the point where precision and recall are the highest in the classification algorithm.

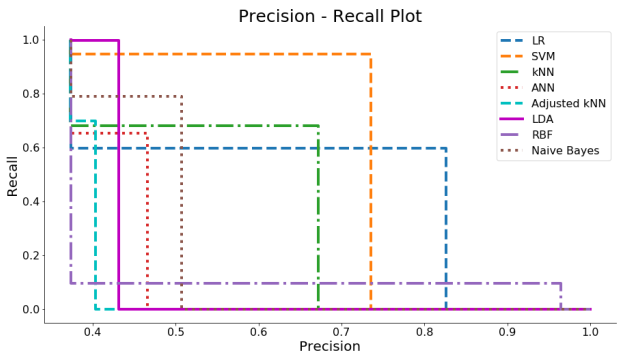

Fig. 5. Precision recall plot

A. *Discussion*

From Table I and Fig. 3 - 5 we observe that the best performing algorithm is the Support Vector Machine with linear kernel which is consistent with the results obtained by [10] . The results also show that Linear Regression has a higher AUC amongst the statistical algorithms which is also consistent with [13], but *k*-NN had a higher accuracy than RBF-SVM [4]. LDA, ANN and adjusted *k*-NN are among the lowest performing algorithms in terms of AUC and accuracy. From Table II and Fig 5 adjusted *k*-NN had among the highest recall for bad labels, which is also consistent with [9], since their measure was bad rate classifications.

V. CONCLUSION

Machine learning approaches are being extensively applied in to the problem of credit scoring, we test the the frequently used algorithms on a much larger dataset than previous research. We found that although all the algorithms were very slow to train, the Support Vector Machine with a linear Kernel function was better performing than all other algorithms, and on average the statistical algorithms performed better than machine learning algorithms with mean AUC of $0.68(0.078)$ against $0.66(0.12)$, but had a lesser accuracy with mean accuracy of $0.65(0.15)$ and $0.67(0.11)$ for statistical and machine learning algorithms respectively. Although no formal test was performed to verify, due to the small number of statistical algorithms.

It is worth noting that the attributes are all non-related to the credit history of the customers, we also make the assumption that if a customer lapsed on the policy they are very likely to lapse on any credit they may acquire.

Both the *k*-NN and adjusted *k*-NN were trained and tested on only 10% of the entire dataset which was due to the memory consumption of the distance matrix. We also note that the ANN had a suspiciously low accuracy and AUC, this

could be due to the low number of iteration in comparison to the logistic regression or the construction of the algorithm.

There are several lines of investigations. Firstly, the LDA and SVM could be constructed using different kernel functions. Secondly, in this paper we did not consider different values of k for k-NN and adjusted k-NN, this is because we wanted to obtain results to are similar to that obtained by [9]. Further research might also consider the adaptive boosted version of the algorithms considered as [2]. Finally, we did not focus much on the feature selection algorithms, thus future research could apply the algorithms provided by [6] and [21].

ACKNOWLEDGEMENTS

I would like to thank both my supervisors, Prof. Turgay Celik and Dr. Terence Van Zyl, for their insight in both the development of the algorithms and the writing of this paper.

I would also like to thank the ABSA team Eon Retief and Daniela Casilli for providing us with data and helping with the preprocessing of it.

REFERENCES

[1] C.-L. Chuang and R.-H. Lin, "Constructing a reassigning credit scoring model," *Expert Systems with Applications*, vol. 36, no. 2, Part 1, pp. 1685 – 1694, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417407005854

[2] F. N. Koutanaei, H. Sajedi, and M. Khanbabaei, "A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring," *Journal of Retailing and Consumer Services*, vol. 27, pp. 11 – 23, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0969698915300060

[3] S. Piramuthu, "Financial credit-risk evaluation with neural and neurofuzzy systems," *European Journal of Operational Research*, vol. 112, no. 2, pp. 310 – 321, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221797003986

[4] T. Bellotti and J. Crook, "Support vector machines for credit scoring and discovery of significant features," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3302–3308, 2009.

[5] Y. Xia, C. Liu, Y. Li, and N. Liu, "A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring," *Expert Systems with Applications*, vol. 78, no. Supplement C, pp. 225 – 241, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417417301008

[6] S. Oreski and G. Oreski, "Genetic algorithm-based heuristic for feature selection in credit risk assessment," *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 2052 – 2064, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417413007239

[7] Z. Zhao, S. Xu, B. H. Kang, M. M. J. Kabir, Y. Liu, and R. Wasinger, "Investigation and improvement of multi-layer perceptron neural networks for credit scoring," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3508 – 3516, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417414007726

[8] V. S. Desai, J. N. Crook, and G. A. Overstreet, "A comparison of neural networks and linear scoring models in the credit union environment," *European Journal of Operational Research*, vol. 95, no. 1, pp. 24–37, 1996.

[9] W. Henley *et al.*, "Construction of a k-nearest-neighbour credit-scoring system," *IMA Journal of Management Mathematics*, vol. 8, no. 4, pp. 305–321, 1997.

[10] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking state-of-the-art classification algorithms for credit scoring," *Journal of the operational research society*, vol. 54, no. 6, pp. 627–635, 2003.

[11] J. S. Ward and A. Barker, "Undefined by data: a survey of big data definitions," *arXiv preprint arXiv:1309.5821*, 2013.

[12] D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems," *Machine learning*, vol. 45, no. 2, pp. 171–186, 2001.

[13] D. West, "Neural network credit scoring models," *Computers & Operations Research*, vol. 27, no. 11, pp. 1131–1152, 2000.

[14] T. M. Mitchell, "Artificial neural networks," *Machine learning*, vol. 45, pp. 81–127, 1997.

[15] S. PERMISSION, "Generative and discriminative classifiers: Naive bayes and logistic regression," 2005.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[17] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[18] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445. SciPy Austin, TX, 2010, pp. 51–56.

[19] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[20] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 116.

[21] V. Kozeny, "Genetic algorithms for credit scoring: Alternative fitness function performance comparison," *Expert Systems with Applications*, vol. 42, no. 6, pp. 2998 – 3004, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417414007143