The Comparison of Classification Algorithms in Detecting Credit Card Default Risk

Di-Heng Liu
M.S. Candidate - Data Mining and Predictive Analytics
St. John's University
diheng.liu17@stjohns.edu

Drew-Elliot Ramsingh
M.S. Candidate – Data Mining and Predictive Analytics
St. John's University
drewelliot.ramsingh12@stjohns.edu

Figure 1: U.S. credit card default rates are the highest in five years [1]. The results of this study can be used to implement measures to mitigate the problem.

Abstract

This study performs several data mining algorithms on a UCI Machine Learning Repository dataset to evaluate an optimal model for predicting credit card payment default. The dataset segments customers' credit card accounts as risky or non-risky based on the amount of the given credit, gender, education, marital status, age, history of past payment, amount of bill statement, and amount of previous payment. The algorithms utilize the 24 features to produce the optimal model based on performance metrics. When a customer is classified to be risky, creditors and lenders can take further actions to investigate the actual personal conditions of the customer and hence take preventive actions. However, when a credit cardholder is incorrectly classified to default, or to be in good status when he/she is due to default, revenue is lost by the creditors and lenders. Therefore, we demonstrate ways of mitigating such risks with data mining techniques to predict customer defaults based on historical data from client observations. If the target label Y is 1, then the account will be treated as non-credible.

Keywords: Credit card, classification, default, KNN, Neural Network, SVM

Introduction

Credit cards are usually issued by banks, corporations and financial institutions so that clients can purchase goods and services on their credit. People can have multiple credit cards from different companies. Meanwhile, companies that evaluate credit scores usually suggest cardholders utilize multiple credit cards to construct client data (history) and at the same time increase client credit score. Client data usually includes their basic economic capability information and most importantly their repayment habits, which indicate the credibility of a person. Due to a timely repayment habit, a person will generally receive a good credit score by the lenders and creditors which can be used for future credit issue [7]. The purpose of this study is to analyze through Python how to classify credit card default risk from a large volume of data.

Yulia et al. (2016) stated that knowing predictors that significantly contribute to default prediction is an emerging issue of credit risk analysis. Their paper focused on using regularization for feature selection and logistic regression for classification, none of which will be used for this study. Default prediction and default predictor selection are two related issues, but many existing approaches address them separately [11]. There needs to be a merge of these two issues where a feature selection process is used before the execution of models. As a preprocessing measure in this study, F-values and p-values for each feature will be computed in a statistical test of significance with the target label. Additionally, a manipulation of a random forest classifier will be used to provide the most important features with regards to the target attribute.

K-nearest neighbors (KNN), a commonly used method of data classification, is also a machine learning algorithm. The main concept of KNN is quite intuitive as each observation is a coordinate position in a multi-dimensional space, which means all the observations are coordinates. KNN became one of the most significant algorithms in classification because of its extension of formula and broad range of applications. One such application is graph construction which is used in similarity search and collaborative filtering [2], which may assist in distinguishing high-risk default people. Another such application is keyword-based search in spatial data [3], which can be used to search for specific feature values, for example, clients who are divorced. In the range of applications, KNN is frequently employed on molecular biology, medical imaging and online multimedia data [4].

Support vector machines (SVM) is also a commonly used method for data classification, but it is unlike KNN. SVM is a method used only for supervised learning, which means it needs to be trained before generating a model. Its basic concept is that in the two-dimensional space, there must be a hyperplane that can clearly separate the two categories. SVM is a well-established machine learning methodology popularly used for classification, regression and ranking [8]. It has met with significant success in numerous real-world learning tasks [10]. In recent years, SVM has been proposed to solve pattern recognition and function approximation problems due to their superior performance [9].

With regards to the problem to be solved, Yeh and Lien (2009) compared six data mining techniques for the predictive accuracy of probability of default of credit

card clients [5]. These techniques were discriminant analysis, logistic regression, Bayes classifier, nearest neighbor, artificial neural networks and classification trees, of which, artificial neural networks showed the best performance based on coefficient of determination. Li-Hua et al. (2017) also conducted a similar study on micro-lending default awareness using an artificial neural network [6]. They compared the network's performance to logistic regression and was able to produce an average accuracy of 75.13%. For this study however, only three previously mentioned techniques will be used, which are KNN, SVM and artificial neural network.

Methodology

The data consisted of credit cardholders' information from a major Taiwanese cash and credit card issuer [5]. There were 30,000 observations among which 6,636 (22.12%) were cardholders who defaulted on their payment. The attributes in Table 1 were used to explain the response variable of default payment.

To avoid any effect of outlying values in the classification methods, a scaled dataset was created using the min-max formula on each attribute:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

This preprocessing step scaled all the dataset's values between 0 and 1, rounded to 6 decimal places.

Using the original data, an analysis was performed for feature selection using F and p-values of significance tests. Each attribute was tested with the target variable. In addition, a random forest classifier provided a sorted list of important features. A k-means model in combination with principal component analysis was then used to distinguish groups within the dataset. An elbow method was employed for both the original and scaled datasets and visualizations were provided through plots.

Attribute	Description
X1	Amount of credit issued in New Taiwan dollars (NTD) including individual consumer and family credit
X2	Gender
X3	Education
X4	Marital status
X5	Age
X6-X11	Repayment status for the previous 6 months
X12-X17	Statement balances for the previous 6 months
X18-X23	Payment amounts for the previous 6 months

Table 1: Attribute Information

For classification, K-NN, SVM and neural network were imported from the Scikit-learn Python library. They were executed using both the holdout method and cross validation for comparative analyses and metrics. For the holdout method, the test size was placed at 0.33. A 10-fold cross validation was used for KNN and neural network, while for SVM, a smaller number of 3-folds was needed for processing. Quite notably, SVM required a longer time for computation. With regards to K-NN, a loop was utilized to determine the number of neighbors with the highest cross validation accuracy in a given range. Odd numbers were preferred to elude ties.

The non-linear radial basis function (rbf) kernel was used in the support vector classification where it can be effective in the high dimensional space. Unfortunately, the time the model takes to fit data increases with the number of samples. Hence, this will make it difficult to scale to the dataset as there are more than 10000 samples. This explains the use of a lower number of folds for cross validation.

A multi-layered perceptron classifier was used in building the neural network. The documentation on the default solver "adam", a stochastic gradient-based optimizer, stated great performance on large datasets with thousands of training samples. The "lbfgs' solver which is an optimizer in the family of quasi-Newton methods was also used in comparison. The documentation stated this as having better convergence and performance for small datasets. Other parameters for this classifier included a hidden layer size of 10 and logistic activation due to the binary response. With 10 as the number of neurons in the hidden layer, the probability of overfitting was lowered, and computation time was optimized. The logistic function worked best as there are only two target values of default and non-default (0 and 1).

The performance metrics for the algorithms were accuracy, mean squared error, precision and recall. For cross validation, the mean of the scores were given as the accuracy.

Results

With regards to feature selection, there was not enough evidence from the F-tests of significance and the associated p-values in Table 2 to determine that an attribute was not meaningful in explaining the client default.

Attributes	F-Values	P-Value
X1	724.068539	1.30E-157
X2	47.9788543	4.40E-12
X3	23.5471118	1.23E-06
X4	17.7812714	2.49E-05
X5	5.78855582	1.61E-02
X6	3537.71497	0.00E+00
X7	2239.16914	0.00E+00
X8	1757.46644	0.00E+00
X9	1476.84597	1.89929659e-315
X10	1304.59118	1.13E-279
X11	1085.40249	7.30E-234
X12	11.5805315	6.67E-04
X13	6.04423789	1.40E-02
X14	5.94438771	1.48E-02
X15	3.09474518	7.86E-02
X16	1.3710874	2.42E-01
X17	0.865820292	3.52E-01
X18	160.40381	1.15E-36
X19	103.291524	3.17E-24
X20	95.2180109	1.84E-22
X21	97.1880005	6.83E-23
X22	91.4298008	1.24E-21
X23	85.0890453	3.03E-20

Table 2: Results of Significance Test

A client's repayment status in the month of September 2005 (X6) returned the highest F-value and lowest p-value. Interpreting these statistics, the attribute has the highest significance in relation to the target attribute. On the other hand, a client's balance in April 2005 (X17) returned the lowest F-value and highest p-value. The interpretation for this attribute and its statistics is that there is a 0.35 probability that this attribute is not significant to the target. However, the results of the random forest classifier in Figure 1 and Table 3 were also used to decide the removal of features before use in models. The random forest classifier ranked the features in order of the best split.

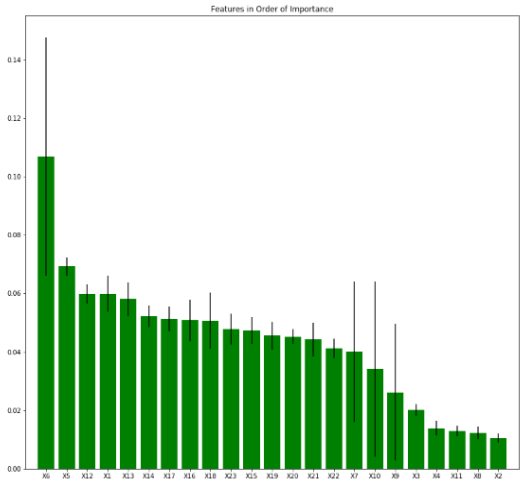

Figure 1: Attributes in Order of Importance

No	Attribute	Percentage
1	X6	-0.08771
2	X5	-0.068534
3	X12	-0.061375
4	X1	-0.057906
5	X18	-0.056705
6	X13	-0.053722
7	X15	-0.05088
8	X13	-0.049404
9	X17	-0.049131
10	X16	-0.048376
11	X19	-0.047038
12	X23	-0.046927
13	X7	-0.045477
14	X20	-0.045347
15	X22	-0.045226
16	X21	-0.04317
17	X8	-0.039622
18	X10	-0.026367
19	X3	-0.02052
20	X11	-0.016189
21	X9	-0.014193
22	X4	-0.014138
23	X2	-0.012043

Table 3: Attributes in Order of Best Split

From Table 3, the repayment status in September 2005 (X6) was determined to provide the most information gain with regards to the target attribute based on Gini impurity and entropy. However, in this method of feature selection, gender (X2) was determined to provide the least information gain as a splitting attribute. Compared to the significance test, the attribute gender had high F-value of 47.97 and a very low p-value of 4.40×10^{-12}, which translates to a very low probability that the attribute has no significance to the target attribute. With regards to the lowest scoring attribute from the significance test, that is, the client's balance in April 2005 (X17) was the 9th important attribute in the random forest classifier. Therefore, there are no clear attributes that can be removed using these two methods of feature selection. All the attributes were then used in the classification algorithms as they were evaluated to be significant or important.

The results from k-means clustering and principal component analysis showed that the data can be distinguished into "default" and "non-default" groups using only the scaled data. Analysis on the original data showed that when using the optimal cluster number of 3 (Figure 2), the flattened dimensions in PCA 1 and PCA 2 were able to provide distinguishable groups (Figure 4). However, a further plot from the original data using PCA 2 and PCA 3 (Figure 5), showed that the clusters could not be distinguished from each other. However, using the scaled data and the optimal cluster number of 2 (Figure 3), very distinguished groups were formed. PCA 1 on the x-axis in Figure 7 displayed approximately 0.6 units between these groups. It was interesting to note that there were also further separations within the classified groups as seen in Figure 8. Further component analysis beyond PCA 3 for the scaled data did not provide distinguishable

groups. These results indicated that through dimensionality reduction, the scaled data could be correctly classified into two labels (default and non-default) and provided motivation to proceed with predictive models.

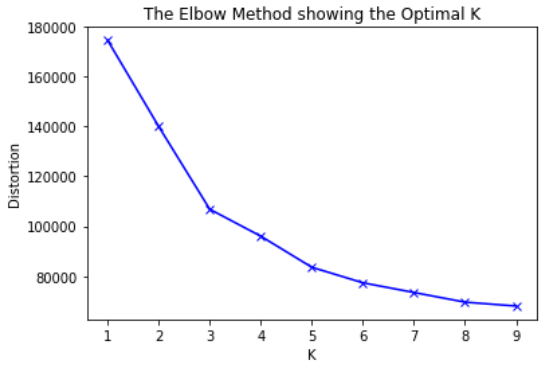

Figure 2: Elbow Method for Optimal K on Original Data

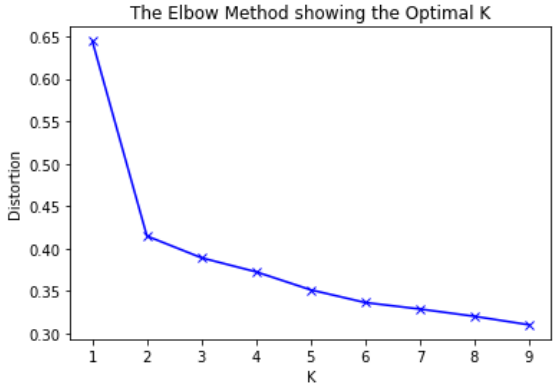

Figure 3: Elbow Method for Optimal K on Original Data

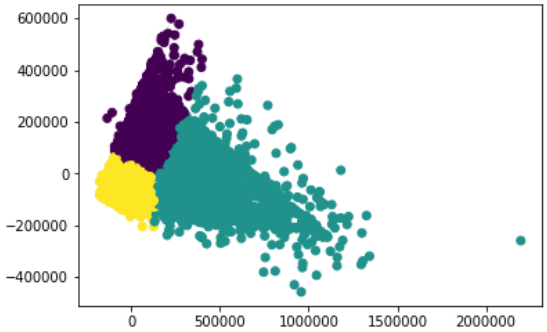

Figure 5: Scatterplot of PCA 1 and PCA 2 using Original Data

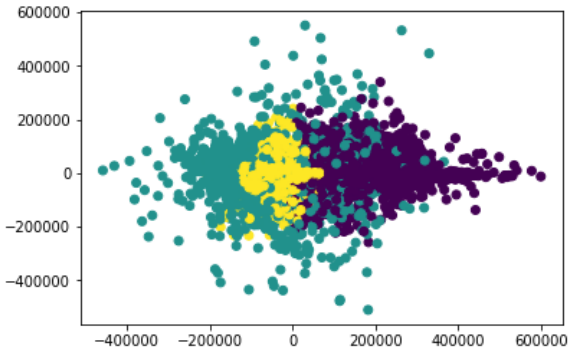

Figure 6: Scatterplot of PCA 2 and PCA 3 using Original Data

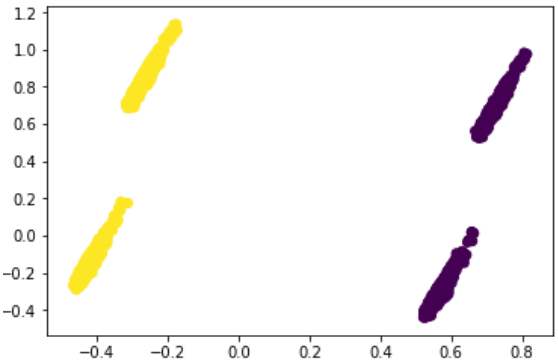

Figure 7: Scatterplot of PCA 1 and PCA 2 using Scaled Data

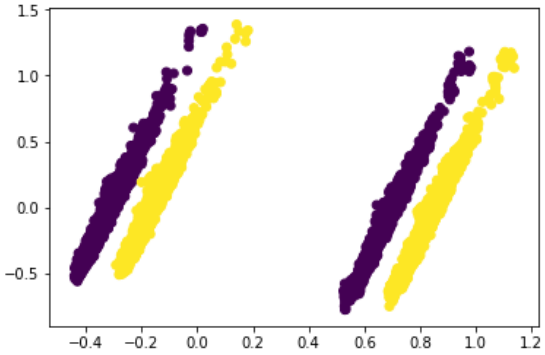

Figure 8: Scatterplot of PCA 2 and PCA 3 using Scaled Data

The results of the three algorithms are shown in Tables 4-6. Cross validation results were preferred to holdout method results as they achieved more generalized relationships in the data and ruled out model overfitting. As expected, from the results of the k-means clustering and principal component analysis,

the scaled data worked best for each classifier. For KNN, the optimal number of neighbors using the loop was optimized at 29.

The neural network was the best performing classifier with a cross validation accuracy of 82.02% which was approximately the same as its test accuracy using the holdout method. The other two classifiers were close in accuracy with KNN at 81.23% and SVM at 78.67%.

	K-Nearest Neighbors
Accuracy	0.810505
Mean Squared Error	0.189495
Precision	0.789352
Recall	0.810505
Cross Validation Accuracy	0.812368

Table 4: KNN Results

	Support Vector Machines
Train Accuracy	0.780647
Test Accuracy	0.785152
Precision	0.789863
Recall	0.785152
Cross Validation Accuracy	0.786667

Table 5: SVM Results

	Neural Network
Train Accuracy	0.820746
Test Accuracy	0.820202
Precision	0.802849
Recall	0.820202
Cross Validation Accuracy	0.820203

Table 6: Neural Network Results

The precision is intuitively the ability of the classifier not to label as positive a sample that is negative while the recall is intuitively the ability of the classifier to find all the positive samples. These performance scores also showed support for the neural network as the best classifier with 80.28% and 82.02% respectively.

The neural network solver "lbfgs" outperformed "adam" even though the latter was documented as having great performance on large datasets. "lbfgs" stands for limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm. This solver may have outperformed the other classifiers due to its use of a limited amount of computer memory and that it is well-suited to optimize many variables as was necessary for this data. Therefore, as seen in feature selection where certain attributes assist in explaining the target better than others, or are more important than others, this algorithm sought to focus on these in correctly classifying data. Additionally, the logistic activation function only enhanced the solver's capability.

With regards to real-world application of classifying credit cardholders or other forms of credit issue, it is a difficult task to use many attributes from an individual to classify him/her as creditworthy. However, this study was able to provide a useful model with an accuracy of approximately 82%. However, instead of a target label of "default" and "non-default", improvements can be made into scoring an individual on a spectrum rather a binary classification since there is significant room for error.

***References*:**

[1] Kevin Fallon McCarthy. 2017. Credit Card Default Rate are Highest in Five Years. (October 2017). Retrieved April 1, 2018 from https://www.mccarthylawyer.com/2015/03/20/credit-card-defaults-highest-in-five-yearsebdefaults-highest-in-five-years/

[2] W. Dong, C. Moses, and K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures," *Proc. 20th Int. Conf. World wide web - WWW '11*, p. 577, 2011.

[3] G. Li, "Keyword-based k -Nearest Neighbor Search in Spatial Databases," pp. 2144–2148, 2012.

[4] K. Zheng, P. C. Fung, and X. Zhou, "K-nearest neighbor search for fuzzy objects," *Proc. 2010 Int. Conf. Manag. data - SIGMOD '10*, p. 699, 2010.

[5] Yeh, I. C., & Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.

[6] L.-H. Li, C.-T. Lin, and S.-F. Chen, "Micro-lending Default Awareness Using Artificial Neural Network," *Proc. 2017 2nd Int. Conf. Multimed. Syst. Signal Process. - ICMSSP 2017*, pp. 56–60, 2017.

[7] S. R. Islam, W. Eberle, and S. K. Ghafoor, "Mining Bad Credit Card Accounts from OLAP and OLTP," *Proc. Int. Conf. Comput. Data Anal. - ICCDA '17*, pp. 129–137, 2017.

[8] H. Yu, I. Ko, Y. Kim, S. Hwang, and W.-S. Han, "Exact indexing for support vector machines," *Proc. ACM SIGMOD Int. Conf. Manag. Data*, pp. 709–720, 2011.

[9] Y. Xiao, F. Deng, B. Liu, S. Liu, D. Luo, and G. Liang, "A learning process using svms for multi-agents decision classification," *Proc. - 2008 IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. - Work. WI-IAT Work. 2008*, no. 4, pp. 583–586, 2008.

[10] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, 2002.

[11] Z. Yulia, O. Krasotkina, and V. Mottl, "Sparse logistic regression with supervised selectivity for predictors selection in credit scoring," *Proc. Seventh Symp. Inf. Commun. Technol. - SoICT '16*, pp. 167–172, 20