Theoretical and Applied Economics
Volume XXVI (2019), No. 4(621), Winter, pp. 75-84

A Deep Neural Network (DNN) based classification model in application to loan default prediction

Selçuk BAYRACI
R&D Center, C/S Information Technologies, Istanbul, Turkey
selcuk.bayraci@cybersoft.com.tr
Orkun SUSUZ
R&D Center, C/S Information Technologies, Istanbul, Turkey

Abstract. *In this study, we applied a Deep Neural Networks (DNN) based classification model along with the conventional classification methods (Logistic Regression, Decision Tree, Naïve Bayes and Support Vector Machines) on a two distinct datasets containing characteristics of the loan clients in a medium-sized Turkish commercial bank. Python programming language and libraries (Sklearn, Tensorflow and Keras) have been used in data cleaning, data preparation, feature engineering and model implementation processes. Our empirical findings document that the accuracy of the deep learning classification model increases with the size of the dataset, implying that the deep learning models might yield better results than regression-based models in more complex datasets.*

Keywords: data analytics; credit scoring; deep learning; risk management.

JEL Classification: C01, C40, C87.

1. Introduction

Following the 2008 US subprime mortgage crisis which had devastating effects on many financial institutions across the globe, the need for building reliable and solid credit scoring systems has been intensified, thus, for the banks and lending institutions, discriminating bad customers from the good ones became pivotal. Since the pioneering study of Altman (1968), many statistical and machine learning techniques have been employed for credit risk measurement. There is a plethora of research in the literature devoted to prediction of defaults in the consumer loan market. The regression and classification based models have been the status quo in both the industry and the academic literature for a long time. Discriminant analysis, Logistic Regression, Support Vector Machines (SVMs), classification and regression trees and Naïve Bayes classifiers have been frequently used for classifying loans into bad and good categories.

Previous studies in the field (Hand and Henley, 1997; Abdou and Pointon, 2011; Lessman et al., 2015; Louzada et al., 2017) suggest that, machine learning methods yield better predictive accuracy compared to statistical models. On the other hand, as suggested by Hinton and Salakhutdinov (2006), these methods mainly focus on the outputs of classifiers at the shallow level, while ignoring the rich information hidden in the confidence degree thus their limited modelling and representational power can cause difficulties when dealing with more complicated datasets.

Due to recent developments in computer technology, it is possible to construct training algorithms for deep architectures. The deep belief networks (DBN) and ANNs with sufficient hidden layers are developed as powerful ensemble techniques to capture the rich information hidden in the datasets. Deep learning methods have been applied for classification tasks in various fields from computer vision to speech and language processing. Recently, the deep learning methods have also been used in financial applications (Ribeiro and Lopes, 2011; Tomczak and Zieba, 2015; Giesecke et al., 2016; Luo et al., 2017; Kvamme et al., 2018; Hamori et al., 2018).

In this study, we applied a Deep Neural Networks (DNN) with multiple hidden layers to assess the risk profiles of loan clients on datasets taken from a Turkish commercial bank. We compared the predictive ability of deep learning method vis-à-vis Logistic Regression (LR), J48 Decision Tree, Naïve Bayes and Support Vector Machines (SVM).

The rest of this paper is laid out as following. Classification methods are described in section 2. Section 3 gives a brief overview of the datasets used in the experiments. Model implementation processes and results are presented in section 4, and section 5 concludes.

2. Methods

The quantitative techniques that adopted in this paper are these four aforementioned methods of constructing the default characteristic predicting model: DNN, Logistic Regression, J48, Naïve Bayes and SVM. The brief description of each model that used in this study is as follows.

2.1. Deep Neural Networks

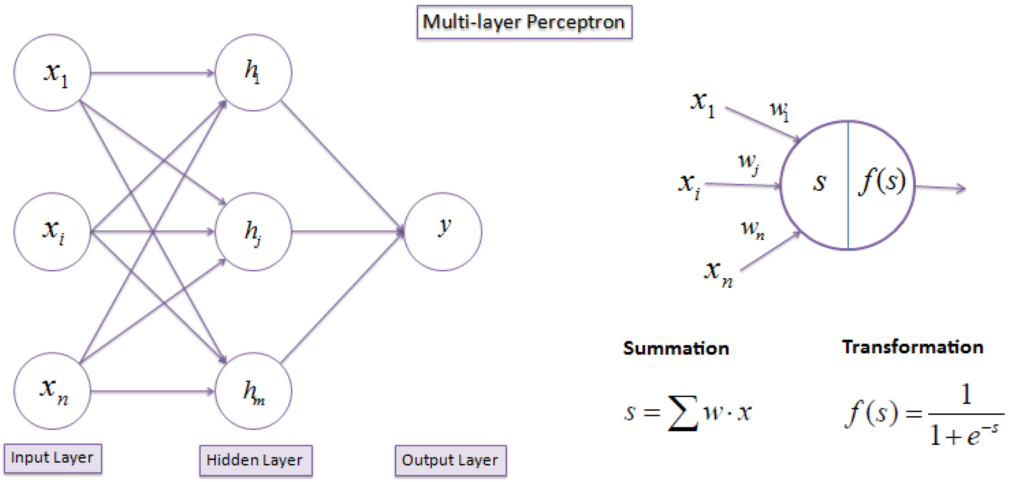
A neural network (NN) is a network structure comprising multiple connected units. It has three layers of units: input layers, hidden layers, and output layers (Figure 1). The neural network configuration is determined by the manner in which the units are connected. When the number of hidden layers is more than or equal to two, the network is called a deep neural network (DNN). The feed-forward neural network is the most widely used neural-network model and is configured by the connection of multiple units, with reference to West (2000), the propagation of the network in each layer is accomplished in following steps.

Step 1: A weighted sum is calculated at each neuron, that is the output value of each neuron in the proceeding network layer times the respective weight of the connection with that neuron.

Step 2: A transfer function $f(s)$ is then applied to this weighted sum to determine the neurons output value.

Step 3: The output value y, can be expressed as a function of the input values and network weights

Figure 1. *The architecture for a three-layer artificial neural network*



2.2. Logistic regression

Logistic Regression is part of a widely used general family of models, introduced by McCullagh and Nelder (1989), known as generalized linear models (GLM). GLMs provide a unified framework to model response from any member of the exponential family distributions, such as Gaussian, Binomial, or Poisson. In GLM framework, the model is quantified from a binary target variable, Y which represents the status of a loan over the outcome window where bad (defaulted) loan is labelled as 0 and good loan is labelled as 1 and related to the linear combination of predictor $\beta_1 * X1 + \ldots + \beta_m * X_m$ in the form of

$$\boldsymbol{G(E(Y|X)) = G(u) = \beta_0 + \sum_{i=1}^{m} \beta_i X_i}, \tag{1}$$

where $\boldsymbol{u}$ is the mean of dependent variable $\boldsymbol{Y}$ and $\boldsymbol{G}(.)$ is a monotonic differentiable function known as Link Function. For Logistic Regression, the functional form can be expressed as

$$\boldsymbol{Logit(p)} = \mathbf{log}\left(\frac{\boldsymbol{p}}{\mathbf{1}-\boldsymbol{p}}\right) = \boldsymbol{\beta_0} + \sum_{i=1}^{m} \boldsymbol{\beta_i X_i}, \tag{2}$$

where $(\boldsymbol{u})$ is $\boldsymbol{p = Prob(Y = 1|X)}$ and $\boldsymbol{G}(.)$ is $\boldsymbol{Logit}(.)$ function in this case.

2.3. J48 Decision Tree

Decision tree learning algorithms use a decision tree as a predictive model which maps observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a finite set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. In this study, J48 which is an open-source Java implementation of C4.5 (Quinlan, 1986) algorithm has been used to construct the decision tree.

2.4. Support Vector Machines

The main aim of the SVM algorithm is to separate good credits ($y = 1$) from bad credits ($y = 0$) described with a d dimensional vector of characteristics x. We use $y = \{-1,1\}$ instead of the common $y = \{0,1\}$ notation since it is more convenient in the following formal expressions. The SVM separates the two groups with the maximum distance (margin) between them. The score for x is computed as

$$\boldsymbol{f(x)} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{\alpha_i y_i K(x, x_i) + b} \tag{3}$$

In our classification problem, we use radial basis function or Gaussian kernel due to unknown relationships between the input variables. The Gaussian kernel on two samples $\boldsymbol{x}$ and $\boldsymbol{x_i}$ is defined to be:

$$\boldsymbol{K(x, x_i)} = \mathbf{exp}\left(-\frac{||\boldsymbol{x} - \boldsymbol{x_i}||^2}{\mathbf{2}\boldsymbol{\sigma}^{\mathbf{2}}}\right) \tag{4}$$

When the distance between $\boldsymbol{x}$ and $\boldsymbol{x_i}$ gets narrower, $\boldsymbol{K(x, x_i)}$ becomes wider; therefore, the score $\boldsymbol{f(x)}$ is mainly defined by the observations that are close to $\boldsymbol{x}$. The $\boldsymbol{n}$ factors $\boldsymbol{\alpha_i}$ (Lagrange multipliers) are the free coefficients which are the solution of an SVM optimization problem and have higher magnitudes for the observations at the boundary between the classes which are most relevant for classification (Moro, 2006).

2.5. Naïve Bayes

The Naïve Bayes classification method is based on the Bayesian theorem which postulated independence amongst the predictors. A Bayesian network represents a joint probability distribution over a set of discrete input variables $\boldsymbol{X_i}$. The following equation designs a Naïve Bayes classifier as;

$$\boldsymbol{P\left(Y = y_j \middle| X_1, \dots, X_n\right) = \prod_i P(X_i|Y = y_j)} \tag{5}$$

Assume a new occurrence that $\tilde{X} = (X_{1,\dots,}X_n)$, Eq.(5) shows the estimation of the probability that Y will take on any given value, given the observed input values of $\tilde{X}$ and the distributions of $P(Y)$ and $P(X_i|Y)$ estimated from the training set. Incorporating the above assumption, the Naïve Bayes classifier is given by,

$$Y_{NB} \leftarrow \underset{y_j}{\operatorname{argmax}} P(Y = y_j) \prod_i P(X_i|Y = y_j) \tag{6}$$

3. Data

The methods were applied on two distinct real-world credit datasets (loan performance data and loan application data). The first dataset is the loan performance dataset that composed of the default characteristics of 79254 granted loans in a Turkish financial institution(1) for the period between August, 2015 and September, 2017, of whom 71513 were good credits and 7741 were bad credits implying that the default rate is around 10%.

Due to the low proportions of defaulted observations, the accuracy rate expectedly remains high at almost 90% when all observations are used for model implementation, thus making it difficult to understand the importance of using deep learning algorithms. In order to keep a more balanced sample, we randomly selected 14303 observations from all non-defaulted credits, by this means preventing misrepresentation. The sample dataset in this case consists of 22044 observations and the ratio of bad credits is about 35%. The dataset includes one target variable (Default = 0, and Non-Default = 1) and 64 explanatory attributes (31 numeric and 33 categorical).

The second dataset is the application dataset that lists the characteristics of the 496196 loan applicants from the abovementioned institution between the period of January, 2014 and December, 2017. The dataset includes one target variable (Rejected = 0, and Approved = 1) and 60 explanatory attributes (22 numeric and 38 categorical). The number of approved loans was 257524 (52%) and 238972 (48%) of them were rejected applications. Thus, it is a more balanced sample.

The explanatory attributes in both datasets can be summarized into several groups such as:

- *demographical characteristics* (age, gender, education, disposable income, marital status, number of dependents, housing status, length of current residency etc.);
- *employment characteristics* (occupation, length of present employment, total employment etc.);

- *credit characteristics (*type, amount, maturity, frequency, collaterals, instalment rate, purpose etc.);
- *credit history* (number/amount of previous credits, number of previous defaults/late payments/prepayments, credit score etc.).

4. Model implementation and results

In model implementation process, we use 80% of the both dataset, randomly selected from the whole sample and referred to as the training set. The outstanding 20% is used for model evaluation purposes and is referred to as the out-of-sample test set. Extreme observations and missing data points were handled by using Elliptic Envelope (Rousseeuw and Driessen, 1999) and Soft Impute (Mazumder et al., 2010) methods, respectively.

Before the implementation of LR, J48, Naïve Bayes and SVM algorithms, a prior feature engineering process was carried out. For this purpose; redundant attributes were eliminated with the use of Information Value and Kolmogorov-Smirnov algorithms. The Principal Component Analysis (PCA) was employed to eliminate collinearity and extract the statistical factors that were used as final input variables in model estimation. In contrast, instead of using handcrafted input variables, we use raw features in the construction of the DNN model in order to allow the algorithm to extract all the information hidden in the deep levels of the datasets.

We implement the models, using Python programming language- specifically, the "scikit-learn" (Pedregosa et al., 2011) library for LR, J48, Naïve Bayes and SVM; "Tensorflow" (Abadi et al., 2015) and "Keras" (Cholet et al., 2015) libraries for the DNN. We evaluate the classification performance of each model through "weighted accuracy", "Type I Error" (misclassification of good loans) and "Type II Error" (misclassification of bad loans) rates obtained from confusion matrix which gives us a summary of prediction results on a classification problem (Table 1).

Table 1. *Confusion matrix*

			Actual Class	
			Good Loan	Bad Loan
Predicted Class		Good Loan	TP (True Positive)	FP (False Positive)
		Bad Loan	FN (False Negative)	TN (True Negative)

The "accuracy" is a common measure for evaluating a classification model's ability to discriminate between two classes. The "accuracy" is calculated by dividing all correctly classified instances (TP+TN) by all observations. However, due to imbalanced nature of the response variable and asymmetric misclassification cost (Type II Error yields more financial losses than Type I Error); we used an asymmetric accuracy rate that weights specificity rate three times higher than sensitivity rate. The weighted accuracy rate (WACC), in our case, is calculated as;

$$\mathrm{WACC} = 0.25 * \left(\frac{\mathrm{TP}}{\mathrm{TP+FN}}\right) + 0.75 * \left(\frac{\mathrm{TN}}{\mathrm{TN+FP}}\right) \tag{7}$$

4.1. Results for the loan performance data

In this part, DNN is compared with other models to predict the default characteristics of the loan borrowers in a loan performance dataset. As illustrated in Table 2, the proposed DNN model acquires a testing accuracy (WACC) of 77.98%, which is higher than the accuracy rates obtained by LR, J48 and Naive Bayes methods. However, SVM model performs slightly better than the DNN model due to its relatively lower Type II error.

When we look at the Type I and Type II error values, we can notice that J48 and Naive Bayes models are not suitable for the classification tasks. Since, J48 model yields highest Type I and Type II errors, while Naive Bayes predictions are heavily biased towards positive class as shown with imbalanced Type I (very low) and Type II (very high) errors.

Based on the Type I and Type II errors of the competing models, we can say that a lower Type I error indicates that DNN model has a smaller probability to misclassify a good borrower when predicting the loan defaults. And lower Type II error value represents that SVM is more powerful to identify the default behaviour.

Table 2. *The comparison between predicting models for the loan performance data*

	WACC	Type I Error	Type II Error
LR	77.31%	12.31%	26.15%
J48	70.05%	17.99%	33.94%
SVM	78.14%	11.51%	25.31%
Naive Bayes	57.04%	8.19%	54.55%
DNN	77.98%	10.20%	25.95%

4.2. Results for the loan application data

When dealing with the loan application data, we test the usefulness of our model to discriminate between the creditworthy and non-creditworthy applicants to decide whether approve or reject the loan application. The validation results are displayed in Table 3 below. At this model validation stage, the predictive ability of the DNN model is found to be 85.69%, which is significantly higher than the other models.

Moreover, the DNN also performs better than LR and SVM models in the aspects of sensitivity and specificity. The Type I error rate of 15.45% suggest that, there is a 15.45% chance that a creditworthy application will be misjudged and rejected. On the other hand, the Type II error rate of 13.92% implies that, there is a 13.92% probability that a highly risky and unworthy borrower will be accepted for credit.

Table 3. *The comparison between predicting models for the loan application data*

	WACC	Type I Error	Type II Error
LR	78.01%	16.67%	23.76%
J48	82.34%	17.91%	17.58%
SVM	77.93%	14.27%	24.67%
Naive Bayes	75.25%	90.00%	3.00%
DNN	85.69%	15.45%	13.92%

Looking at the results obtained from both datasets, one might conclude that deep learning models perform better than conventional models in bigger datasets. As the bigger sample size of the loan application dataset significantly improves the DNN model's performance over the other models in terms of accuracy, Type I and Type II error values.

5. Conclusion

In this paper, we have presented an application of Deep Neural Network (DNN) based classification model to credit risk assessment. The model has been trained and tested on two different data sets related to the characteristics of loan borrowers/applicants of a Turkish commercial bank. The performance of the DNN model was compared with four different predicting methods, namely LR, J48, Naïve Bayes and SVM.

Our experimental results indicated that, the DNN model significantly improves the performance of a credit scoring system relative to LR and SVM models in terms of balanced accuracy, Type I error and Type II error metrics in loan application dataset which has a larger sample. On the other hand, the DNN model does not significantly outperforms LR and SVM models in the loan performance dataset. Thus, linear classification models might be preferred in small sample datasets due to simplicity of their implementation. Furthermore, it is also known that deep learning algorithms are hard to implement and require a rigorous process of hyper-parameter tuning. Thus, deep learning based classification models are not always panacea, especially for datasets which have relatively small dimensions.

Future work is focused on both methodological and application issues. As to applications, we plan to assess the predicting abilities of the deep learning models by testing them on more complex data sets and to explore on the applicability. On the side of methodology, we are currently working on the design of other deep learning methods, such as classification Restricted Boltzmann Machine (classRBM) and Deep Belief Networks (DBN).

Note

[1] For confidentiality reasons, we are obliged to keep the name of the financial institution unpublished.

References

Abadi, M. et al., 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, Available at <tensorflow.org>

Abdou, H.A. and Pointon, J., 2011. Credit scoring, statistical techniques and evaluation criteria: a review of the literature. *Intelligent Systems in Accounting, Finance and Management*, *18*(2-3), pp. 59-88.

Altman, E.I., 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The journal of finance*, *23*(4), pp. 589-609.

Chollet, F. et al., 2015. Keras, GitHub, Available at <https://github.com/fchollet/keras>

Giesecke, K., Sirignano, J. and Sadhwani, A., 2016. Deep learning for mortgage risk. Working paper, Stanford University.

Hamori, S., Kawai, M., Kume, T., Murakami, Y. and Watanabe, C., 2018. Ensemble Learning or Deep Learning? Application to Default Risk Analysis. *Journal of Risk and Financial Management*, *11*(1), pp. 12.

Hand, D.J. and Henley, W.E., 1997. Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *160*(3), pp. 523-541

Hinton, G.E. and Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), pp. 504-507.

Kvamme, H., Sellereite, N., Aas, K. and Sjursen, S., 2018. Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications*, *102*, pp. 207-217.

Moro, R.A., 2006. *Estimating Probabilities of Default with Support Vector Machines*, Master's thesis, Humboldt-Universität zu Berlin, Wirtschaftswissenschaftliche Fakultät.

Larochelle, H., Mandel, M., Pascanu, R. and Bengio, Y., 2012. Learning algorithms for the classification restricted Boltzmann machine. *Journal of Machine Learning Research*, *13*(Mar), pp. 643-669.

Lessmann, S., Baesens, B., Seow, H.V. and Thomas, L.C., 2015. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, *247*(1), pp.124-136.

Louzada, F., Ara, A. and Fernandes, G.B., 2016. Classification methods applied to credit scoring: Systematic review and overall comparison. *Surveys in Operations Research and Management Science*, *21*(2), pp. 117-134.

Luo, C., Wu, D. and Wu, D., 2017. A deep learning approach for credit scoring using credit default swaps. *Engineering Applications of Artificial Intelligence*, *65*, pp. 465-470.

McCullagh, P. and Nelder, J.A., 1989. *Generalized linear models* (Vol. 37). CRC press.

Mazumder, R., Hastie, T. and Tibshirani, R., 2010. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, *11*, pp. 2287-2322.

Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, 12, pp. 2825-2830.

Quinlan, J.R., 1986. Induction of decision trees. *Machine learning*, *1*(1), pp. 81-106.

Ribeiro, B. and Lopes, N., 2011. Deep belief networks for financial prediction. In *International Conference on Neural Information Processing* (pp. 766-773). Springer, Berlin, Heidelberg.

Rousseeuw, P.J. and Driessen, K.V., 1999. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, *41*(3), pp. 212-223.

Tomczak, J.M. and Zięba, M., 2015. Classification Restricted Boltzmann Machine for comprehensible credit scoring model. *Expert Systems with Applications*, *42*(4), pp. 1789-1796.

West, D., 2000. Neural network credit scoring models. *Computers & Operations Research*, *27*(11-12), pp. 1131-1152.