The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients

By Cheng Yeh, Che-hui Lien (2009)

Expert Systems with Applications

Introduction

In the past decade, the credit card issuers in Taiwan faced the cash and credit card debt crisis. To increase the market-share the banks issued cards to unqualified candidates or clients. Therefore, most clients of the bank irrespective of their repayment ability, overused credit card and accumulated heavy credit or debts. In a developed financial system, risk prediction and management are important aspect. The real motivation behind risk forecast is to utilize financial information, such as financial statement, customer transaction, age, sex, housing saving account, duration of credit, purpose of debt, housing and repayment records, etc., to predict individual customers' credit risk and to reduce or manage the damage and uncertainty.

This experimental research is undergone to evaluate different data mining methods on default installments made by clients of a bank in Taiwan. This research was based on analyses and comparison of predictive performance of default payments towards the banking institution using six data mining methods. In the paper, it is demonstrated that the forecasting model created by ANN has the highest coefficient of determination; its regression intercept is near to zero, and regression coefficient to one. The data mining method and their application are used for credit scoring of clients. Data mining methodologies are an amazing tool for decision support system and play a key role in market segmentation, customer segmentation, fraud detection, credit and behavior scoring benchmarking.

Methodology:

Dataset

The data set used in this research was taken from UCI ML data repository. The data is from an important bank (a cash and credit card issuer) in Taiwan and the targets were credit card holders of the bank. Of the entire dataset there were 30% of default payments to the debit card. The dataset consists of total 522 client records. The research was employed on a binary variable risk (Good = 1, Bad = 0), as the target variable. The attributes under consideration are "Age", "Sex", "Job", "Housing", "Saving.accounts", "Checking.account", "Credit.amount", "Duration", "Purpose", "Risk".

Sorting smoothing method

The researcher used the novel approach, called Sorting Smoothing Method (SSM), to estimate the real probability of default, was proposed in this study. They used it to estimate the real probability of default through the formula:

$$P_i = \frac{Y_{i-n} + Y_{i-n+1} + \cdots + Y_{i-1} + Y_i + Y_{i+1} + \cdots + Y_{i+n-1} + Y_{i+n}}{2n+1}$$

where Pi = estimated real probability of default in the ith order of validation data; Yi = default risk in the ith order of validation data; n=numbers of data for smoothing.

Data mining models and evaluation

The researchers used the six data mining techniques used in this paper, which are discriminant analysis, logistic regression, Bayes classifier, nearest neighbor, artificial neural networks, and classification trees to perform evaluation of prediction on the credit card fraud transaction dataset. They performed evaluation of model by using statistical analysis using Regression R2, Regression intercept, Regression coefficient and area ratio from lift chart for the evaluation of models.

Results and conclusion from the research

According to the experiment taken up the researcher, K-nearest neighbor classifiers

and classification trees have the lowest error rate as compared to lift charts of other data mining models. The following is the observation for classification accuracy:

Table 1
Classification accuracy

Method	Error rate		Area ratio	
	Training	Validation	Training	Validation
K-nearest neighbor	0.18	0.16	0.68	0.45
Logistic regression	0.20	0.18	0.41	0.44
Discriminant analysis	0.29	0.26	0.40	0.43
Naïve Bayesian	0.21	0.21	0.47	0.53
Neural networks	0.19	0.17	0.55	0.54
Classification trees	0.18	0.17	0.48	0.536

From the area ratio, it has been found out K-nearest neighbor classifiers, has the highest area ratio=0.68 with the training set. But artificial neural networks perform the best result with the highest area ratio of 0.54 with the validation set and low error rate of 0.17. As the validation data is the effective data set used to measure the generalization classification accuracy of evaluation of data mining models, so researchers acknowledged artificial neural networks to be the best performing model in this credit card scoring. The following is the observation of regression analysis on different data mining model by researchers:

Table 2
Summary of linear regression between real probability and predictive probability of default

Method	Regression Coefficient	Regression Intercept	Regression R^2
K-nearest neighbor	0.770	0.0522	0.876
Logistic regression	1.233	−0.0523	0.794
Discriminant Analysis	0.837	−0.1530	0.659
Naïve Bayesian	0.502	0.0901	0.899
Neural networks	0.998	0.0145	0.965
Classification trees	1.111	−0.0276	0.278

The result of R2 showed that the predictive ability of artificial neural networks (R2 = 0.9647) has the highest explanatory ability of the variance of target variable with respect to variance of dependent variables.

Evaluation from experimentation

I have undergone evaluation of the six data mining models i.e, discriminant analysis, logistic regression, Bayes classifier, nearest neighbor, artificial neural networks, and classification trees on prediction on the credit card fraud transaction dataset. The evaluation metric used by me are R2 for the fit of the model, Accuracy for the strength of the prediction, and Recall accuracy which defines the proportion of relevant results of the number of samples with respect to actual value of the samples. Accuracy provides the strength of model to predict correct instances with respect to total number of instances. The following are the observation from my experiments:

	Regression R2	*Accuracy*	*Recall*
Logistic Regression	0.5264	0.7164	0.7399
Classification trees	0.29854	0.7567	0.7626
K- nearest neighbor	0.7490	0.7719	0.7933
Discriminate Analysis	0.6399	0.6418	0.6196
Naïve Bayesian	0.5599	0.8199	0.8769
Neural Network	0.9184	0.8747	0.9177

As from our experiment basis, Neural network performs the best with R2=0.9184, which is near to 1, that means the model best fits the data, Accuracy=0.8747 which is better than all other model and its recall value = 0.9177 which is really good in terms of predicting the fraud transactions. Moreover, Naïve Bayesian also performs relatively good and it's the second-best performing model for the dataset, with R2=0.8599, accuracy=0.8199, Recall= 0.8769. The result produced by our data mining model are quite like what proposed by the researchers in the paper. We can say that, the data used by the researcher have undergone through other data preparation techniques, so the results are different. But its significant in the research paper and my observations that neural network performs the best in credit data scoring.

Refrences

[1] Cheng Yeh, Che-hui Lien (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications,36,* 2473–2480. doi:10.1016/j.eswa.2007.12.020

APPENDIX

````
---
title: "M task 2"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r}
df<-read.csv("german_credit_data.csv")
library(arm)
```


```{r}
df1<-na.omit(df)
#df1<-as.numeric(df1)
df1<-lapply(df1,as.numeric)
df1<-as.data.frame(df1)
#df1$Risk<- as.factor(df1$Risk)

#dummyRisk
df1$dummyRisk<-as.numeric(df1$Risk==2)
class(df1$dummyRisk)
names(df1$dummyRisk)
df1$Risk<-NULL

temp<- df1
```



```{r}
lr<-glm(df1$dummyRisk~.,family = binomial(link = "logit"), data = df1)
summary(lr)
lr
# Use the model to predict the evaluation.
df1$prediction <- (predict(lr, newdata=df1))
#df1$prediction<-(df1$prediction>0)
s<-sum(range(df1$prediction))
df1$prediction<-as.numeric(df1$prediction>s)

#RegressionR2
library(rsq)
rsq(lr)

#Intercept
coef(lr)["(Intercept)"]
````
````

```
#Regression Coeffient
coef(lr)["X"]


# Calculate the overall accuracy.
df1$Correct <- df1$prediction == df1$dummyRisk
print(paste("% of predicted classifications correct", mean(df1$Correct)))

```


```{r}
df2<- temp
library(caret)
library(ISLR)
library(DMwR)

df2$dummyRisk<-as.numeric(df2$dummyRisk)
Knn<-kNN(df2$dummyRisk~.,train = df2,test = temp, cl= df2$dummyRisk, k=2)

class(df2$dummyRisk)
# Use the model to predict the evaluation.
predict(Knn, newdata=df2)
#df1$prediction<-(df1$prediction>0)
s<-sum(range(df1$prediction))
df1$prediction<-as.numeric(df1$prediction>s)

#RegressionR2
library(rsq)
rsq(lr)

#Intercept
coef(lr)["(Intercept)"]

#Regression Coeffient
coef(lr)["X"]

confusionMatrix(as.factor(df1$dummyRisk), as.factor(df1$prediction))



```


```{r}
df3<-temp
library(MASS)
ld <- lda(df3$dummyRisk ~ ., data = df3)
prediction<- predict(ld, newdata = df3)
df3<-cbind(df3, prediction)
```

```
class(df3$dummyRisk)

df3$dummyRisk<-as.factor(df3$dummyRisk)
class(df3$class)

df3$class<- as.numeric(df3$class)

df3$class<- as.numeric(df3$class > 1)

df3$class<- as.factor(df3$class)

confusionMatrix(df3$dummyRisk, df3$class)

levels(df3$class)
levels(df3$dummyRisk)

#RegressionR2
library(rsq)
rsq(ld)

#Intercept
coef(ld)["(Intercept)"]

#Regression Coeffient
coef(ld)["X"]


confusionMatrix(as.factor(df3$dummyRisk), as.factor(df3$predictiom))

```

```{r}
library(e1071)
library(klaR)
df4<-temp
nb<-naiveBayes(df4$dummyRisk ~ ., data = df4)
#prediction <-
predict(nb, newdata=df4[,-1])
df4<-cbind(df4,prediction)

#RegressionR2
library(rsq)
rsq(nb)

#Intercept
coef(nb)["(Intercept)"]

#Regression Coeffient
coef(nb)["X"]

confusionMatrix(as.factor(df4$dummyRisk), as.factor(df4$prediction))
```

```

```{r}
library(rpart)
df5<-temp
df5$dummyRisk<-as.numeric(df5$dummyRisk)
ct<-rpart(df5$dummyRisk ~ ., data = df5, method = "class")
prediction<-predict(ct, newdata=df5)
prediction<-as.data.frame(prediction)
prediction$pred<-prediction$`0`<prediction$`1`
df5<-cbind(df5,prediction$pred)
df5$`prediction$pred`<-as.numeric(df5$`prediction$pred`)

#RegressionR2
library(rsq)
rsq(ct)

#Intercept
coef(ct)["(Intercept)"]

#Regression Coeffient
coef(ct)["X"]



confusionMatrix(as.factor(df5$dummyRisk), as.factor(df5$pred))
levels(df5$dummyRisk)
class(df5$`prediction$pred`)


```

```{r}
library(neuralnet)
df6<-temp[,-1]
df6$dummyRisk<-as.factor(df6$dummyRisk)
nn <- neuralnet(df6$dummyRisk~. , data = df6)

predict(nn, newdata=df6)
prediction<-as.data.frame(prediction)
prediction$pred<-prediction$`0`<prediction$`1`
df5<-cbind(df5,prediction$pred)
df5$`prediction$pred`<-as.numeric(df5$`prediction$pred`)

confusionMatrix(as.factor(df6$dummyRisk), as.factor(df6$pred))
levels(df5$dummyRisk)
class(df5$`prediction$pred`)

#RegressionR2
library(rsq)
```
```

rsq(nn)

#Intercept
coef(nn)["(Intercept)"]

#Regression Coeffient
coef(nn)["X"]

```
```