

Performance evaluation of fourteen machine learning algorithms on credit card default classification

Hussein Altabrawee

Assistant lecturer - The College Of Engineering - Almutanna University
hussein.a.hassan@live.com

Abstract

Banks process their financial data by machine learning techniques to get knowledge from the data and use that knowledge in decision making and risk management. In this research, fourteen classification models have been built and trained using a real financial data from a bank in Taiwan. The models forecast the credit card default of a customer which is the repayment delay of the credit granted to the customer. The main idea of the research is evaluating and comparing the models based on their predictive average class accuracy.

ملخص

تقوم البنوك بمعالجة بياناتها المالية بواسطة تقنيات (machine learning) لغرض الحصول على المعلومات و المعرفة من البيانات. تستخدم البنوك تلك المعلومات و المعرفة في عملية صنع القرار و ادارة المخاطر. في هذا البحث، تم بناء و تدريب أربعة عشرة نموذج يستخدم للتصنيف باستخدام بيانات حقيقية لحد البنوك في تايوان. النماذج تتنبأ بالتأخير الذي ممكن ان يحصل في تسديد الاموال المستحقة للبنك من قبل العملاء عند استخدام بطاقات الائتمان. في هذا البحث، تتم مقارنة و تقييم لاداء النماذج بالاعتماد على مقياس (average class accuracy).

keywords: tree based calssification; rule based classification; naïve bayes; ann;dnn;banking

1. Introduction

We live in the information age, all the world's major companies and organizations produce tremendous amount of data every year. They turn the data into knowledge by using data mining and machine learning techniques. They make use of that knowledge in decision making and risk management. For example, banks use their financial data to predict the risk of the customer credit. By using machine learning techniques, banks can classify a new credit card applicant into risky or non-risky applicant and based on that banks accept or reject the application. Another approach is using the bank financial data to forecast the applicant default probability which is the probability of the repayment delay of the credit granted to the customer. This approach is more helpful and meaningful than the first approach [1]. There are many machine learning and data mining techniques and algorithms that are used in order to build prediction models. Those algorithms can be categorized into information-based learning algorithms, similarity-based learning algorithms, probability-based learning algorithms, and error-based learning algorithms [2]. A good predictive model has to capture the relationship between the input features and the target feature and avoid underfitting and overfitting. In the case of underfitting, the model fails to capture the relationship because it is too simple. On the other hand, overfitting can occur when the model fits the training data very well because it is too complex and fails to generalize to the new data [2]. Overfitting and underfitting can be avoided by using the right amount of regularization in the model and by using the right model's parameters. No regularization or small regularization amount can cause the model to be overfitted. On contrast, using a large amount of regularization can cause the model to be underfitted. In this paper, fourteen classifiers are built to predict the default of the credit card holders. Each one of the models has different set of parameters and configurations than the other models. The models are compared against one another based on their predicative average class accuracy. The dataset used in this experiment is a real financial data of a bank in Taiwan which is the same dataset used in [1]. The dataset is obtained from the UC Irvine Machine Learning Repository.

2. Literature Review

2.1 Related Work

Many research have been done on building predicative models using credit card dataset. Those models can be used to classify the credit card applicants into risky or non-risky, to forecast the probability of default of credit card customers' payments, or to forecast the churn of the customers. Yeh and Lien [1] have built predicative classifiers to forecast the payment's default probability of credit card customers. They compared the predicative accuracy of six data mining techniques used to build the models. They used KNN, LR, DA, NB, ANN, and classification trees in order to build the models. They introduced a novel sorting and smoothing method to estimate the real probability of default. They found that the model built by using the artificial neural network is the only model that can forecast the real probability of default. Wah and Ibrahim [3] has built three predicative models using three data mining techniques. They used neural networks, CART, and logistic regression to build classifiers that predict the credit scoring, credit risk, which associated with credit card application. They compared between the three techniques and they found that the neural network model has a slightly better classification accuracy. Kambal et al. [4] have built credit scoring models for the Sudanese credit dataset using artificial neural network and decision trees as main data mining techniques. In addition, they used PCA and Genetic Algorithms as features selection methods. They showed that the model built by the artificial neural network with Genetic Algorithms has the highest accuracy and outperformed all the other models. Kou et al. [5] have built twelve classification models to predict the churn of credit card customers. Those models can be categorized into models based on decision trees algorithms, function based algorithms, models based on Bayes, models based on clustering, and rule based models. Kou et al have ranked and compared the twelve algorithms used PROMETHEE and TOPSIS methods. They showed that Logistic regression and J48 algorithms are more efficient algorithms to build classification models for churn analysis. Another work on credit dataset has been done by Embong et al. [6] to build a credit risk classification model based on German credit dataset. They compared two well-known classification techniques, kernel logistic regression and support vector machine. They found that using kernel logistic regression is better than using support vector machine for the dataset.

2.2. Machine Learning Techniques

The following is an overview of the machine learning techniques that are used in this experiment.

2.2.1 PART

PART is an elegant, efficient and accurate machine learning algorithm for deriving learning rules by generate partial decision trees repeatedly. Its main idea based on C4.5 algorithm and RIPPER learning scheme. PART can learn a very good rule set without using a global optimization by learning one rule at a time. It uses two well-known learning rules inferring techniques, decision trees technique and separate and conquer technique. PART overcomes the huge slow performance of the C4.5 algorithm on pathological datasets by not using post processing steps. [7]

2.2.2 DTNB

DTNB is a simple efficient machine learning algorithm that combines two learning techniques, Decision Table and Naïve Bayes. It divides the attributes into two parts. In one part, DTNB uses Naïve Bayes technique in order to assign class label probabilities. In the other part, DTNB uses Decision Table technique in order to assign class label probabilities. Then, the algorithm combines the two probability estimations. Based on experiments showed in [8] DTNB outperforms both Naïve Bayes and Decision Table. [8]

2.2.3 Hoeffding Trees

Hoeffding Trees is a decision tree based learning algorithm that produces an online, incremental, learning model that learns in small fixed time per every data row. An online model runs continuously and without any limitation and processes any new data examples as they arrive. Hoeffding Trees can process and learn from large data streams. When there are enough data examples, Hoeffding Trees can produce a tree model that is very similar to the trees produced by a typical batch learning algorithms. As the number of processed data examples increases the similarity between Hoeffding Trees and the conventional trees increases exponentially. The idea behind Hoeffding Trees is choosing an optimal splitting

feature by using only a small sample of the data and that is enough in most cases. Mathematically, this idea can be supported by the Hoeffding bound. [9]

2.2.4 RIPPER

RIPPER is an efficient rule learning algorithm proposed by William W. Cohen as an improved version of IREP algorithm. RIPPER has lower error rates than IREP algorithm and its error rates is very similar to C4.5 rules error rates. The relationship between the error rates and the number of the training data examples is linear. RIPPER is better than C4.5 rules since it can handle noisy large data efficiently. The RIPPER improvements, over IREP, are using different measure for determining the value of the rules at the phase of pruning, a new stopping condition that determines when to stop adding more rules to the set of rules, and finally adding a rule optimization process. [10]

2.2.5 LADTree

LADTree, Least Absolute Deviation Trees, can be viewed as a combination of the boosting predictive accuracy and the decision trees and that results into set of a classification rules. Next, they are adapted to AdaBoost and multiclass LogitBoost. LogitBoost is fused with AdaBoost. This structure is used in order to build a multi-class alternating decision trees. The algorithm selects single variable for the splitter node. The goal is minimizing the least squares between the examples' mean value and the working return. [11]

2.2.6 C4.5 consolidated tree

In this classification method, C4.5 consolidated tree is built by using the Consolidated Tree Construction, CTC, algorithm. CTC algorithm is a method used to fix the high imbalanced class problem in the classification task. CTC produces a single, simple, understandable, and self-explaining decision tree. First, CTC makes a group of subsamples that is taken from a training sample. It uses each one of the subsamples to create a decision tree. This process is similar to Bagging except it executes the ensemble procedure during creating the tree by voting in order to select the split on each node of the tree. All trees participate in the voting process. All the trees make the same exact split decision that is made by the majority voting.

C4.5 consolidated tree is more stable and simpler than the C4.5 tree which is based on. [12]

2.2.7 Artificial Neural Network (ANN)

An ANN is a computational model that consists of many computing unites, neurons, which are connected to each other. The ANN could be viewed as a directed graph, cyclic or acyclic graph. The nodes in the graph represent neurons while the edges represent the links that connect the neurons. ANN can be constructed of many layers and each one of them can have many neurons. The first layer is the input layer and the last layer is the output layer. The rest of the layers are the hidden layers. Each one of the neurons, except the neurons in the input layer, gets as an input the weighted sum of outputs of neurons that are connected to its incoming edges. A well-known heuristic used to train the ANNs depends on the SGD framework. Back Propagation algorithm is a widely used ANN training algorithm. [13]

2.2.8 Deep Neural Network (DNN)

DNN is an artificial neural network that simulates the deep human brain architecture by having many hidden layers. Using deep architecture is necessary in many fields since learning deep neural networks needs much less computational power and much less training data than learning shallow ANNs. That is because a DNN requires exponential less neurons and exponential less training data than a shallow neural network having one less layer in order to do the same function. [14] Deep neural networks have provided an excellent practical performance on many different domains such as convolutional networks, restricted Boltzmann machines, auto-encoders, and sum-product networks. [13]

2.2.9 Support Vector machine

Support Vector machine, SVM, is one of the most useful machine learning algorithms that is used to build and learn linear prediction models in high dimension feature space. Working on feature space with high dimensions causes two challenges, computational complexity and sample complexity. SVM algorithm searches for a large margin decision boundary, separating hyperplane, in order to overcome the sample complexity challenge. Hard-SVM searches for the decision

boundary that separates the training data perfectly with the largest margin. In contrast, Soft-SVM is based on the assumption that the learning data is not perfectly separable. In order to solve the problem of computational complexity, SVM algorithm is used with Kernels which are similarity measures between data instances. [13]

2.2.10 Naive Bayes classifiers

Naive Bayes classifier is a well-known, simple, and clear supervised machine learning algorithm that can learn and represent probabilistic knowledge. Bayes rule is used in order to calculate the probability of every class when given the data examples. Naive Bayes algorithm is based on two simplifying assumptions. The first assumption states that the data features used in the prediction are conditionally independent when given the class label. The second assumption states that there is no hidden features or latent features that could affect the process of prediction. [15]

2.2.11 C4.5 Algorithm

C4.5 algorithm is the enhanced version of the ID3 machine learning algorithm. C4.5 produces univariate decision trees that can be used as classification models. C4.5 uses the information entropy in order to create the decision tree from the training dataset. At each tree node, C4.5 algorithm selects only one data feature that is used in order to divide the sample set at that node into subsets. Most of the data samples in these subsets are having only one class. The process is based on the normalized information gain criterion. This criterion is calculated from selecting a data feature that is used for dividing the data. C4.5 chooses the data feature that has the highest normalized information gain in order to make the decision. The process of building the tree continues by performing further subset dividing operation. The goal is making the smallest data subsets that contains all the data samples that are belong to only a single class. [16]

2.2.12 ExtraTree

Extremely randomized tree is an efficient machine learning algorithm for tree based induction that can be used to build a supervised classification model. Extremely randomized tree chooses the splits attribute and the splits cut-point totally

at random or partially at random when splitting a tree node. When using this algorithm, the smoothing degree and attribute randomization strength can be controlled using the algorithm parameters. Extra Tree algorithm decreases the variance and increases the bias at the same time. The amount of decreasing and increasing depends on the selected randomization level of the algorithm. The algorithm has been used in many applications such as image classification, mass-spectrometry datasets, time series classification, and reinforcement learning. [17]

2.2.13 NBTree

NBTree is a hybrid machine learning algorithm that combines two classification techniques, Naïve Bayes and decision trees. It produces a decision tree in which each tree node contains univariate splits while each leaf contains a Naïve Bayes classifier. The decision tree produced by the algorithm is similar to Utgoff Perceptron tree yet the induction steps are different. The data is segmented by the decision tree and by that the Naïve Bayes conditional independence assumptions are likely to be met. One tree leaf represents only one data segment which is described by Naïve Bayes classifier. [18]

2.2.14 BayesNet

Bayesian Network is a graph based classifier and a probability based classifier because it is made of an acyclic graph and a table of conditional probability. In Bayesian Network, the uncertain domain knowledge is represented by the graph structure. Random variables are represented by the nodes in the graph. The dependencies of probability between the random variables are represented by edges that connect the nodes in the graph. Bayesian Network uses well-known computational and statistical procedures in order to estimate the conditional dependences. The first step in the learning process is identifying the network structure. The second step in the learning process is learning the probability tables. [16]

3. The Experiment

3.1 The Dataset

The dataset used in this research is obtained from the UC Irvine machine learning repository. It is the same dataset used in [1]. It is a financial data, actual credit card holders' data, gathered from an important bank in Taiwan at October 2005. The dataset contains 30000 instances. The instances with the target attribute value of 0 represent the customers with no default payments while the instances with the target attribute value of 1 represent the customers with default payments. There are 6636 instances with default payments and that equals 22.12 % of the data. The dataset is divided into three parts. They are the training part, cross validation part, and the test part. The training data contains 18000 instances. There are 4150 instances with default payments and that equals 23.056 % of the training data. The training data is used to train the predictive models. The cross validation data contains 6000 instances. There are 1220 instances with default payments and that equals 20.33 % of the cross validation data. The cross validation data is used to compare between the different predictive models. The model with the highest accuracy measured on the cross validation data is the best model. The test data contains 6000 instances. There are 1266 instances with default payments and that equals 21.1 % of the test data. The test data is used to calculate and report the accuracy of the best model. There are 23 descriptive attributes in the dataset that describe the following: the amount of the given credit, the status history of the past payments for six months starting from April to September 2005, gender, the value of the bank statements for six months from April to September, education, marital status, age, and the amount of the previous payments for the six months.

3.2 Measures of Accuracy

As mentioned earlier, the dataset is unbalanced because there are much more instances without defaulted payments. In order to get an accurate classification accuracy for the unbalanced data, the harmonic mean average class accuracy is the best measure to use [2]. In addition, recall, precision, and F1 score are calculated as the following equations:

$$\text{Average Class Accuracy}_{HM} = \frac{1}{\frac{1}{|Levels(t)|} \sum_{l \in Levels(t)} \frac{1}{Recall_l}} \quad (1)$$

Where the recall equals:

$$Recall = \frac{TP}{(TP + FN)} \quad (2)$$

In addition to these measures, the following measures are calculated in order to get better insight of the accuracy.

$$Precision = \frac{TP}{(TP + FP)} \quad (3)$$

$$F1 \text{ Score} = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (4)$$

TP, True Positives, represents the number of instances in the test dataset which had a positive target feature value and that were predicted to have a positive target feature value. TN, True Negatives, represents instances in the test dataset that had a negative target feature value and that were predicted to have a negative target feature value. FP, False Positives, represents instances in the test dataset which had a negative target feature value but that were predicted to have a positive target feature value. FN, False Negative, represents instances in the test dataset that had a positive target feature value but that was predicted to have a negative target feature value. [2]

3.3 The Results

Fourteen classification models have been built and trained using Matlab programming Language and machine learning software such as RapidMiner and Weka. The models have been tested using the cross validation dataset which is used to select the best model. Then each model has been tested using the Test dataset which is used to find the final accuracy for each model. Table 1 shows the average class accuracy with harmonic mean for the fourteen classification models.

Based on Tabel 1, PART classification technique outperforms all the other techniques. In general, the rule based techniques, the tree based techniques, and BayesNet achieved very good results and outperforms ANN, DNN, SVM, and NaiveBayes. NBTree which is a hybrid tree algorithm has the lowest classification accuracy among all the techniques. Table 2 shows the models average class accuracy achieved on the Test dataset.

Table 1
The average class accuracy on the Cross Validation Dataset

N o	Classification Technique	Precision	Recall	F Score	Average Accuracy
1	PART	0.48679514 6	0.55901639 3	0.52041205 6	0.674330138
2	DTNP	0.45448505	0.56065573 8	0.50201834 9	0.668672387
3	C4.5 Consolidated	0.47622427 3	0.55	0.51046025 1	0.666496777
4	BayesNet	0.49923195 1	0.53278688 5	0.51546391 8	0.659007069
5	ExtraTree	0.45767575 3	0.52295082	0.48814078	0.645140798
6	C4.5	0.51782682 5	0.5	0.50875729 8	0.637988488
7	Hoeffding Tree	0.50164203 6	0.50081967 2	0.50123051 7	0.636499643
8	JRIP	0.50164203 6	0.50081967 2	0.50123051 7	0.636499643
9	LADTree	0.51896551 7	0.49344262 3	0.50588235 3	0.63316327
10	ANN	0.26634528 5	0.72459016 4	0.38951310 9	0.585057053
11	DNN	0.27096774 2	0.55081967 2	0.36324324 3	0.584142732
12	SVM	0.27150974	0.54836065 6	0.36319218 2	0.583948874
13	NaiveBayes	0.26906318 1	0.80983606 6	0.40392477 5	0.568933042
14	NBTree	0.70631970 3	0.15573770 5	0.25520483 5	0.268894636

Table 2

The average class accuracy on the Test Dataset

	Classification Technique	Precision	Recall	F Score	Average Accuracy
1	DTNP	0.473924381	0.574249605	0.519285714	0.678678505
2	PART	0.494018297	0.55450237	0.522515817	0.67057899
3	C4.5	0.485457064	0.55371248	0.517343173	0.668413382
	Consolidated				
4	ExtraTree	0.468814256	0.54028436	0.502018349	0.656462246
5	BayesNet	0.499252616	0.52764613	0.513056836	0.653579439
6	Hoeffding Tree	0.511829653	0.512638231	0.512233623	0.644928565
7	JRIP	0.511829653	0.512638231	0.512233623	0.644928565
8	C4.5	0.526970954	0.501579779	0.513961959	0.63885754
9	LADTree	0.529756915	0.499210111	0.514030094	0.637429839
10	SVM	0.274071115	0.541864139	0.364022287	0.57663787
11	ANN	0.264408794	0.70300158	0.384283247	0.568340449
12	NaiveBayes	0.277506775	0.808846761	0.413236481	0.567295998
13	DNN	0.266480224	0.52685624	0.353940037	0.566316939
14	NBTree	0.726923077	0.1492891	0.247706422	0.259280997

The Conclusion

In this research, fourteen machine learning techniques are compared based on their average class accuracy to predict the default of credit card customers. The research showed that the rule based models, DTNP and PART, outperformed all the other classification models and achieved the highest accuracy. Tree based models and BayesNet achieved good accuracy that is close to the accuracy achieved by the rule based models. ANN, DNN, Naïve Bayes, and SVM have a lower accuracy than the tree based models. NBTree has the lowest accuracy among all models.

References

- [1] I. Yeh, C. Lien, The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients, *Expert Syst. Appl.* 36 (2009) 2473–2480. doi:10.1016/j.eswa.2007.12.020.
- [2] J.D. Kelleher, B. Mac Namee, A. D'Arcy, *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*, 1st ed., MIT Press, Cambridge, Massachusetts, 2015. <https://mitpress.mit.edu/books/fundamentals-machine-learning-predictive-data-analytics>.
- [3] Y.B. Wah, I.R. Ibrahim, Using Data Mining Predictive Models to Classify Credit Card Applicants, in: 6th Int. Conf. Adv. Inf. Manag. Serv. (IMS), 2010, IEEE, Seoul, Korea, 2010: pp. 394–398. <http://ieeexplore.ieee.org/document/5713481/>.
- [4] E. Kambal, I. Osman, M. Taha, N. Mohammed, S. Mohammed, Credit scoring using data mining techniques with particular reference to Sudanese banks, in: 2013 Int. Conf. Comput. Electr. Electron. Eng., IEEE, 2013: pp. 378–383. doi:10.1109/ICCEE.2013.6633966.
- [5] G. Wang, L. Liu, Y. Peng, G. Nie, G. Kou, Y. Shi, Predicting Credit Card Holder Churn in Banks of China Using Data Mining and MCDM, in: 2010 IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol., IEEE, 2010: pp. 215–218. doi:10.1109/WI-IAT.2010.237.
- [6] S.P. Rahayu, S.W. Purnami, A. Embong, Applying Kernel Logistic Regression in data mining to classify credit risk, in: 2008 Int. Symp. Inf. Technol., IEEE, 2008: pp. 1–6. doi:10.1109/ITSIM.2008.4631725.
- [7] E. Frank, I.H. Witten, Generating accurate rule sets without global optimization, in: *Proceeding ICML '98 Proc. Fifteenth Int. Conf. Mach. Learn.*, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA ©1998, 1998: pp. 144–151. <http://dl.acm.org/citation.cfm?id=657305>.
- [8] M. Hall, E. Frank, Combining Naive Bayes and Decision Tables, in: *Proc. Twenty-First Int. FLAIRS Conf.*, 2008: p. vol. 2118, 318–319. <https://www.aaai.org/Papers/FLAIRS/2008/FLAIRS08-076.pdf>.
- [9] P. Domingos, G. Hulten, Mining high-speed data streams, in: *Proc. Sixth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '00*, ACM Press, New York, New York, USA, 2000: pp. 71–80. doi:10.1145/347090.347107.
- [10] W.W. Cohen, Fast Effective Rule Induction, in: *Proc. Twelfth Int. Conf. Mach. Learn.*, Morgan Kaufmann, 2016, Tahoe City, California, 1995: pp. 115–123. https://books.google.com/books?id=akjBQAAQBAJ&dq=Fast+Effective+Rule+Induction&lr=&source=gbs_navlinks_s.
- [11] E. Turanoglu-Bekar, G. Ulutagay, S. Kantarcı-Savas, Classification of Thyroid Disease by Using Data Mining Models: A Comparison of Decision Tree Algorithms, *Oxford J. Intell. Decis. Data Sci.* 2016 (2016) 13–28. doi:10.5899/2016/ojids-00002.

- [12] I. Ibarguren, J.M. Pérez, J. Muguerza, I. Gurrutxaga, O. Arbelaitz, Coverage-based resampling: Building robust consolidated decision trees, *Knowledge-Based Syst.* 79 (2015) 51–67. doi:10.1016/j.knosys.2014.12.023.
- [13] S. Shalev-Shwartz, S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014. <http://www.cambridge.org/us/academic/subjects/computer-science/pattern-recognition-and-machine-learning/understanding-machine-learning-theory-algorithms?format=HB&isbn=9781107057135>.
- [14] H. Wang, B. Raj, A Survey: Time Travel in Deep Learning Space: An Introduction to Deep Learning Models and How Deep Learning Models Evolved from the Initial Ideas, *CoRR*. abs/1510.0 (2015) 1–43. <http://arxiv.org/abs/1510.04781>.
- [15] G.H. John, P. Langley, Estimating continuous distributions in Bayesian classifiers, in: *Proceeding UAI'95 Proc. Elev. Conf. Uncertain. Artif. Intell.*, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA ©1995, Montréal, Qué, Canada, 1995: pp. 338–345. <http://dl.acm.org/citation.cfm?id=2074196>.
- [16] K. Singh, S. Agrawal, Performance Evaluation of Five Machine Learning Algorithms and Three Feature Selection Algorithms for IP Traffic Classification, *Int. J. Comput. Appl.* 1 (2011) 25–32. <http://www.ijcaonline.org/specialissues/encc/number1/3716-encc005>.
- [17] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, *Mach. Learn.* 63 (2006) 3–42. doi:10.1007/s10994-006-6226-1.
- [18] R. Kohavi, Scaling Up the accuracy of Naive-Bayes Classifiers : A Decision Tree Hybrid, in: *Proceeding KDD'96 Proc. Second Int. Conf. Knowl. Discov. Data Min.*, AAAI Press, Portland, Oregon, 1996: pp. 202–207. <http://dl.acm.org/citation.cfm?id=3001502>.