Nikhil Paonikar and Connor Pigott

Dr. Olfa Nasraoui

CECS 621-50 Web Mining

22 July 2018

PREDICTING CREDIT CARD DEFAULT USING CLASSIFICATION MODELS

INTRODUCTION

The world functions and flourishes as a result of the global economy. Often, consumers availing financial services find themselves facing financial comedown. This is usually the slow, resultant effect of consistently exacerbating financial decisions. Several red flags can be identified to deduce any possibilities of late payments, fraud, default. Similarly, credit worthiness, raising credit, customer-tier upgradability can be predicted. This can be done by predicting or classifying the reliability of a client given the available attributes taken from a loan application.

1. DATASET

The dataset is derived from default payments in Taiwan.[1] A response variable "**Y**" represents the real probability of default. An independent variable "**X**" represents the predictive probability of default.

Thus, a simple linear regression result is derived as follows:

Y = A + BX;

which shows that the forecasting model produced by artificial neural network has the highest coefficient of determination. Thus, its regression intercept "**A**" is close to zero and regression coefficient "**B** to one.

Excerpt from the dataset information section: *"This research aimed at the case of customers' default payments in Taiwan and compares the predictive accuracy of probability of default among six data mining methods. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients."*

ATTRIBUTE	DESCRIPTION
X1	Amount of the given credit (NT dollar): includes both the individual consumer credit and family (supplementary) credit.
X2	Gender (1 = male; 2 = female)
X3	Education (1 = graduate school; 2 = university; 3 = high school; 4 = others)
X4	Marital status (1 = married; 2 = single; 3 = others)
X5	Age (year)
X6 - X11	History of past payment
X12-X17	Amount of bill statement (NT dollar)
X18-X23	Amount of previous payment (NT dollar)

[1] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#

2. DATA MINING GOAL

Classification: Our data mining goal is to classify a potential customers ability to make consistent credit card payments.

The data set used represents the backgrounds of 30,000 individuals and their propensity to repay a given loan. Each of the 30,000 records contains the 23 following exploratory attributes: ID, Amount of credit afforded in Taiwanese dollars, gender, educations level, marital status, age in years, and the payment timeliness, payment amount, and bill statements for the respective individuals over the course of five months. A 24th, binary, class variable indicates whether or not the member defaulted on the loan. Of the 30,000 data samples, 6,636 of the records were indicative of payment default.

The goal of this analysis is to develop an accurate model for classifying the reliability of a client given the available attributes taken from a loan application. To identify the best model to meet this objective, a diverse range of algorithms such as Naïve Bayes, Decision tree and Support Vector Machines shall be used for classification of these records.

Pre-processing: Our first step in preprocessing the data was to remove the ID number, as this figure would have no bearing on our outputs and may impact the accuracy of our model. The next step, as our data was imported in the form of a CSV file, was to categorize the age, education, and marriage exploratory variables, which were imported as numerical. Similarly, our predicted default class was changed to categorical. These changes were made directly within the ARFF file created from the CSV.

3. PERFORMANCE/INTERESTINGNESS MEASURES

Our goal is to conduct an analysis which intends to use compiled data to test various classification models to determine which design best identifies whether a client is expected to default on a credit payment or not. Based on the contents of this data detailing payment default amongst a set of individuals, we expect that our models will classify these records into "**predicted default**" class and a "**predicted payment**" class that will indicate a client's ability to pay off a credit card payment.

1. Predicted default: This measure is indicative of a customer being more likely to default on a potential future payment.

2. Predicted payment: This measure is indicative of a customer being more likely to be consistent with potential future payments.

4. SYSTEM INPUTS & OUTPUTS

Inputs:

The inputs will be the following customer information:

1. Amount of the given credit
2. Gender
3. Education
4. Marital status
5. Age
6. History of past payments
7. Amount of bill statements
8. Amount of previous payment

Output:

The output will classify the customs into either of these categories:

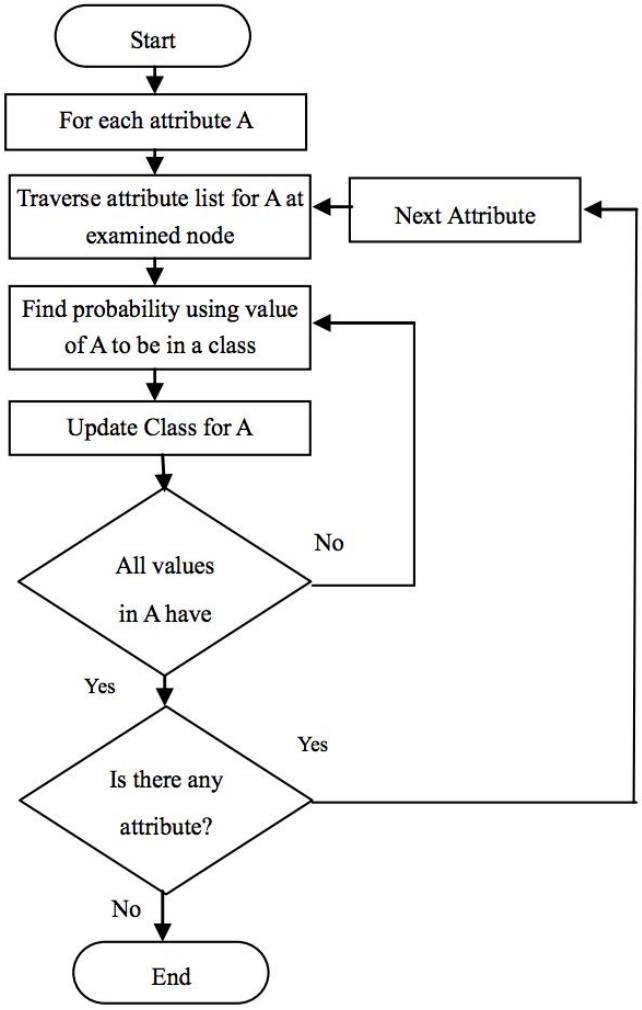
1. Predicted default
2. Predicted payment

5. ALGORITHM

1. Naïve Bayes

The Naïve Bayes is a classification technique based on Bayes' Theorem. It assumes independence among its predictors; i.e., a Naïve Bayes classifier assumes that the presence of a specific feature in a class is unrelated to the presence of other features. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to a resultant probability. This model is easy to build and particularly useful for very large data sets. It has been known to outperform even highly sophisticated classification methods.[2]

Flowchart:

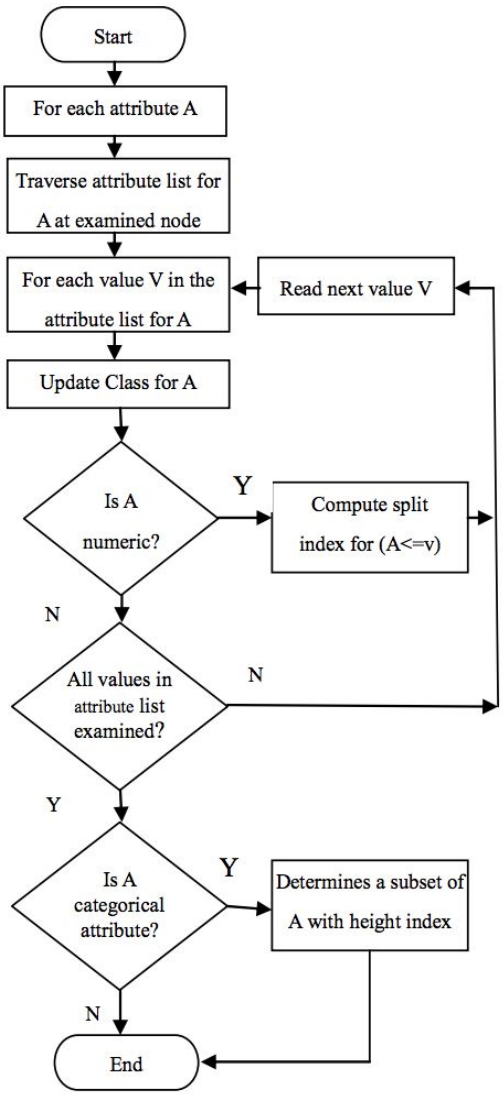


[2] https://file.scirp.org/pdf/JSEA_2013042913162682.pdf

2. Decision tree

A decision tree algorithm creates a tree-like model. It does so by making use of values of a single attribute at a given instance of time. As a primary step, the dataset is sorted according to the accounted attribute's value. Then it looks for regions in the dataset containing only one class and then marking those regions as leaves. For the rest of the regions with multiple classes, another attribute is chosen and the branching process is continued with only the number of instances in those regions until it either produces all leaves or there remains no attribute that can be used to produce one or more leaves in the conflicted regions.[3]

Flowchart:



[3] https://file.scirp.org/pdf/JSEA_2013042913162682.pdf

3. Support Vector Machine

A Support Vector Machine (SVM) is a discriminative classifier. It is formally defined by a separating hyperplane; i.e., given labeled training data (supervised learning), this algorithm produces outputs in the form of an optimal hyperplane which categorizes new examples. In two dimensional space, this hyperplane is a line dividing a plane in two parts where in each class lay in either side.[4]

Pseudocode:

```
candidateSV = { closest pair from opposite classes }
while there are violating points do
   Find a violator
   candidateSV = candidateSV ⋃ violator
   if any α_p < 0 due to addition of c to S then
      candidateSV = candidateSV \ p
      repeat till all such points are pruned
   end if
end while
```

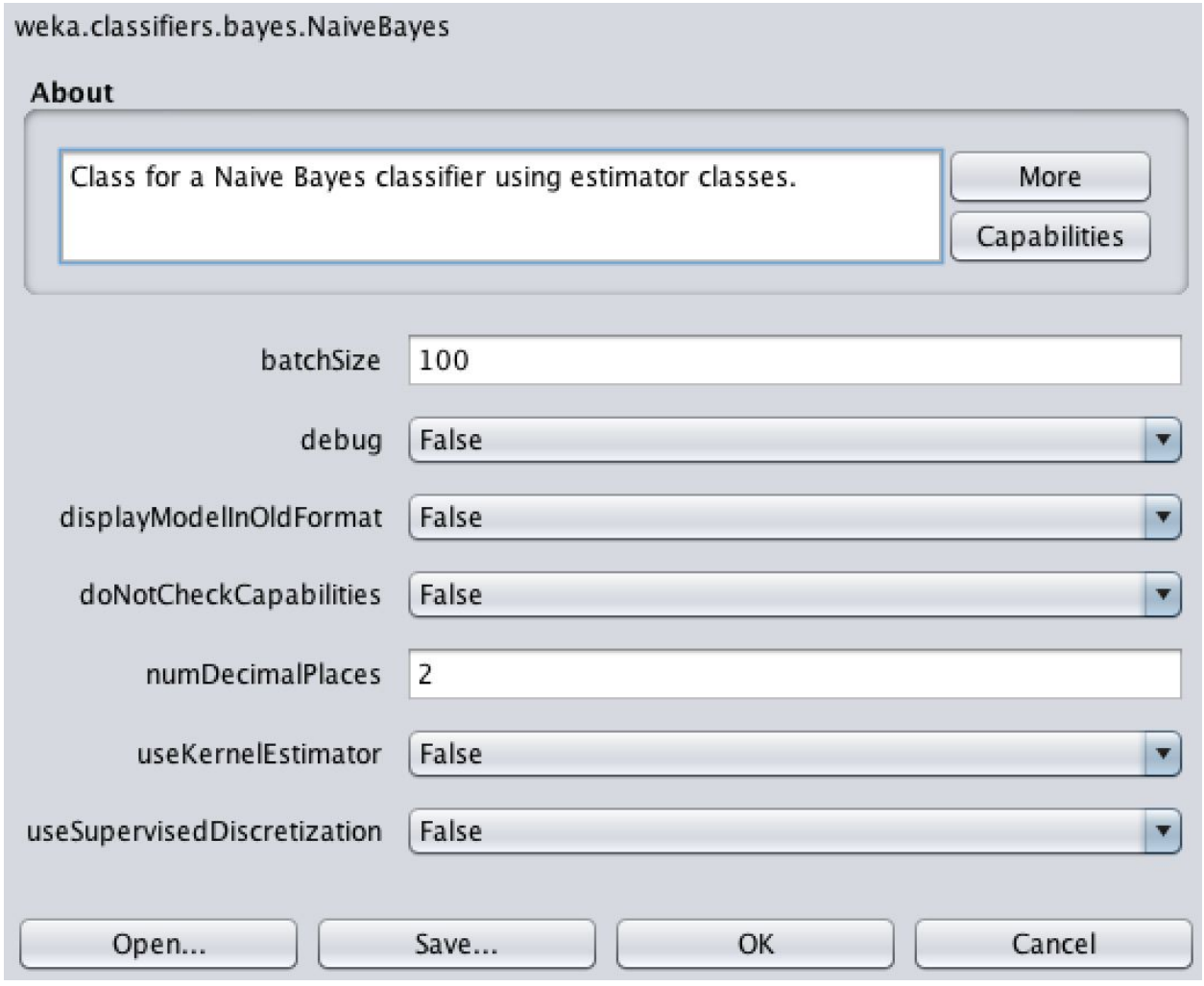
Complexity	Naïve Bayes[5]	Decision tree[6]	SVM[7]
Space	O(pqr), where p is the number of features, q is values for each feature, and r is alternative values for the class.	O(1)	$O(N^2)$
Time	O(Np), where N is the number of training examples and p is the number of features	$O(mn^2)$	$O(N^3)$

[4] https://file.scirp.org/pdf/JSEA_2013042913162682.pdf
[5] http://www.sysnet.ucsd.edu/~cfleizac/cse250b/project1.pdf
[6] https://static.aminer.org/pdf/PDF/000/014/727/a_fast_decision_tree_learning_algorithm.pdf
[7] http://www.cs.cmu.edu/~chunlial/docs/16uaiKernel.pdf

6. IMPLEMENTATION

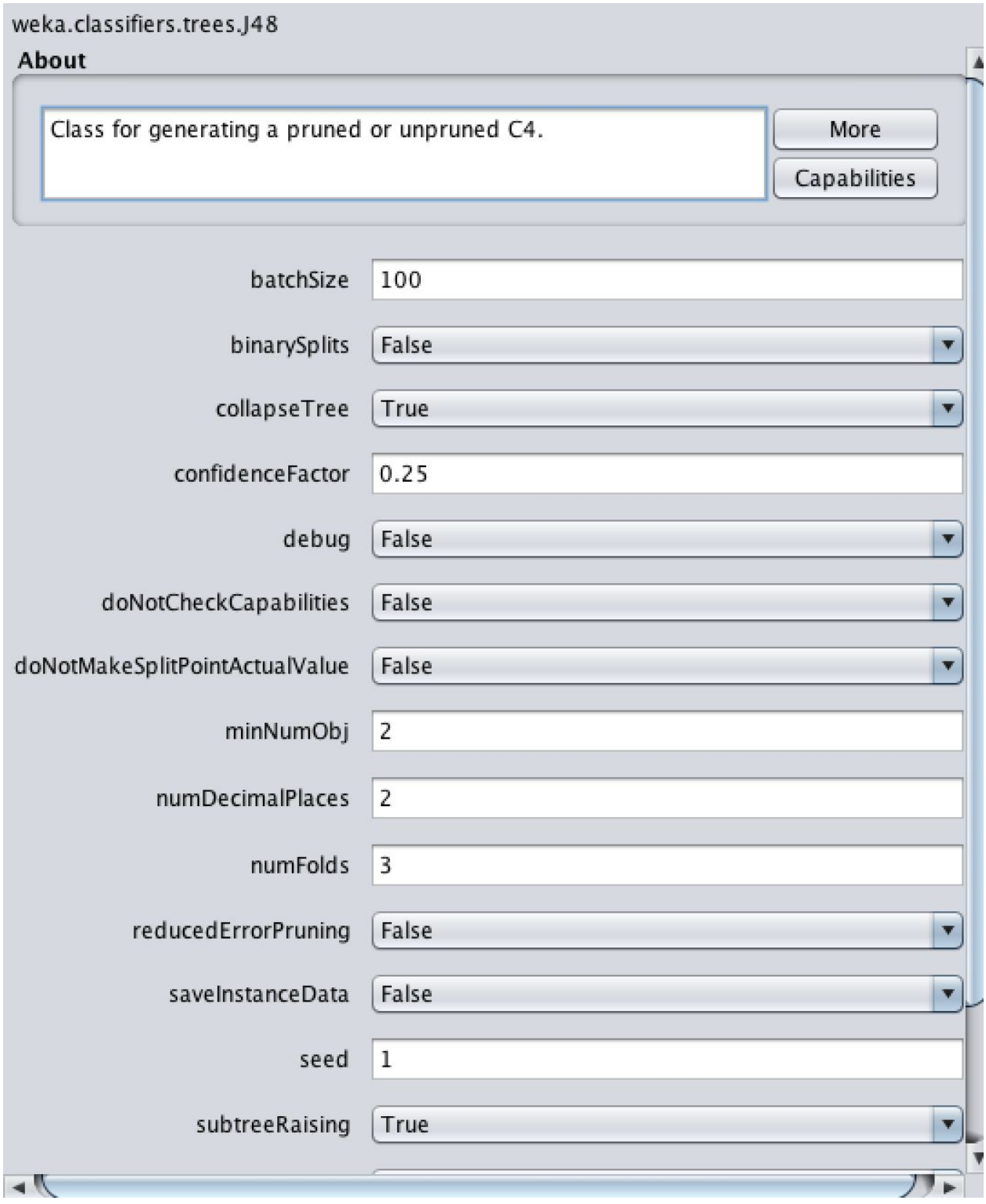
We use Weka to execute the classification techniques. Within the software Weka, a number of other filters were applied. Initially, Weka was used to discretize the remaining data in order to apply models such as Naïve Bayes. The information gain filter was also used to rank the inductiveness of the suite of attributes. The algorithms to be used in the analysis are the Naïve Bayes, Decision Tree (J48 in Weka), and Support Vector Machine (LibSVM in Weka).

1. Naïve Bayes



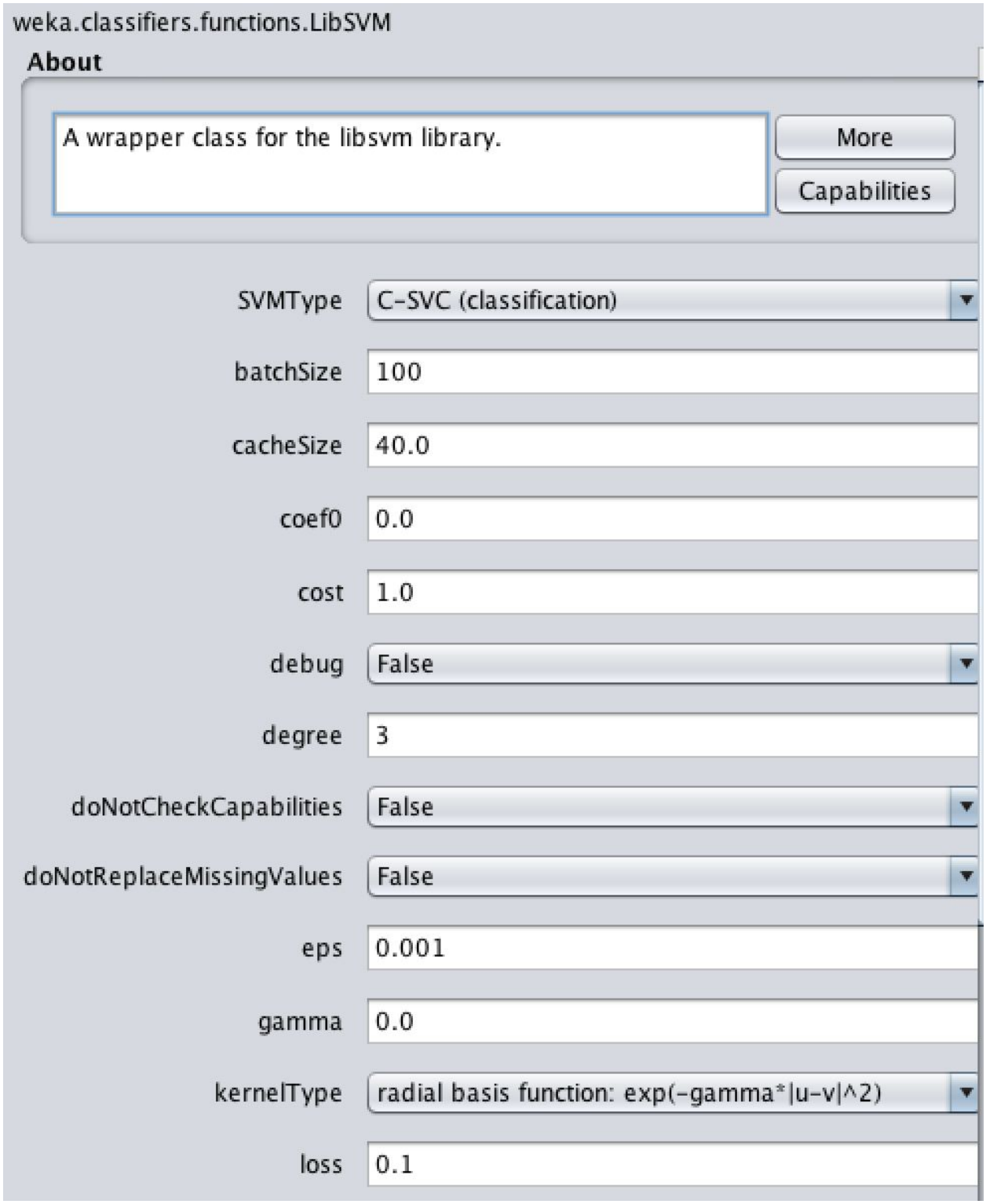
Screenshot showing settings of Naïve Bayes classifier used on the dataset.

2. Decision tree (J48)



Screenshot showing settings of Decision tree classifier used on the dataset.

3. Support Vector Machine (LibSVM)



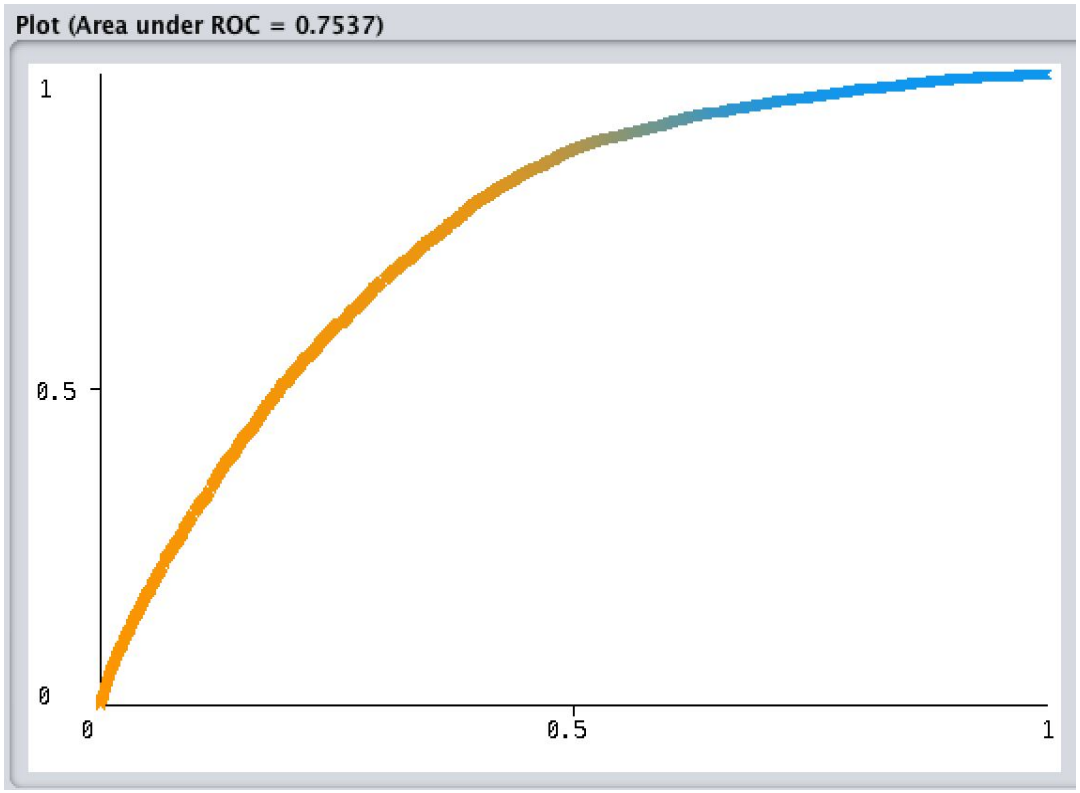
Screenshot showing settings of Support Vector Machines classifier used on the dataset.

7. TESTING

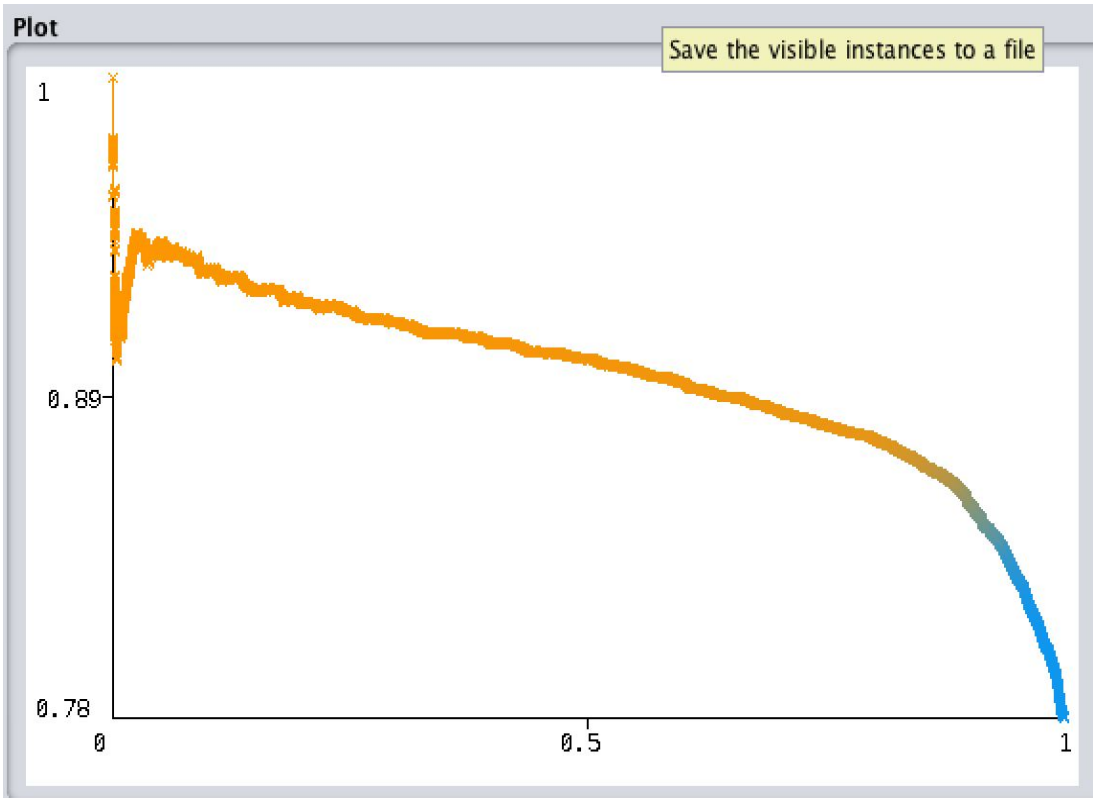
1. Naïve Bayes

```
Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       24055               80.1833 %
Incorrectly Classified Instances      5945               19.8167 %
Kappa statistic                          0.3777
Mean absolute error                      0.2268
Root mean squared error                  0.4051
Relative absolute error                 65.8242 %
Root relative squared error             97.5925 %
Total Number of Instances            30000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.903    0.554    0.852      0.903    0.876      0.382    0.754     0.896     0
                 0.446    0.097    0.566      0.446    0.499      0.382    0.754     0.511     1
Weighted Avg.    0.802    0.453    0.788      0.802    0.793      0.382    0.754     0.811
```

Summarized results of running Naïve Bayes classifier on the dataset.



ROC curve for Naïve Bayes classifier
Area under the curve=75.37%



Precision-recall curve for Naïve Bayes classifier

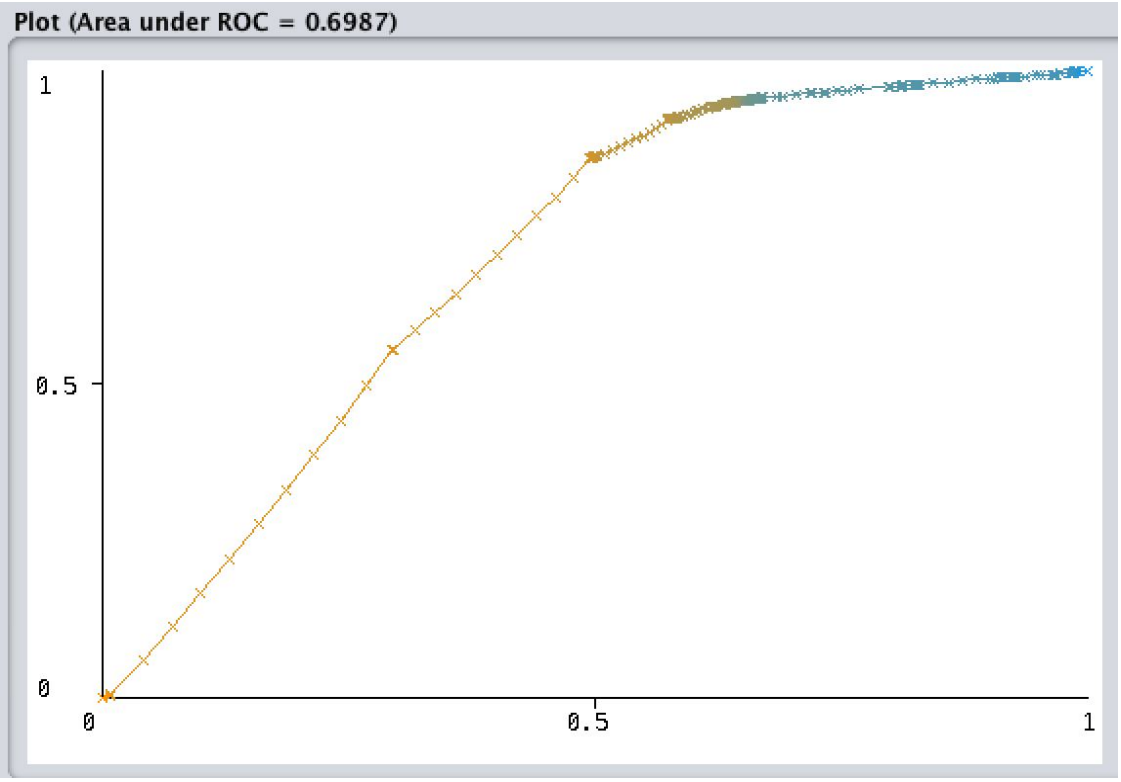
	Classified "Predicted Payment"	Classified "Predicted Default"
Actual Payment	21093	2271
Actual Default	3674	2962

Naïve Bayes Confusion Matrix

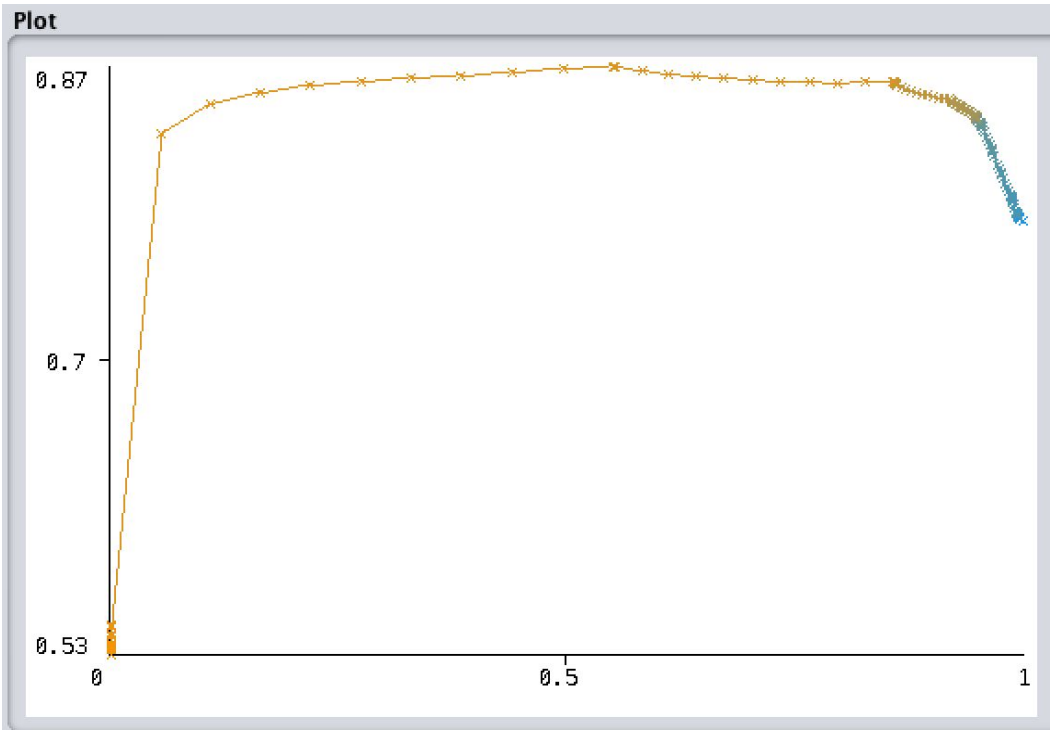
2. Decision Tree (J48)

```
Number of Leaves  :     404

Size of the tree :      459


Time taken to build model: 0.51 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       24538               81.7933 %
Incorrectly Classified Instances      5462               18.2067 %
Kappa statistic                          0.3641
Mean absolute error                      0.2804
Root mean squared error                  0.3794
Relative absolute error                 81.3736 %
Root relative squared error             91.404  %
Total Number of Instances            30000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.950    0.648    0.838      0.950    0.890      0.391    0.699     0.854     0
                 0.352    0.050    0.668      0.352    0.461      0.391    0.699     0.467     1
Weighted Avg.    0.818    0.515    0.800      0.818    0.796      0.391    0.699     0.768
```

Summarized results of running J48 decision tree classifier on the dataset.



ROC curve for J48 decision tree
Area under the curve=69.87%



Precision-recall curve for J48 decision tree

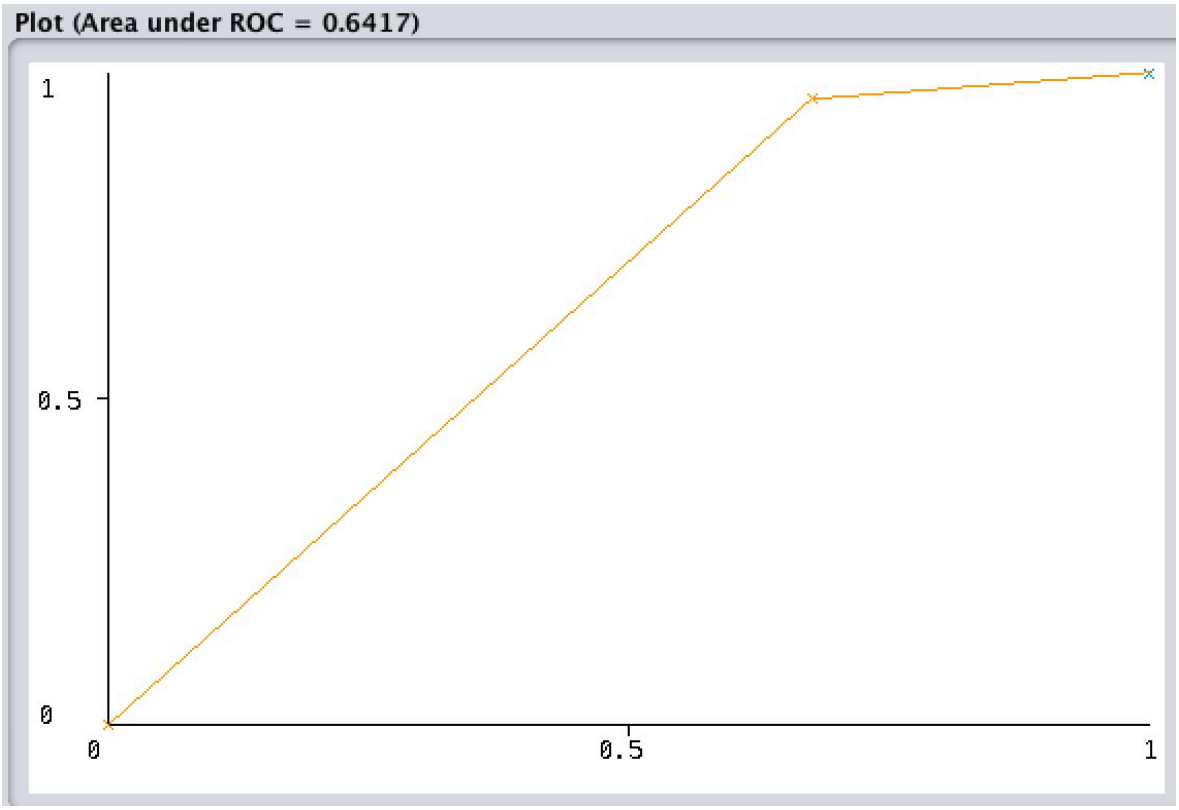
	Classified “Predicted Payment”	Classified “Predicted Payment”
Actual Payment	22199	1165
Actual Default	4297	2339

Decision tree Confusion Matrix

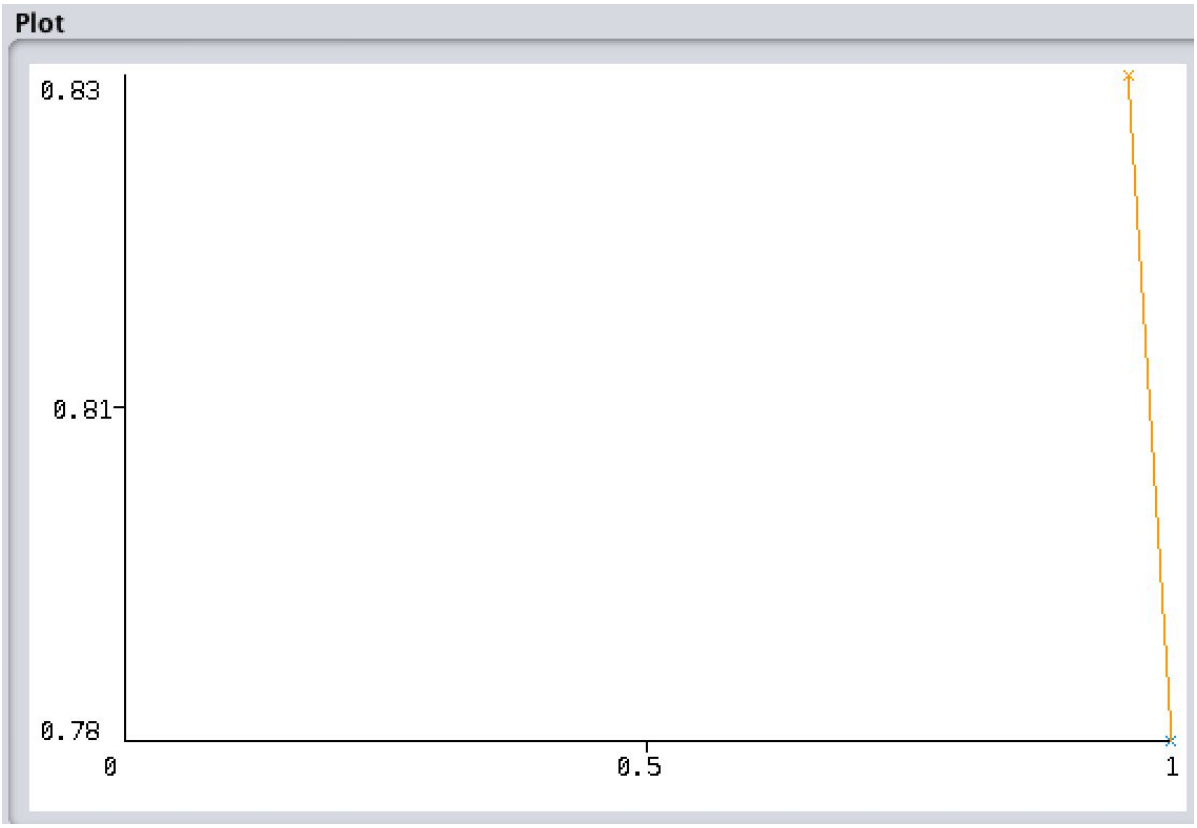
3. Support Vector Machine

```
=== Classifier model (full training set) ===

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 330.27 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       24576               81.92   %
Incorrectly Classified Instances      5424               18.08   %
Kappa statistic                          0.3506
Mean absolute error                      0.1808
Root mean squared error                  0.4252
Relative absolute error                 52.4738 %
Root relative squared error            102.4457 %
Total Number of Instances            30000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.960    0.677    0.833      0.960    0.892      0.388    0.642     0.831     0
                 0.323    0.040    0.697      0.323    0.442      0.388    0.642     0.375     1
Weighted Avg.    0.819    0.536    0.803      0.819    0.792      0.388    0.642     0.730
```

Summarized results of running LibSVM classifier on the dataset.



ROC curve for LibSVM Support Vector Machine classifier
Area under the curve=64.17%



Precision-recall curve for LibSVM Support Vector Machine classifier

	Classified “Predicted Payment”	Classified “Predicted Default”
Actual Payment	22431	933
Actual Default	4491	2145

Support Vector Machines Confusion Matrix

Experimental Results

After reviewing all models, the three different algorithms appeared to perform relatively similarly reaching an accuracy of just over 80 percent. Overall, the SVM model proved slightly to be the most accurate followed by the Decision Tree and the Naïve Bayes models respectively. However, the SVM model required significantly greater computation as illustrated by the times taken to build each of the models.

In the cases of the Naïve Bayes, Decision Tree, and SVM models, the times taken to construct the models were 0.02 seconds, 0.51 seconds, and 330.27 seconds respectively. While more accurate, the SVM required exponentially greater time and computational power to build, but its end performance did not equate to the time or power required.

Classifier	Accuracy	Error	Area Under ROC
Naïve Bayes	80.1833%	19.8167%	0.754
Decision Tree	81.7933%	18.2067%	0.699
Support Vector Machines	81.9200%	18.0800%	0.642

Based on the above analysis, we contend that with a large enough dataset, each of the aforementioned classification models are relatively effective in detecting credit card default. With only 30,000 entries from a single state and limited attributes observed, each of the models adequately predicted credit card delinquency within a reasonable margin of error. Given the significant difference in processing times, one could surmise that use of a decision tree model or Naïve Baye model may prove to be more time and cost effective than that of the Support Vector Machines.

However, the analysis of the SVM model brings begs the question as to whether or not the additional complexity of and accuracy of the classifier is

worth the additional time and computing power required to apply it. Based on our cursory research, we contend that implementing the Decision Tree or Naïve Bayes model may be more effective, but additional research could be conducted to develop a more robust conclusion. In instances where the amount of attributes observed, number of records collected, completeness of data, or alternative preprocessing measures may have a major impact on the model, which may warrant its usage over more efficient models.

CONCLUSION

In conclusion, all three models performed relatively similarly. However, SVM required almost an hour to process; while the other two took a few seconds. Based on the ROC curves generated, Naïve Bayes has the best performance, followed by the J48 decision tree classifier and the LibSVM Support Vector Machine classifier finishing in the last place.

Naïve Bayes performed the best as it generally does quite well even if provided with a small amount of training data to estimate the parameters. However, unlike an extensively optimized classifying algorithm, it wasn't very efficient as dependencies existed among variables in the dataset which could not be modeled by the classifier.

The J48 decision tree classifier was able to perform classification without too many calculations and dealt successfully with any incomplete data. The high classification error rate while training set was small in comparison with the number of classes as indicated by it's AUC value of 0.6987.

The LibSVM classifier is an efficient implementation of SVM and could perform multi-class classifications. It also performed well with training samples containing errors. But, its performance was slowest compared to the other two because its training-time is generally quite long.

This project could possibly be improved by having a more thorough dataset containing data from other countries, banks and consortiums. Additional preprocessing on more powerful computers would be certainly helpful.

Ultimately, it can be contended that as collected data on individuals delinquent on credit card payments increases over time, each of the aforementioned models may be used in the accurate detection of at-risk borrowers.