IIT Kharagpur

Report for Project 4

Group 1

Customer Churn Prediction

Authors:
Rameshwar Bhaskaran
Aniket Choudhary
Shubham Sharma
Asket Agarwal
Apurv Kumar

1 Introduction

Customer Relationship Management (CRM) is a comprehensive strategy for building, managing and strengthening loyal and long-lasting customer relationships. It is broadly acknowledged and extensively applied to different fields, for example, telecommunications, banking and insurance, retail market, etc.

Thus, tools to develop and apply customer retention models (churn models) are required and are essential Business Analytics applications. In the dynamic market environment, churning could be the result of low-level customer satisfaction, aggressive competitive strategies, new products, regulations, etc. Churn models aim to identify early churn signals and recognize customers with an increased likelihood to leave voluntarily.

So, given a data set of 5000 costumers we need to prepare a model or its variation that we have learned in class to learn and predict the churn rate of a new costumer based on his/her previous records.

The features for data set are as follows :

Variable	Name	Type
Account length	(number of months active user)	Num
Total eve charge	(total charge of evening calls)	Num
area code Num	(area code of customer)	Num
total night minutes	(total minutes of night calls)	Num
international plan	(local/international call)	Binary (Yes/No)
total night calls	(total number of night calls)	Num
voice mail plan	(voice mail or normal)	Binary (Yes/No)
total night charge	(total charge of night calls)	Num
number vmail messages	(number of voice-mail messages)	Num
total intl minutes	(total minutes of international calls)	Num
total day minutes	(total minutes of day calls)	Num
total intl calls	(total number of international calls)	Num
total day calls	(total number of day calls)	Num
total intl charge	(total charge of international calls)	Num
total day charge	(total charge of day calls)	Num
customer service calls	(number of calls to customer service)	Num
total eve minutes	(total minutes of evening calls)	Num
total eve calls	(total number of evening calls)	Num
churn	(customer churn - target variable)	Binary (Yes/No)

2 Objective

The following are the objectives that is to be met by this project using the data set and the paper provide :
1. Review the predictor variables and guess what their roles in customer churn. You are free to create new derived variables from these predictors. [Hint:You have solved similar problems in Project 3]
2. Divide the data into training and test set.Make sure relative proportions of true and false in the target variable are maintained in training and test set. [Hint: Use stratified random sampling : You can use data partition function of R Caret package.]
3. Using training data set, develop classification models using at least 3 classification techniques (1) Naïve Bayes' Classifier, 2) Any one decision tree classifier and 3) SVM classifier. You can get the classifier in R Caret package.]
4. Construct confusion matrices using test data set for each model. Compute a) Accuracy, b) Precision and c) Recall for each model.
5. Try to improve classification accuracy by choosing right set of predictor variables and model parameters and choose the best model. [Hint: Follow ROC curve, or any other if you think suitable.]

3 Preprocessing the data

Data is provided as a csv file(comma-seperated values). Preprocessing of the data involves following steps:

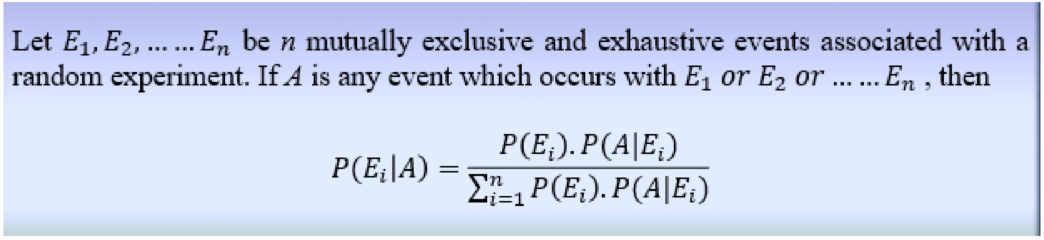
- Categorical variables like international plan, voice mail plan, churn were converted to dummy variables.
- ID and phone number columns were removed.
- area code, state, number of night calls, number of evening calls and account length columns were also removed to get better results based upon the random forest importance matrix and correlation matrix.

4 Algorithm Used

The partition used was 2:1. For training, different parameters were used for different models based upon the optimum output like for random forest, number of trees chosen was 6000.

4.1 Naive-Bayes Classifier

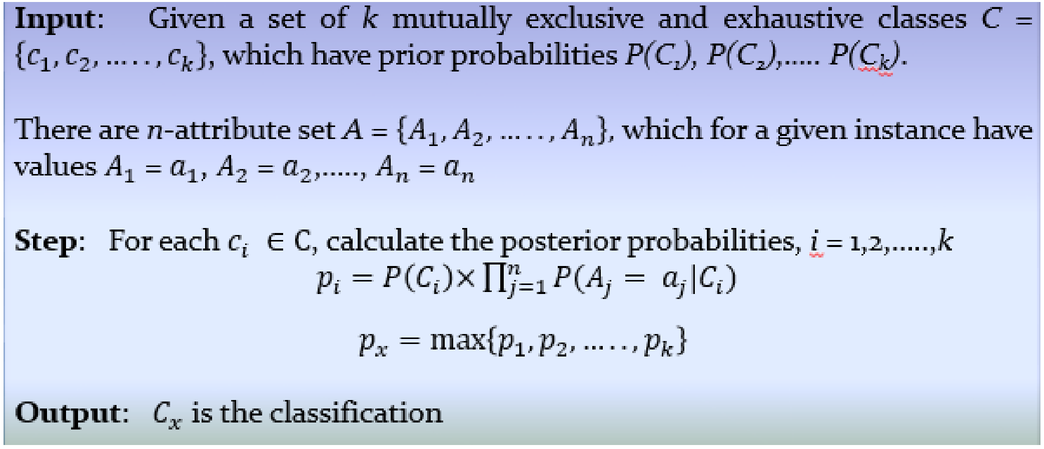
The Bayesian classifier is an approach for modeling probabilistic relationships between the attribute set and the class variable. Bayesian classifier use Bayes' Theorem of Probability for classification. Bayes' Theorem :



Let $E_1, E_2, \ldots\ldots E_n$ be n mutually exclusive and exhaustive events associated with a random experiment. If A is any event which occurs with E_1 *or* E_2 *or* $\ldots\ldots E_n$, then

$$P(E_i|A) = \frac{P(E_i).\,P(A|E_i)}{\sum_{i=1}^{n} P(E_i).\,P(A|E_i)}$$

Algorithm for Naive Bayes Classification :



Input: Given a set of k mutually exclusive and exhaustive classes $C = \{c_1, c_2, \ldots\ldots, c_k\}$, which have prior probabilities $P(C_1)$, $P(C_2)$,..... $P(C_k)$.

There are n-attribute set $A = \{A_1, A_2, \ldots\ldots, A_n\}$, which for a given instance have values $A_1 = a_1$, $A_2 = a_2$,....., $A_n = a_n$

Step: For each $c_i \in$ C, calculate the posterior probabilities, i = 1,2,.....,k

$$p_i = P(C_i) \times \prod_{j=1}^{n} P(A_j = a_j|C_i)$$

$$p_x = \max\{p_1, p_2, \ldots\ldots, p_k\}$$

Output: C_x is the classification

4.2 Decision Tree Classifier

Decision tree builds classification or regression models in the form of a tree structure. The decision tree induction algorithm works by recursively selecting the best attribute to split the data and expanding the leaf nodes of the tree until the stopping criterion is met. The choice of best split test condition is determined by comparing the impurity of child nodes and also depends on which impurity measurement is used.

4.3 Support Vector Machine Classifier

SVM Classifier searches for optimal hyper plane separating the tuples of one class from another. A good separation is achieved by the hyper plane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

4.4 Random Forest Classifier

Random Forests grows many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree ”votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

Each tree is grown as follows:

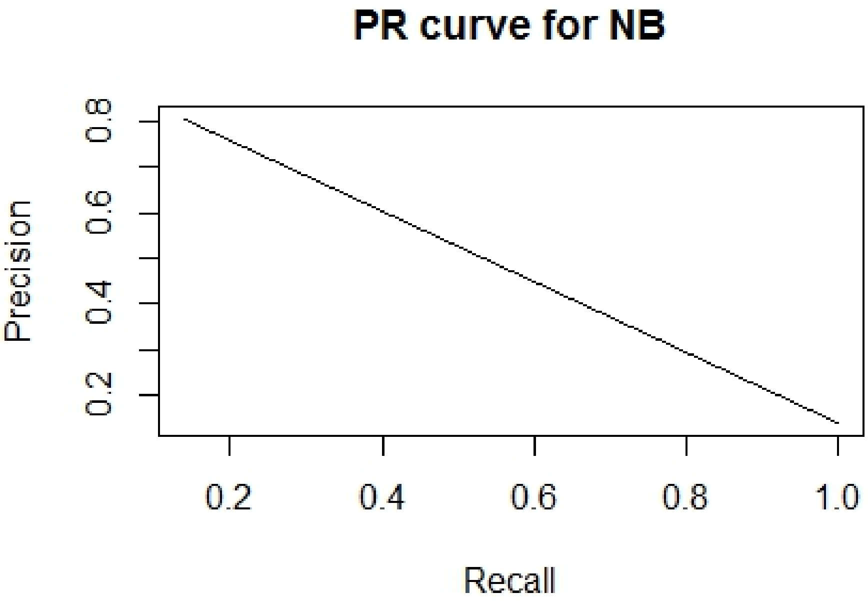
If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree. If there are M input variables, a number m¡¡M is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing. Each tree is grown to the largest extent possible. There is no pruning. In the original paper on random forests, it was shown that the forest error rate depends on two things:

The correlation between any two trees in the forest. Increasing the correlation increases the forest error rate. The strength of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate. Reducing m reduces both the correlation and the strength. Increasing it increases both. Somewhere in between is an ”optimal” range of m - usually quite wide. Using the error rate (see below) a value of m in the range can quickly be found. This is the only adjustable parameter to which random forests is somewhat sensitive.

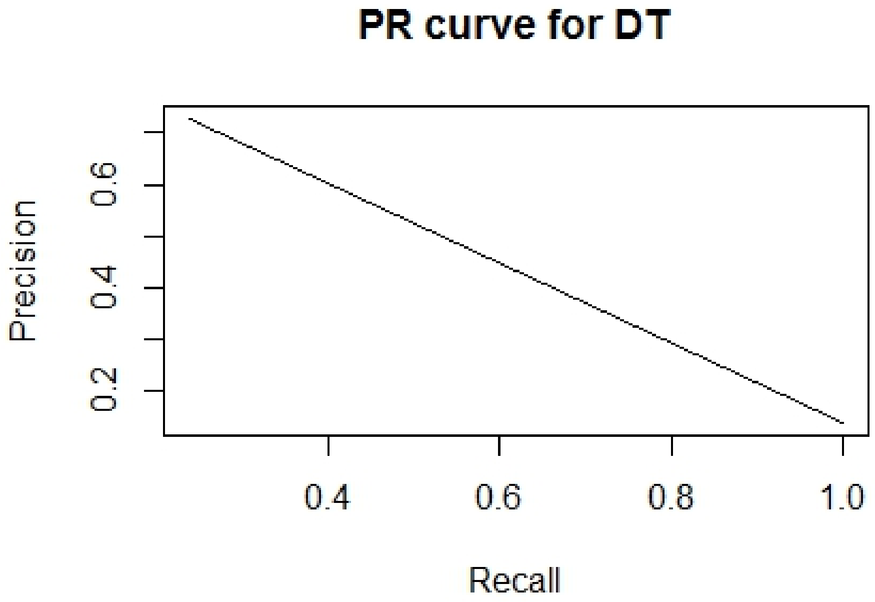
5 Performance

5.1 Precision-Recall Curve

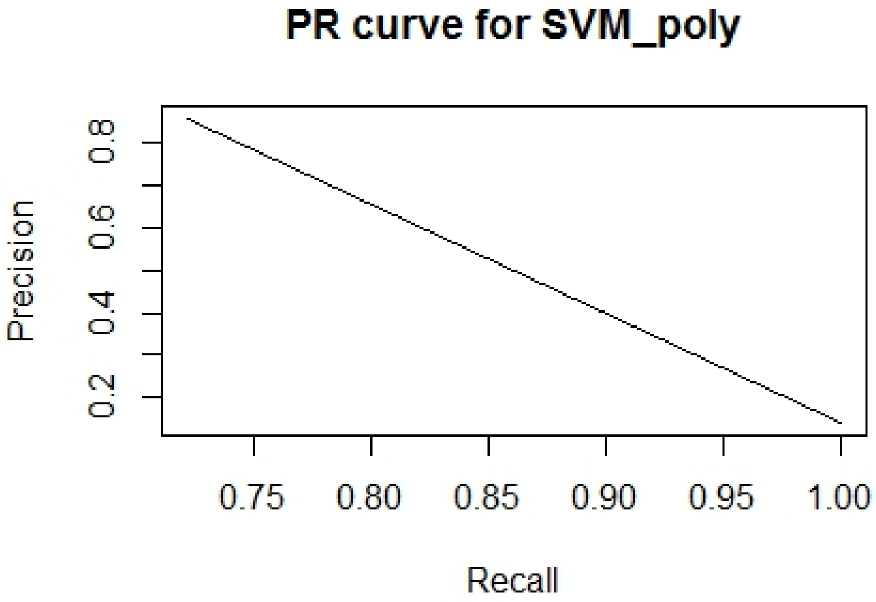
- **Naive-Bayes Classifier**



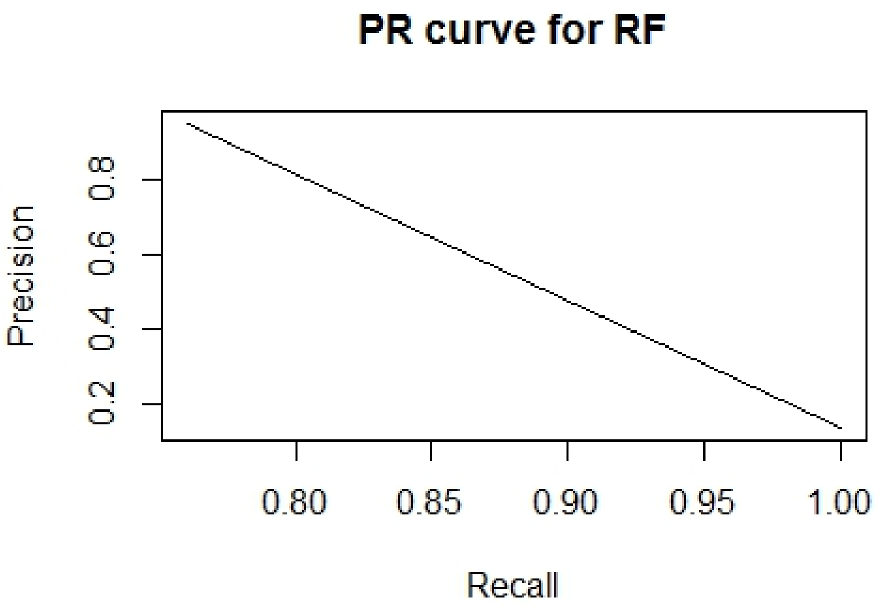
- **Decision Tress Classifier**



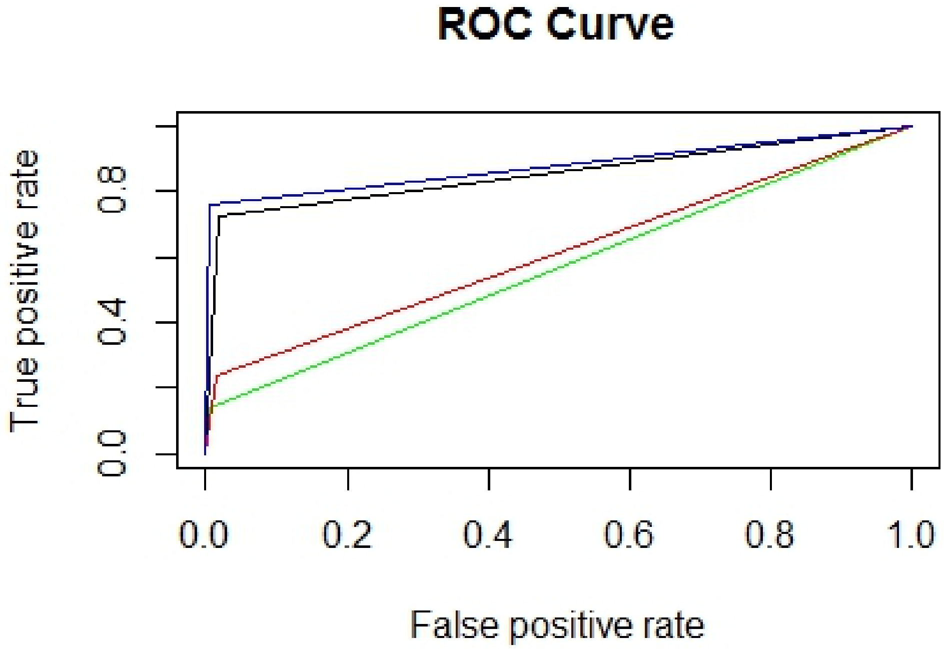
- **Support Vector Machines Classifier**



- **Random Forest Classifier**


PR curve for RF
Precision
0.2
0.4
0.6
0.8
0.80
0.85
0.90
0.95
1.00
Recall

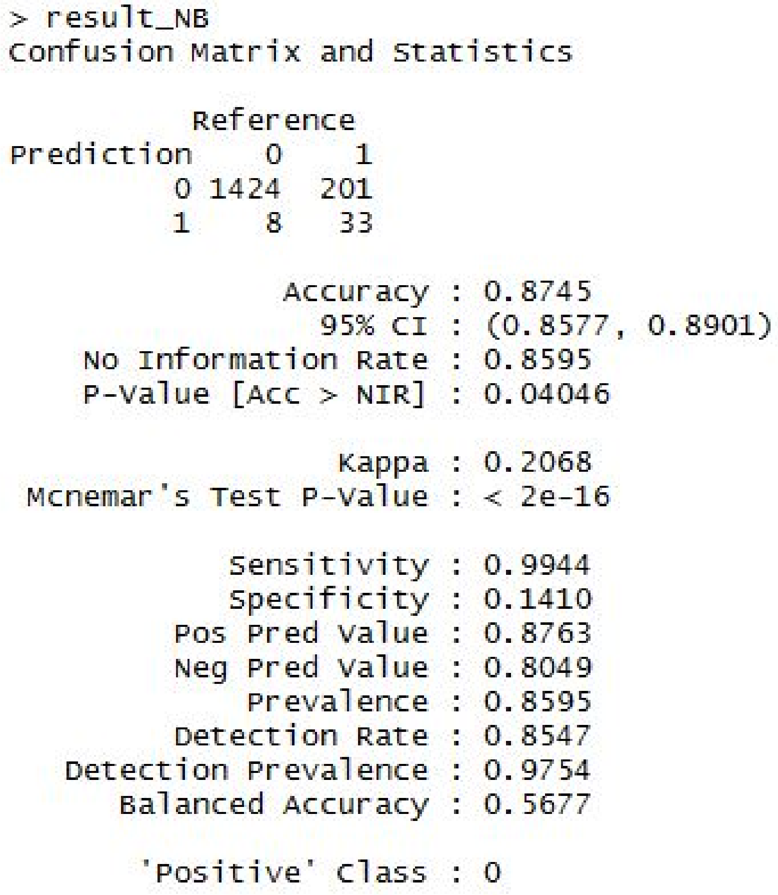
5.2 ROC Curve



Green Line: Naive Bayes
Red Line: Decision Trees
Black Line: SVM
Blue Line: Random Forest

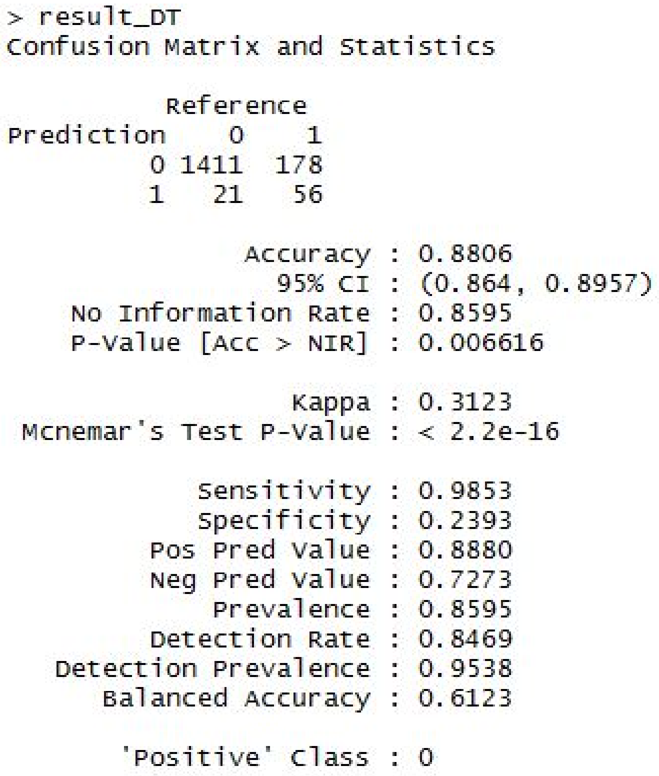
5.3 Confusion Matrix

Naive-Bayes Classifier



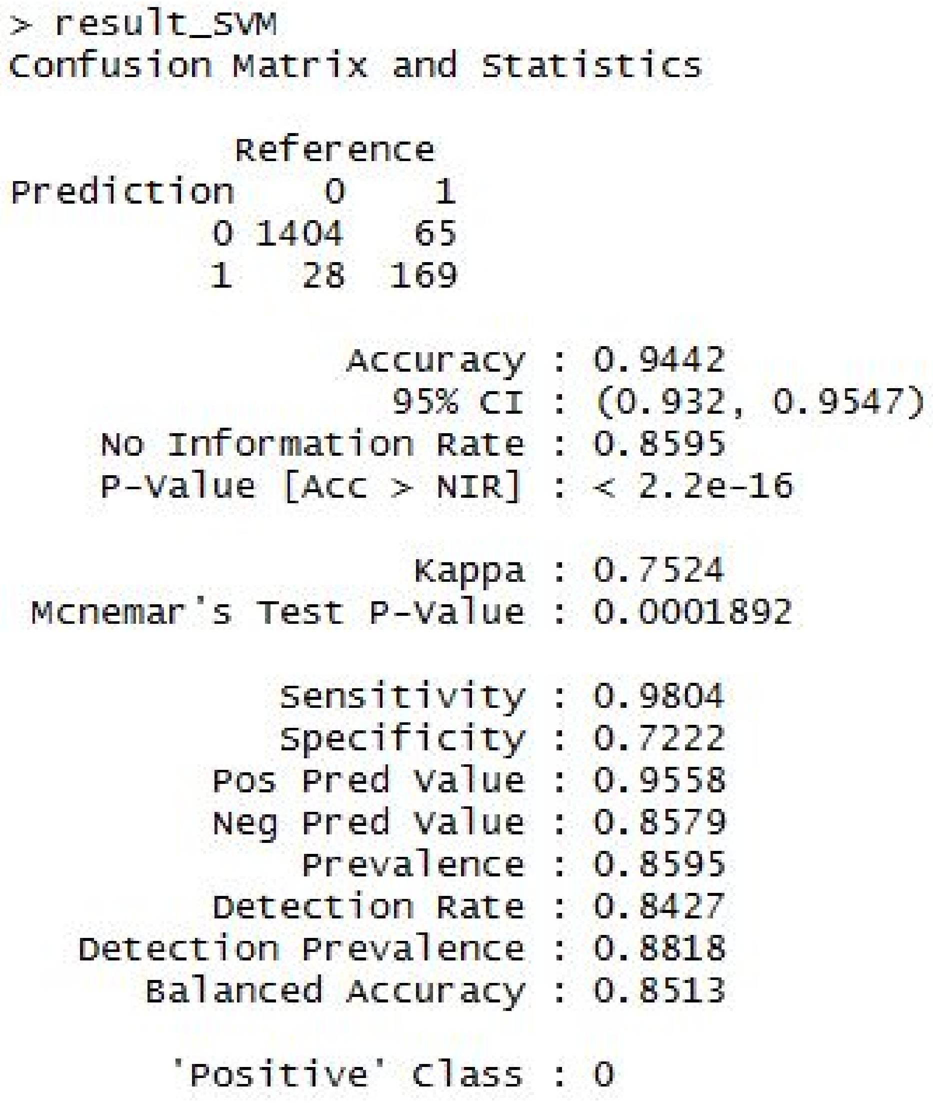
```
> result_NB
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1424  201
         1    8   33

               Accuracy : 0.8745
                 95% CI : (0.8577, 0.8901)
    No Information Rate : 0.8595
    P-Value [Acc > NIR] : 0.04046

                  Kappa : 0.2068
 Mcnemar's Test P-Value : < 2e-16

            Sensitivity : 0.9944
            Specificity : 0.1410
         Pos Pred Value : 0.8763
         Neg Pred Value : 0.8049
             Prevalence : 0.8595
         Detection Rate : 0.8547
   Detection Prevalence : 0.9754
      Balanced Accuracy : 0.5677

       'Positive' Class : 0
```

Decision Tree Classifier



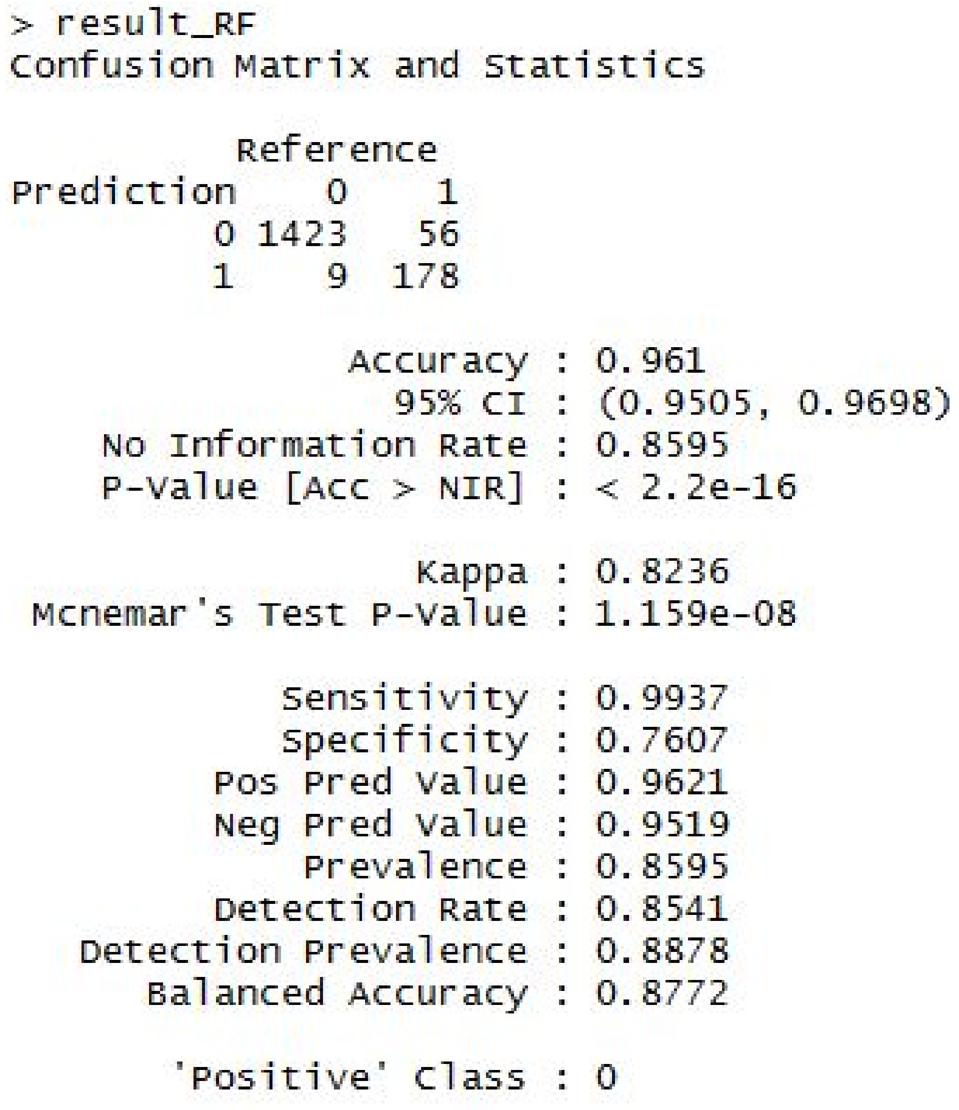
```
> result_DT
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1411  178
         1   21   56

               Accuracy : 0.8806
                 95% CI : (0.864, 0.8957)
    No Information Rate : 0.8595
    P-Value [Acc > NIR] : 0.006616

                  Kappa : 0.3123
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9853
            Specificity : 0.2393
         Pos Pred Value : 0.8880
         Neg Pred Value : 0.7273
             Prevalence : 0.8595
         Detection Rate : 0.8469
   Detection Prevalence : 0.9538
      Balanced Accuracy : 0.6123

       'Positive' Class : 0
```

Support Vector Machine Classifier



```
> result_SVM
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1404   65
         1   28  169

               Accuracy : 0.9442
                 95% CI : (0.932, 0.9547)
    No Information Rate : 0.8595
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7524
 Mcnemar's Test P-Value : 0.0001892

            Sensitivity : 0.9804
            Specificity : 0.7222
         Pos Pred Value : 0.9558
         Neg Pred Value : 0.8579
             Prevalence : 0.8595
         Detection Rate : 0.8427
   Detection Prevalence : 0.8818
      Balanced Accuracy : 0.8513

       'Positive' Class : 0
```

Random Forest Classifier



```
> result_RF
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1423   56
         1    9  178
                                          
               Accuracy : 0.961           
                 95% CI : (0.9505, 0.9698)
    No Information Rate : 0.8595          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.8236          
 Mcnemar's Test P-Value : 1.159e-08       
                                          
            Sensitivity : 0.9937          
            Specificity : 0.7607          
         Pos Pred Value : 0.9621          
         Neg Pred Value : 0.9519          
             Prevalence : 0.8595          
         Detection Rate : 0.8541          
   Detection Prevalence : 0.8878          
      Balanced Accuracy : 0.8772          
                                          
       'Positive' Class : 0               
```

6 References

1. CCP Paper

2. Class Notes

3. Random Forest Classifier