King Abdul-Aziz University
Faculty of Computing and Information Technology
Computer Science Department
Artificial Intelligence I – CPCS-331 | Fall 2023

BANK MARKETING DATA SET
- Machine Learning -

Instructor: Ms. Noha Alnahdi

Hands in Date

November 6, 2022

Project Team:

Name	ID	Section
Razan Arif Alamri		B3A
Shatha Khalid Binmahfouz		

Task Assignment

Team Member	Contribution
Razan Arif Alamri	• Describe The Dataset Chosen • Random Forest algorithm • Conclusion
Shatha Khalid Binmahfouz	• Introduction • SVM algorithm • Conclusion

Table of Contents:

1. Introduction

Machine learning is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behavior. In addition, Machine learning is significant for the development of new goods as well as for providing businesses with a trends in customer behavior and business operational patterns.

1.1 Problem Explanation

The problem we aim to solve is to find the appropriate in machine learning algorithm out of these two methods: Random forest and Support vector Machine (SVM). We chose it to build a model on our dataset as accurately as possible by using both Weka and RapidMiner.

1.2 Purpose Of The Project

The purpose of our project is to introduce the concept of the machine learning algorithm and how to implement and calculate its accuracy by using split and cross validation.

1.3 Outline The Approach

- Select a dataset that shows some attributes.
- Implementation to calculate the accuracy by using both Weka and RapidMiner.
- Test the dataset by use both split and cross validation

2. Technical description

2.1 Describe The Dataset Chosen

Bank Marketing Data Set:

We are analyzing phone call-based marketing data from a banking institution. Potential clients are approached by phone to determine whether or not to subscribe to the bank term deposit. Advertising, selling, and delivering things to customers or other businesses is all part of marketing.

The data is related to direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact with the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The classification purpose is to expect if the client will subscribe (yes/no) a term deposit (variable y).

Attribute Information:

Input variables:

# bank client data:
1 - age (numeric)
2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
5 - default: has credit in default? (categorical: 'no','yes','unknown')
6 - housing: has housing loan? (categorical: 'no','yes','unknown')
7 - loan: has personal loan? (categorical: 'no','yes','unknown')
# related with the last contact of the current campaign:
8 - contact: contact communication type (categorical: 'cellular','telephone')

9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

# other attributes:

12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
# social and economic context attributes
16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
17 - cons.price.idx: consumer price index - monthly indicator (numeric)
18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
20 - nr.employed: number of employees - quarterly indicator (numeric)

Output variable (desired target):

21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

2.2 **Describe The Algorithm Chosen**

2.2.1 Random Forest

Random Forest is a well-known machine learning algorithm from the supervised learning approach. It may be applied to both classification and regression issues in machine learning. It is built on the notion of ensemble learning, which is an approach that entails integrating several classifiers to solve a complicated issue and enhance the model's performance.

Random Forest is a classifier that includes a set of decision trees on various subsets of the provided dataset and takes the average to enhance the estimate accuracy of that dataset. Instead, than depending on a single decision tree, the random forest considers the forecast from each tree and estimates the final results based on the majority vote of estimations.

2.2.2 SVM

Support vector machines are a set of supervised learning methods used for classification, regression, clustering and outliers' detection. Large data sets are ineffective for this method. SVMs vary from other machine learning algorithms in that they choose a decision boundary that maximizes the distance between the nearest data points for all classes.

There are more terms to understand SVM mathematically:

- **Support vectors** are special data points in the dataset and its help in decreasing and increasing the size of the boundaries.
- **Hyperplane** is the central line of the diagram.
- **Decision boundaries** in SVM are the two lines that we see alongside the hyperplane.

3. Results

The details of the source code and the results of the experiments are in Appendix.

3.1 Results Of Random Forest Algorithm

3.1.1 Cross Validation:

<table>
<tr><th rowspan="2"></th><th colspan="2">Cross Validation (1)</th></tr>
<tr><th>Weka</th><th>RapidMiner</th></tr>
<tr><td>Cross-Validation Folds</td><td colspan="2">10</td></tr>
<tr><td>Accuracy</td><td>90.3895 %</td><td>89.74%</td></tr>
<tr><td>Confusion Matrix</td><td>=== Confusion Matrix ===
a b <-- classified as
38601 1321 | a = no
3024 2265 | b = yes</td><td>ConfusionMatrix:
True: no yes
no: 39377 4093
yes: 545 1196</td></tr>
</table>

<table>
<tr><th rowspan="2"></th><th colspan="2">Cross Validation (2)</th></tr>
<tr><th>Weka</th><th>RapidMiner</th></tr>
<tr><td>Cross-Validation Folds</td><td colspan="2">20</td></tr>
<tr><td>Accuracy</td><td>90.5266 %</td><td>89.78%</td></tr>
<tr><td>Confusion Matrix</td><td>=== Confusion Matrix ===
a b <-- classified as
38632 1290 | a = no
2993 2296 | b = yes</td><td>ConfusionMatrix:
True: no yes
no: 39407 4105
yes: 515 1184</td></tr>
</table>

3.1.2 Split Validation:

	Split Validation (2)	
	Weka	**RapidMiner**
Percentage-Split	66.0%	
Accuracy	90.3201 %	89.73%
Confusion Matrix	=== Confusion Matrix === a b <-- classified as 13135 428 | a = no 1060 749 | b = yes	ConfusionMatrix: True: no yes no: 13424 1429 yes: 149 369

	Split Validation (2)	
	Weka	**RapidMiner**
Percentage-Split	76.0%	
Accuracy	90.5539 %	89.85%
Confusion Matrix	=== Confusion Matrix === a b <-- classified as 9268 300 | a = no 725 558 | b = yes	ConfusionMatrix: True: no yes no: 9489 1009 yes: 92 260

3.1.3 Analyze Result

Using the Random Forest algorithm and based on our results, we observed that the results using Weka were more accurate than those obtained with RapidMinar. We also observed that both validations provided results close to some, but that split validation was better than cross validation for accuracy.

3.2 Results Of SVM Algorithm

3.2.1 Cross Validation:

	Cross Validation (1)	
	Weka	RapidMiner
Cross-Validation Folds	10	
Accuracy	82.8626 %	88.91%
Confusion Matrix	=== Confusion Matrix === a b <-- classified as 35218 4704 | a = no 3044 2245 | b = yes	ConfusionMatrix: True: no yes no: 39181 4272 yes: 741 1017

	Cross Validation (2)	
	Weka	RapidMiner
Cross-Validation Folds	20	
Accuracy	88.0361 %	88.97%
Confusion Matrix	=== Confusion Matrix === a b <-- classified as 38503 1419 | a = no 3990 1299 | b = yes	ConfusionMatrix: True: no yes no: 39191 4257 yes: 731 1032

3.2.2 Split Validation:

	Split Validation (2)	
	Weka	**RapidMiner**
Percentage-Split	66.0%	
Accuracy	77.2248 %	88.70%
Confusion Matrix	=== Confusion Matrix === a b <-- classified as 10741 2822 | a = no 679 1130 | b = yes	ConfusionMatrix: True: no yes no: 13324 1488 yes: 249 310

	Split Validation (2)	
	Weka	**RapidMiner**
Percentage-Split	76.0%	
Accuracy	65.5516 %	88.72%
Confusion Matrix	=== Confusion Matrix === a b <-- classified as 5978 3590 | a = no 148 1135 | b = yes	ConfusionMatrix: True: no yes no: 9399 1042 yes: 182 227

3.2.3 Analyze Result

Using the SVM algorithm and based on our results, we observed that the results using RapidMiner were more accurate than those obtained with Weka. We also observed that both validations provided results close to some, but that cross validation was better than split validation for accuracy.

4. Conclusion

	Random Forest Algorithm				SVM Algorithm			
	Cross Validation 10 folds	Cross Validation 20 folds	Split Validation 66%	Split Validation 76%	Cross Validation 10 folds	Cross Validation 20 folds	Split Validation 66%	Split Validation 76%
Weka	90.3895%	90.5266%	90.3201%	90.5539%	82.8626%	88.0361%	77.2248%	65.5516%
RapidMiner	89.74%	89.78%	89.73%	89.85%	88.91%	88.97%	88.70%	88.72%

Finally, after comparing the results of our experiments with the Random Forest and SVM algorithms using both Weka and RapidMiner, it was concluded that the random forest algorithm is the most accurate as its accuracy in both validations is about 90%, but in the SVM algorithm its accuracy varies up to about 88% in both validations.

Since the number of data we have is big over 40,000 the random forest algorithm is the best because regardless of the size of the data does not affect its behaviour. In the SVM algorithm, it is the best choice if the data are small.

5. References

I. Machine Learning Repository. (n.d.). UCI Machine Learning Repository: Bank Marketing Data Set. Retrieved November 5, 2022, from https://archive.ics.uci.edu/ml/datasets/Bank+Marketing

II. McGregor, M. (2020, July 2). *SVM machine learning tutorial – what is the support vector machine algorithm, explained with code examples*. freeCodeCamp.org. Retrieved November 5, 2022, from https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/

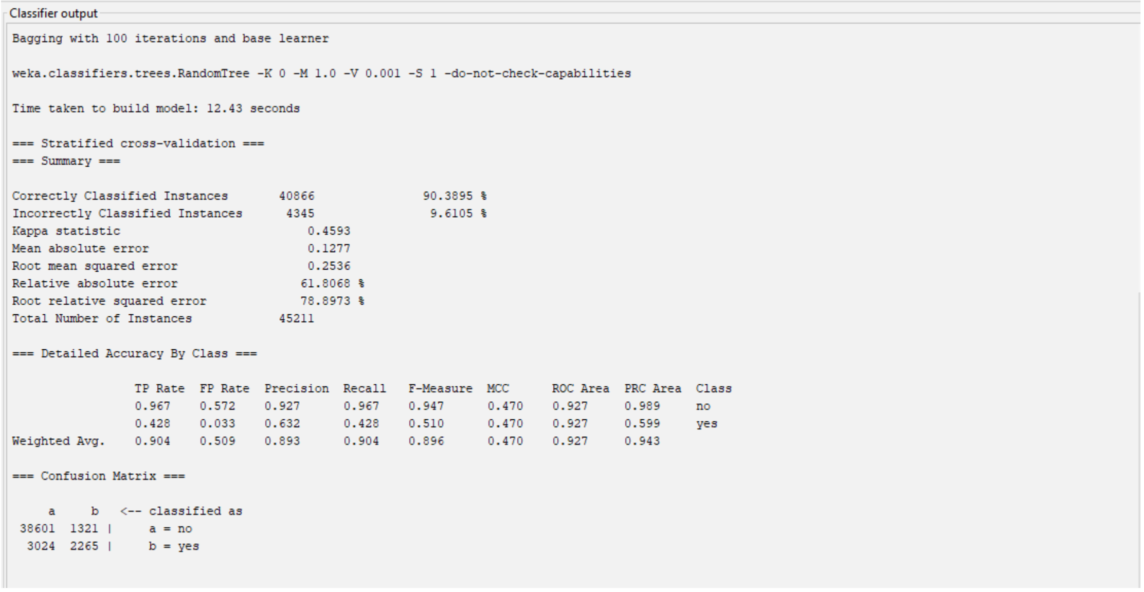
III. Mbaabu, O. (n.d.). *Introduction to random forest in machine learning*. Section. Retrieved November 5, 2022, from https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

IV. Donges, N. (n.d.). *Random forest classifier: A complete guide to how it works in Machine Learning*. Built In. Retrieved November 5, 2022, from https://builtin.com/data-science/random-forest-algorithm

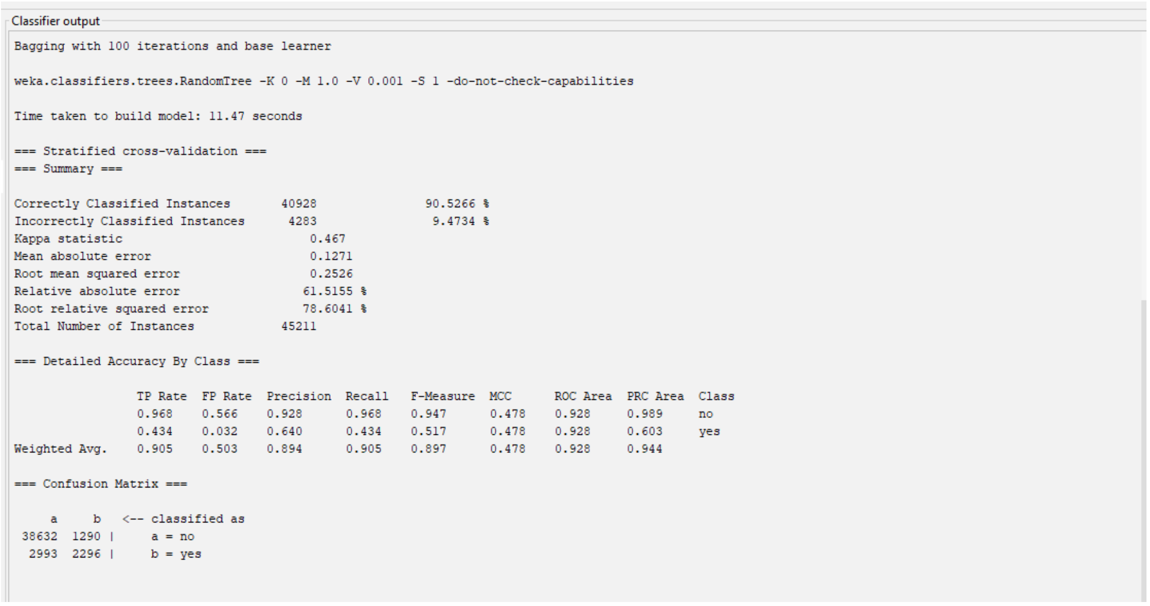
6. Appendix

6.1 Screenshots Of Random Forest Algorithm Results

6.1.1 Weka Cross Validation

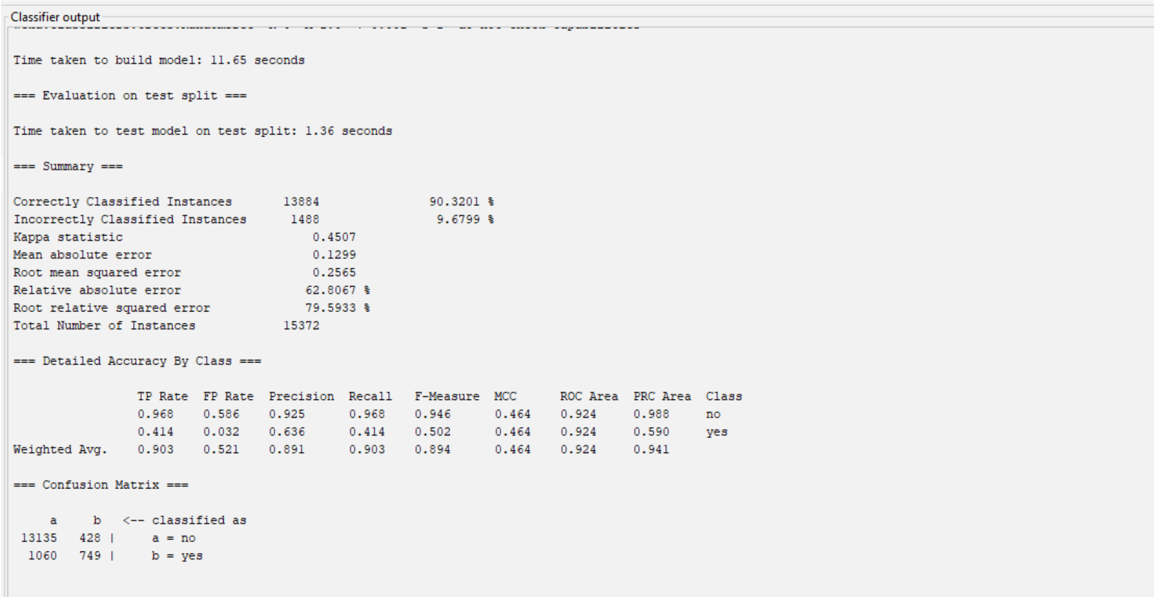


```
Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 12.43 seconds

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

Correctly Classified Instances       40866               90.3895 %
Incorrectly Classified Instances      4345                9.6105 %
Kappa statistic                          0.4593
Mean absolute error                      0.1277
Root mean squared error                  0.2536
Relative absolute error                 61.8068 %
Root relative squared error             78.8973 %
Total Number of Instances            45211

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.967    0.572    0.927      0.967    0.947      0.470    0.927     0.989     no
                 0.428    0.033    0.632      0.428    0.510      0.470    0.927     0.599     yes
Weighted Avg.    0.904    0.509    0.893      0.904    0.896      0.470    0.927     0.943

=== Confusion Matrix ===

     a     b   <-- classified as
 38601  1321 |     a = no
  3024  2265 |     b = yes
```

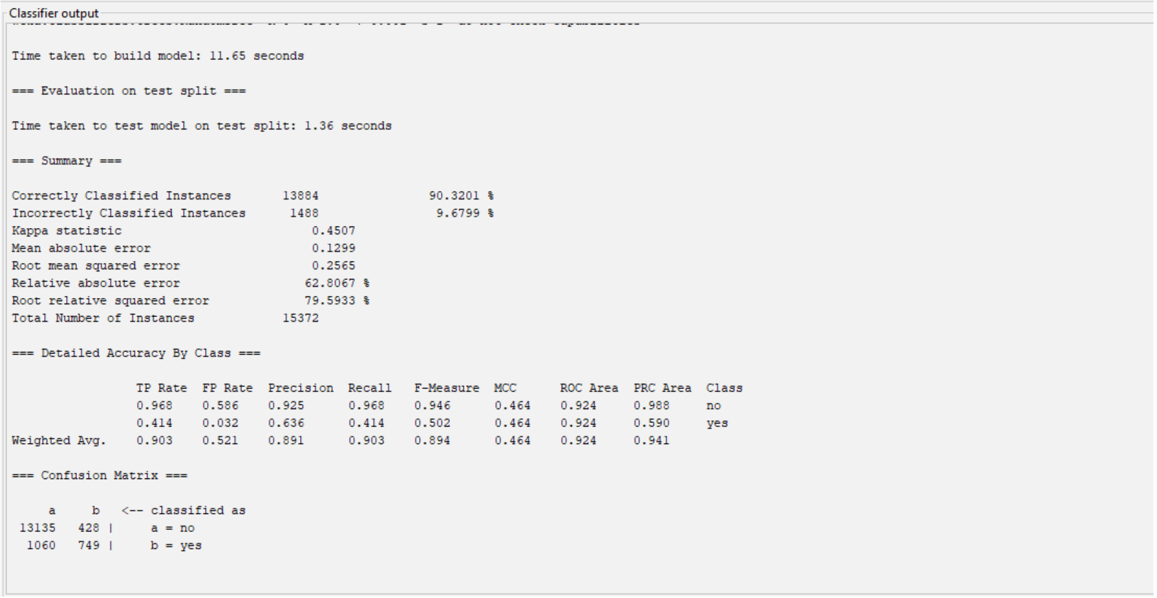


```
Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 11.47 seconds

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

Correctly Classified Instances       40928               90.5266 %
Incorrectly Classified Instances      4283                9.4734 %
Kappa statistic                          0.467
Mean absolute error                      0.1271
Root mean squared error                  0.2526
Relative absolute error                 61.5155 %
Root relative squared error             78.6041 %
Total Number of Instances            45211

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.968    0.566    0.928      0.968    0.947      0.478    0.928     0.989     no
                 0.434    0.032    0.640      0.434    0.517      0.478    0.928     0.603     yes
Weighted Avg.    0.905    0.503    0.894      0.905    0.897      0.478    0.928     0.944

=== Confusion Matrix ===

     a     b   <-- classified as
 38632  1290 |     a = no
  2993  2296 |     b = yes
```

6.1.2 Weka Split Validation



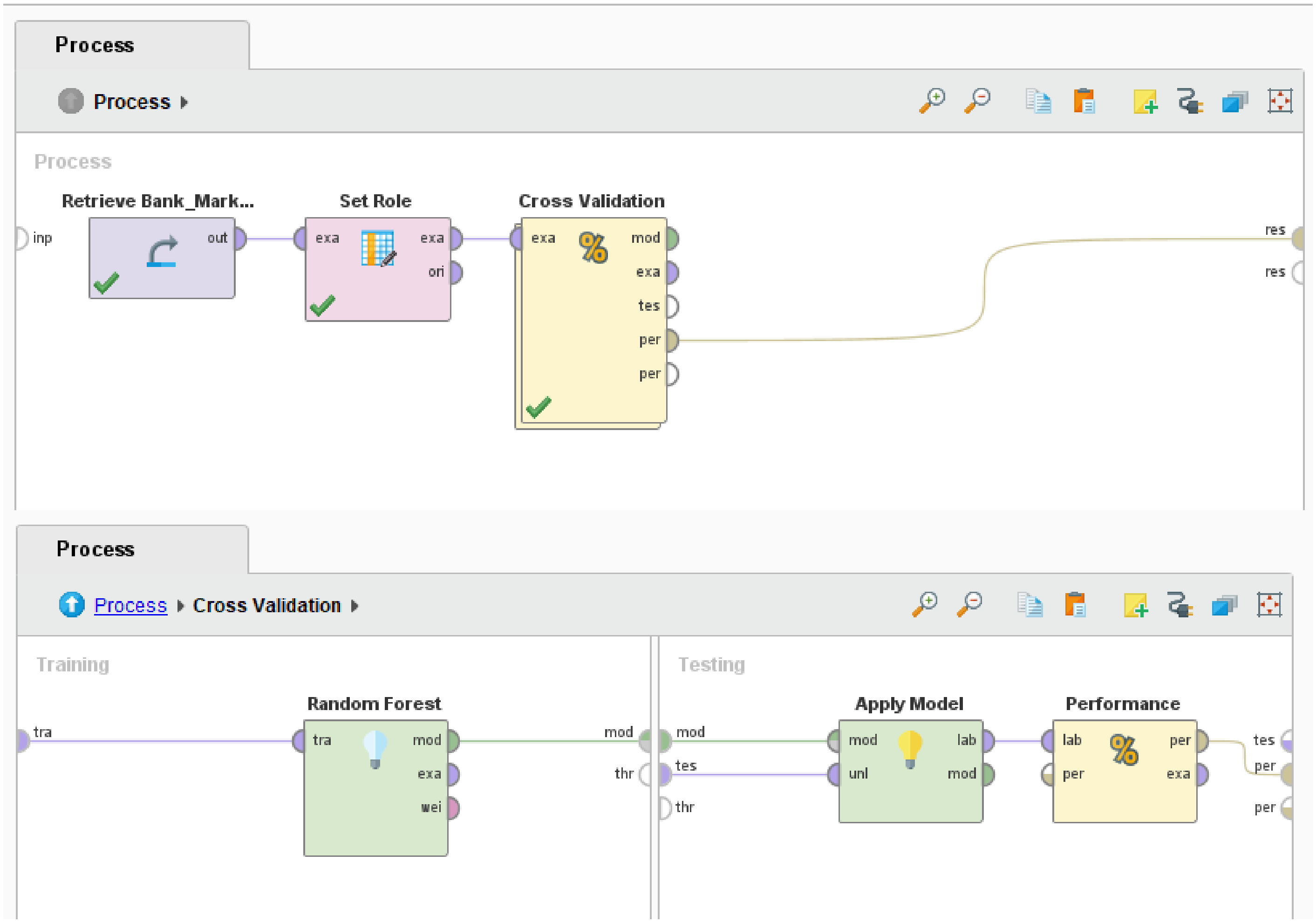
```
Time taken to build model: 11.65 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 1.36 seconds

=== Summary ===

Correctly Classified Instances       13884               90.3201 %
Incorrectly Classified Instances      1488                9.6799 %
Kappa statistic                          0.4507
Mean absolute error                      0.1299
Root mean squared error                  0.2565
Relative absolute error                 62.8067 %
Root relative squared error             79.5933 %
Total Number of Instances            15372

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.968    0.586    0.925      0.968    0.946      0.464    0.924     0.988     no
                 0.414    0.032    0.636      0.414    0.502      0.464    0.924     0.590     yes
Weighted Avg.    0.903    0.521    0.891      0.903    0.894      0.464    0.924     0.941

=== Confusion Matrix ===

     a     b   <-- classified as
 13135   428 |     a = no
  1060   749 |     b = yes
```

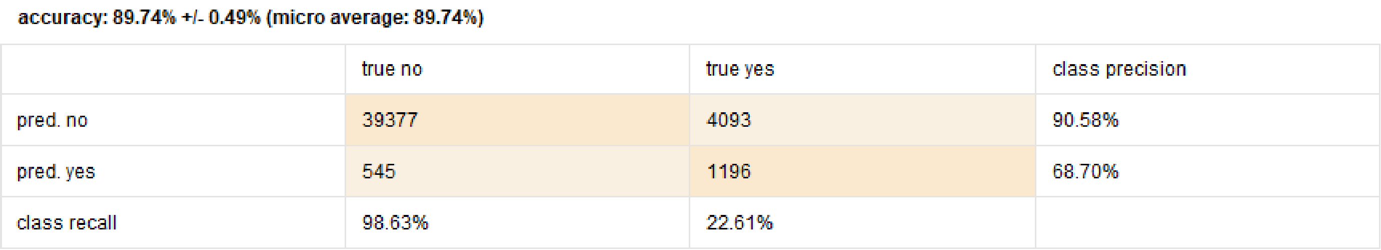


Classifier output

```
Time taken to build model: 11.65 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 1.36 seconds

=== Summary ===

Correctly Classified Instances       13884               90.3201 %
Incorrectly Classified Instances      1488                9.6799 %
Kappa statistic                          0.4507
Mean absolute error                      0.1299
Root mean squared error                  0.2565
Relative absolute error                 62.8067 %
Root relative squared error             79.5933 %
Total Number of Instances            15372

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.968    0.586    0.925      0.968    0.946      0.464    0.924     0.988     no
                 0.414    0.032    0.636      0.414    0.502      0.464    0.924     0.590     yes
Weighted Avg.    0.903    0.521    0.891      0.903    0.894      0.464    0.924     0.941

=== Confusion Matrix ===

     a     b   <-- classified as
 13135   428 |     a = no
  1060   749 |     b = yes
```

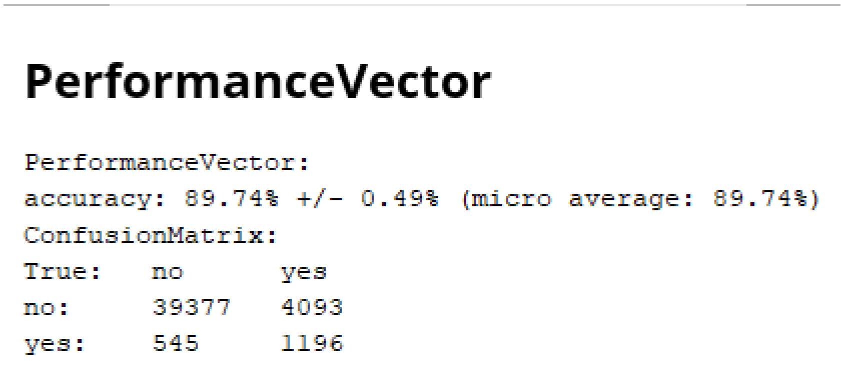
6.1.3 RapidMiner Cross Validation





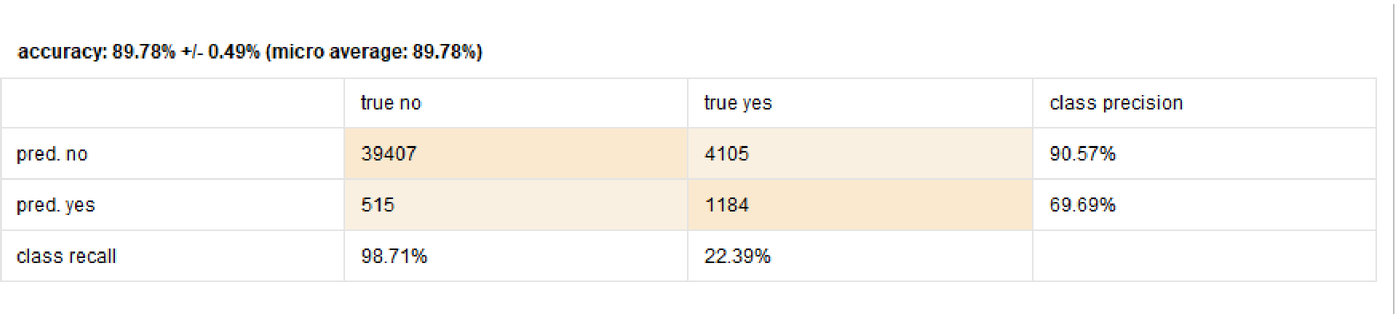
accuracy: 89.74% +/- 0.49% (micro average: 89.74%)

	true no	true yes	class precision
pred. no	39377	4093	90.58%
pred. yes	545	1196	68.70%
class recall	98.63%	22.61%	



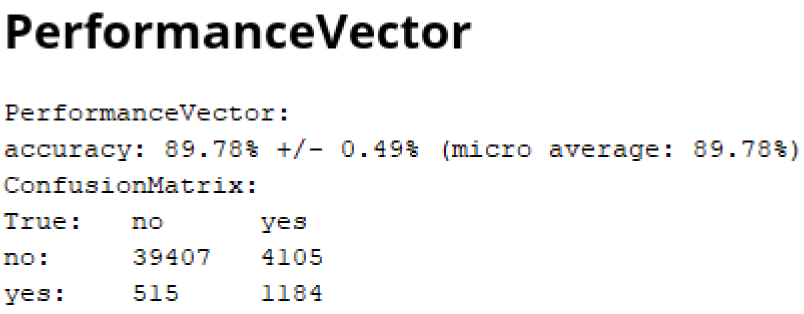
PerformanceVector

```
PerformanceVector:
accuracy: 89.74% +/- 0.49% (micro average: 89.74%)
ConfusionMatrix:
True:   no      yes
no:     39377   4093
yes:    545     1196
```



accuracy: 89.78% +/- 0.49% (micro average: 89.78%)

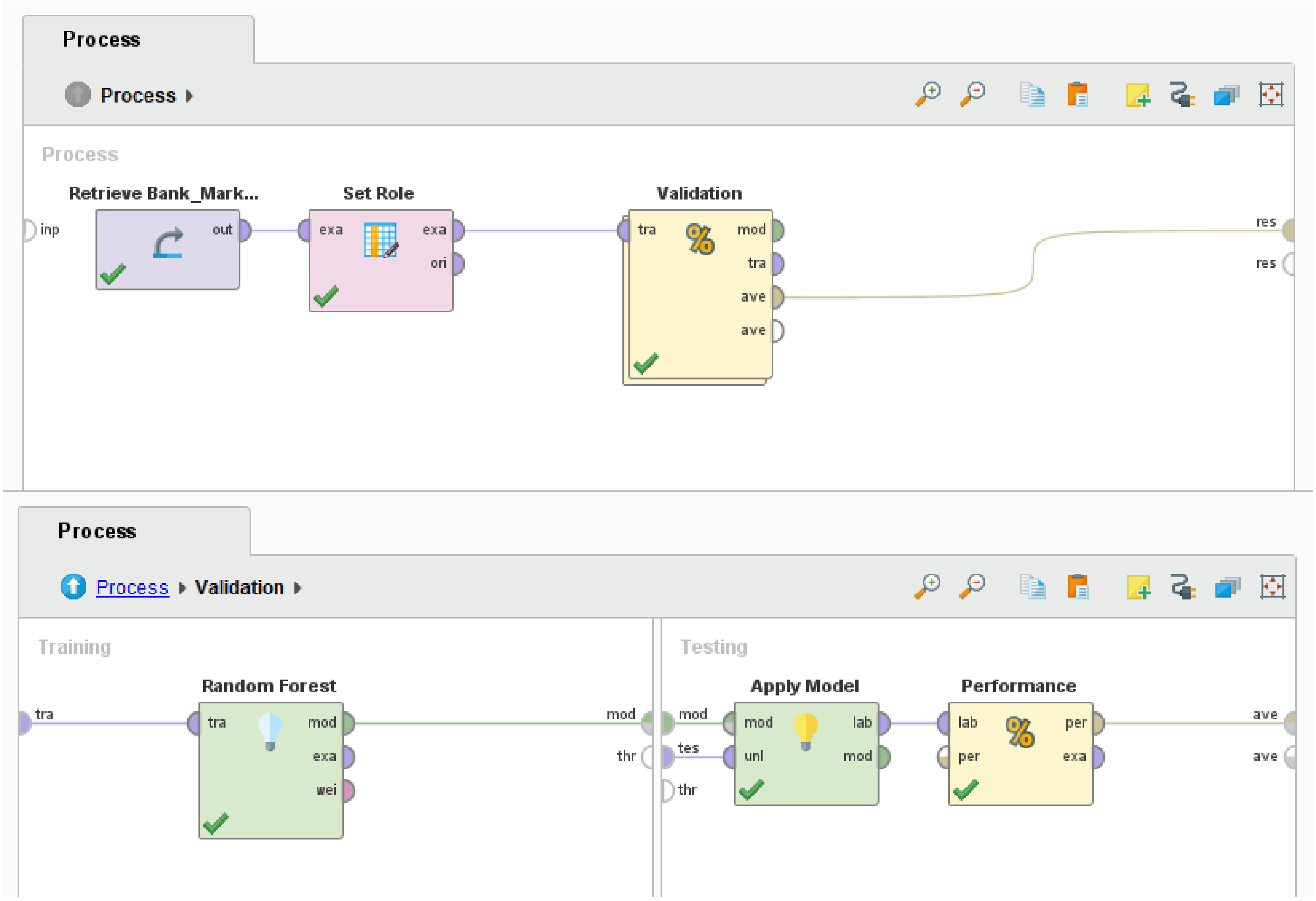
	true no	true yes	class precision
pred. no	39407	4105	90.57%
pred. yes	515	1184	69.69%
class recall	98.71%	22.39%	

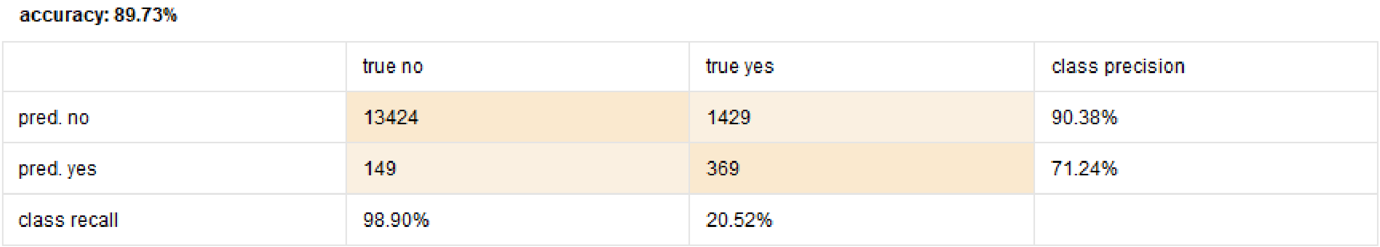


PerformanceVector

```
PerformanceVector:
accuracy: 89.78% +/- 0.49% (micro average: 89.78%)
ConfusionMatrix:
True:   no      yes
no:     39407   4105
yes:    515     1184
```

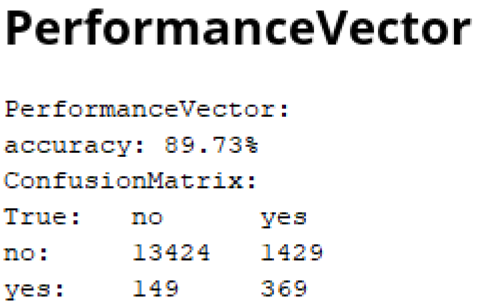
6.1.4 RapidMiner Split Validation





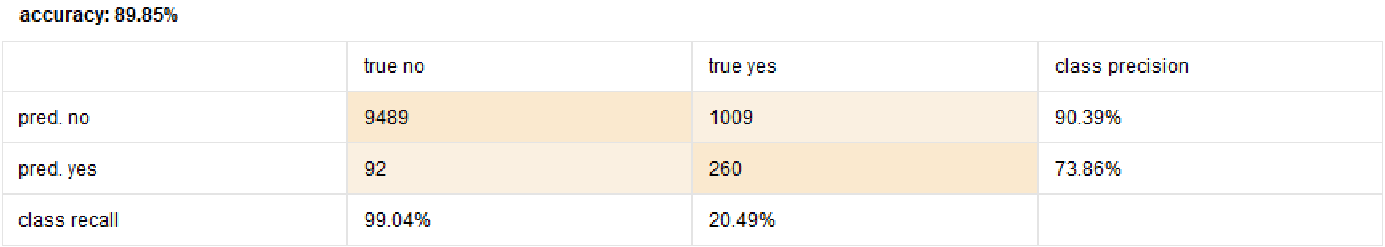
accuracy: 89.73%

	true no	true yes	class precision
pred. no	13424	1429	90.38%
pred. yes	149	369	71.24%
class recall	98.90%	20.52%	



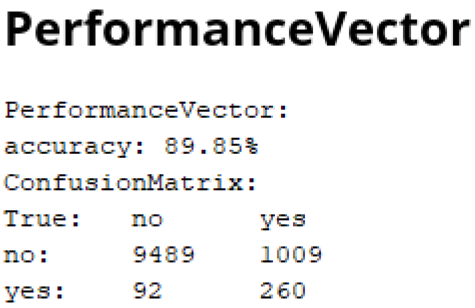
PerformanceVector

```
PerformanceVector:
accuracy: 89.73%
ConfusionMatrix:
True:   no      yes
no:     13424   1429
yes:    149     369
```



accuracy: 89.85%

	true no	true yes	class precision
pred. no	9489	1009	90.39%
pred. yes	92	260	73.86%
class recall	99.04%	20.49%	

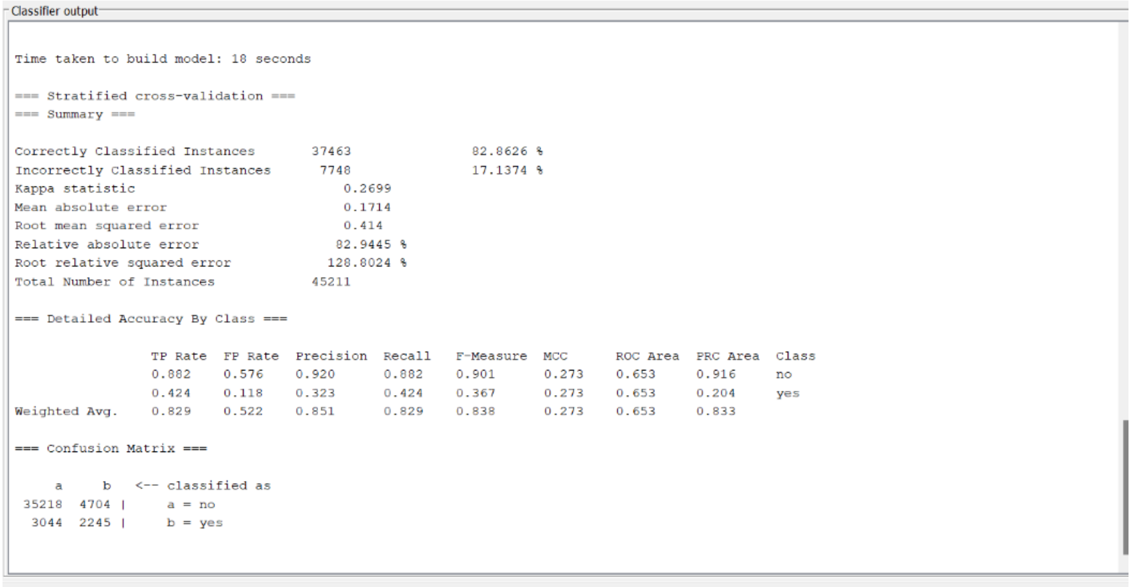


PerformanceVector

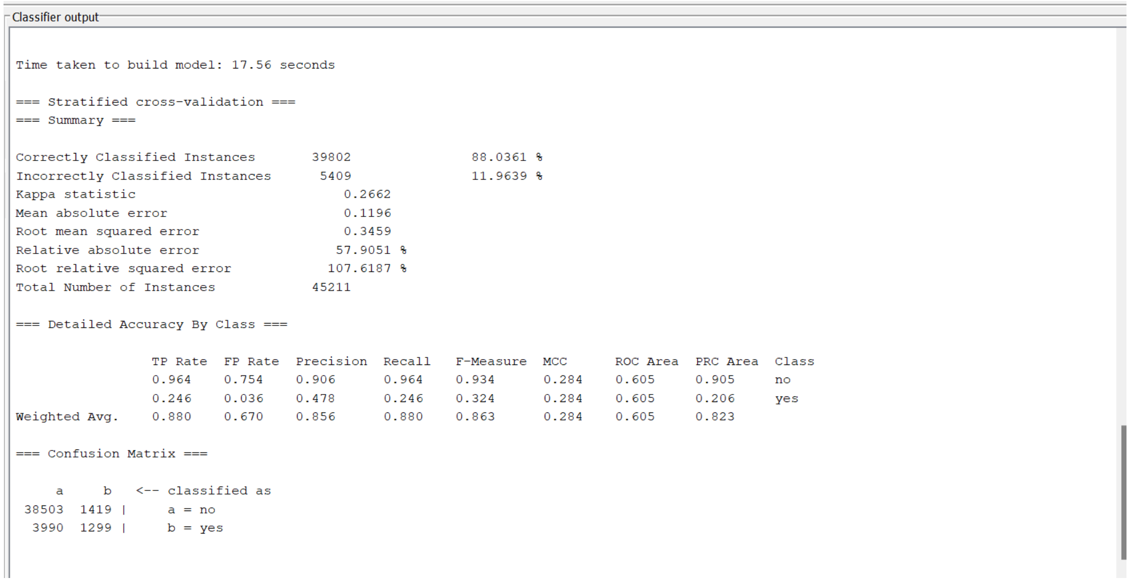
```
PerformanceVector:
accuracy: 89.85%
ConfusionMatrix:
True:   no      yes
no:     9489    1009
yes:    92      260
```

6.2 **Screenshots Of SVM Algorithm Results**

6.2.1 Weka Cross Validation

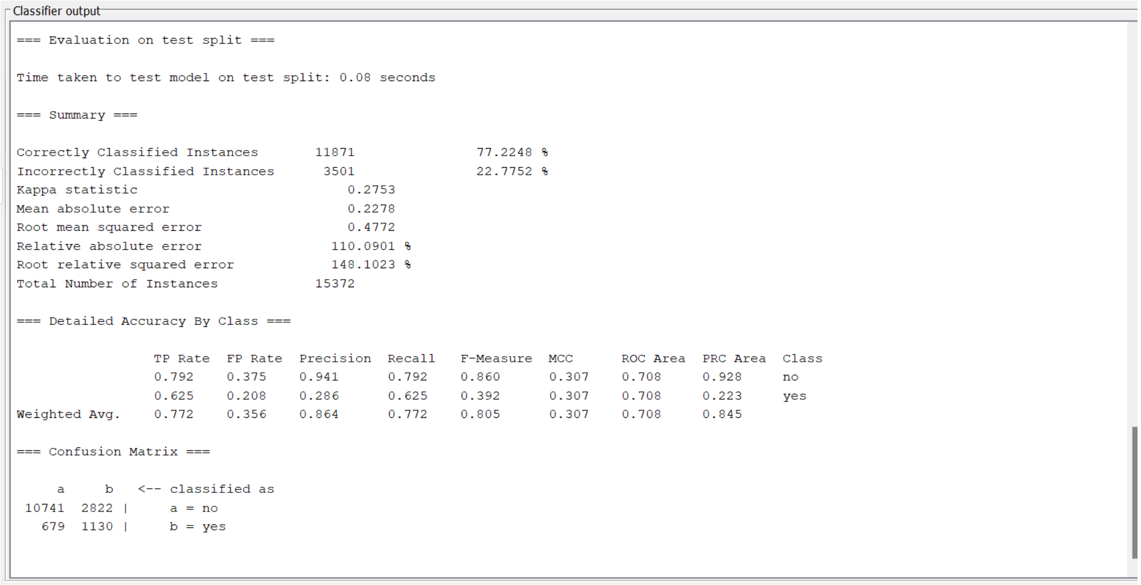


```
Time taken to build model: 18 seconds

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

Correctly Classified Instances       37463               82.8626 %
Incorrectly Classified Instances      7748               17.1374 %
Kappa statistic                          0.2699
Mean absolute error                      0.1714
Root mean squared error                  0.414
Relative absolute error                 82.9445 %
Root relative squared error            128.8024 %
Total Number of Instances            45211

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.882    0.576    0.920      0.882    0.901      0.273    0.653     0.916     no
                 0.424    0.118    0.323      0.424    0.367      0.273    0.653     0.204     yes
Weighted Avg.    0.829    0.522    0.851      0.829    0.838      0.273    0.653     0.833

=== Confusion Matrix ===

     a     b   <-- classified as
 35218  4704 |     a = no
  3044  2245 |     b = yes
```

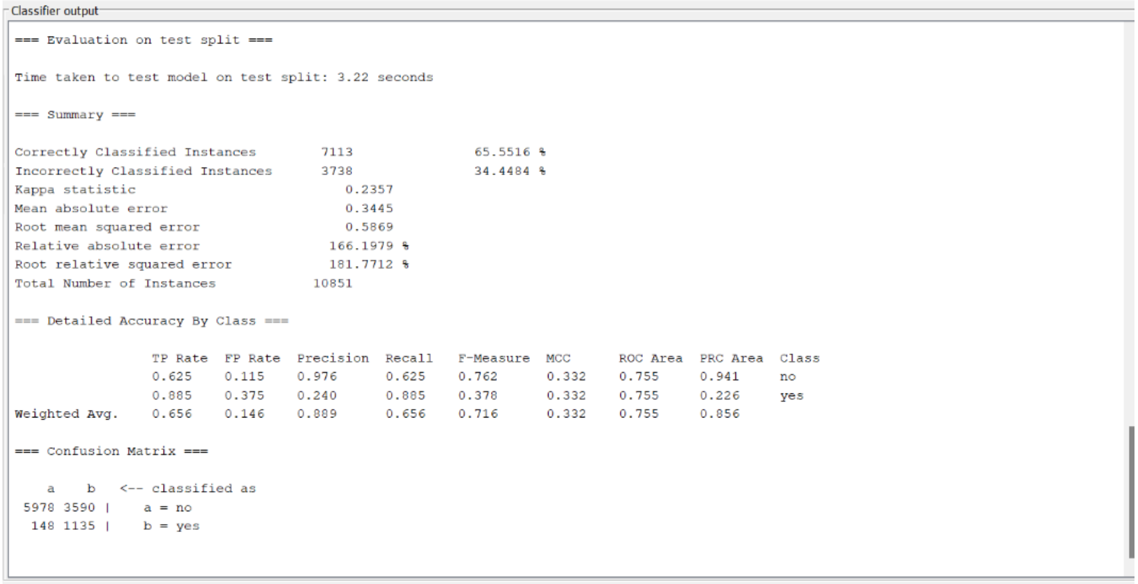


```
Time taken to build model: 17.56 seconds

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

Correctly Classified Instances       39802               88.0361 %
Incorrectly Classified Instances      5409               11.9639 %
Kappa statistic                          0.2662
Mean absolute error                      0.1196
Root mean squared error                  0.3459
Relative absolute error                 57.9051 %
Root relative squared error            107.6187 %
Total Number of Instances            45211

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.964    0.754    0.906      0.964    0.934      0.284    0.605     0.905     no
                 0.246    0.036    0.478      0.246    0.324      0.284    0.605     0.206     yes
Weighted Avg.    0.880    0.670    0.856      0.880    0.863      0.284    0.605     0.823

=== Confusion Matrix ===

     a     b   <-- classified as
 38503  1419 |     a = no
  3990  1299 |     b = yes
```

6.2.2 Weka Split Validation

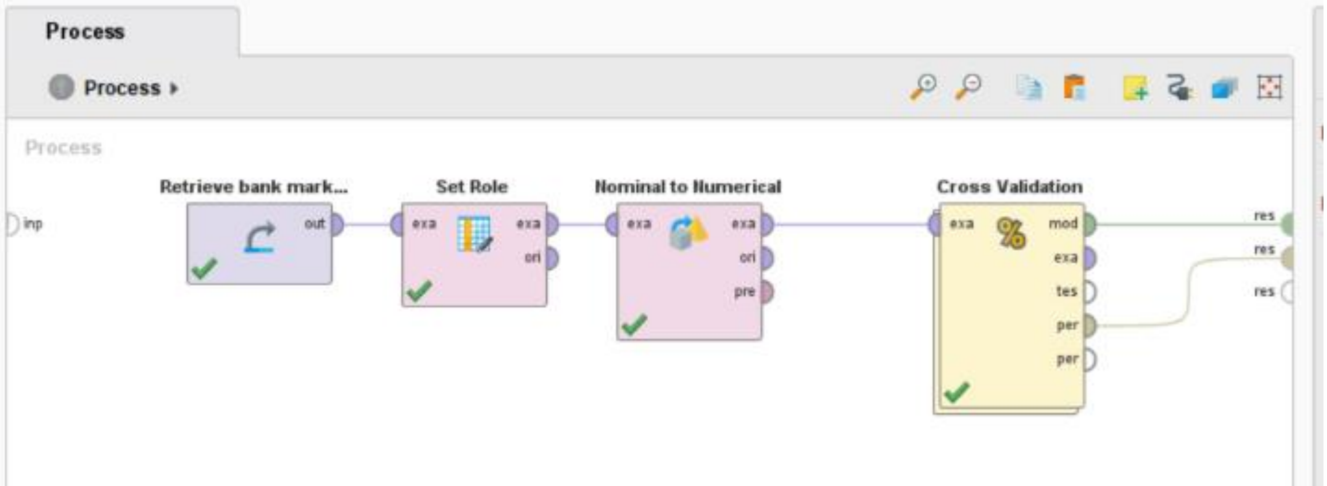


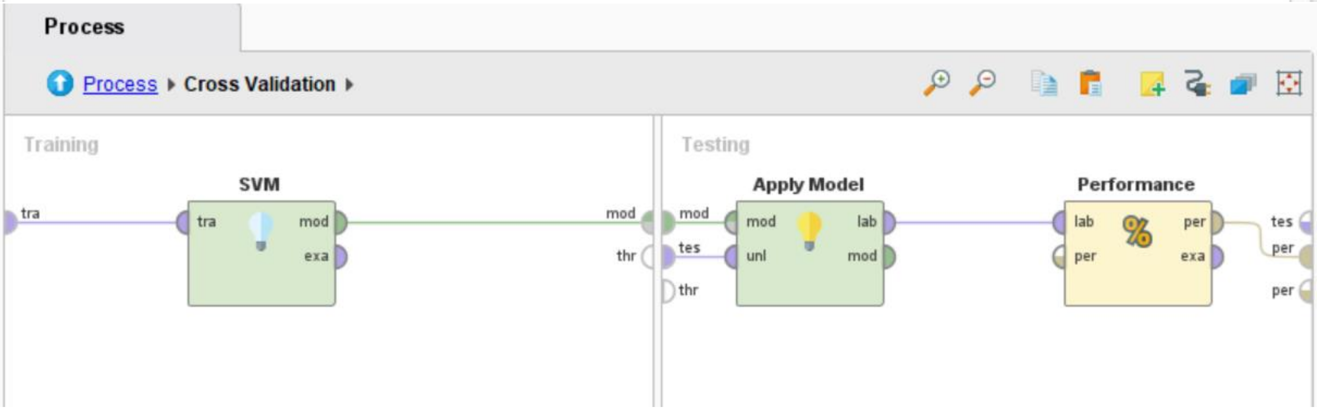
```
=== Evaluation on test split ===

Time taken to test model on test split: 0.08 seconds

=== Summary ===

Correctly Classified Instances       11871               77.2248 %
Incorrectly Classified Instances      3501               22.7752 %
Kappa statistic                          0.2753
Mean absolute error                      0.2278
Root mean squared error                  0.4772
Relative absolute error                110.0901 %
Root relative squared error            148.1023 %
Total Number of Instances            15372

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.792    0.375    0.941      0.792    0.860      0.307    0.708     0.928     no
                 0.625    0.208    0.286      0.625    0.392      0.307    0.708     0.223     yes
Weighted Avg.    0.772    0.356    0.864      0.772    0.805      0.307    0.708     0.845

=== Confusion Matrix ===

     a     b   <-- classified as
 10741  2822 |     a = no
   679  1130 |     b = yes
```

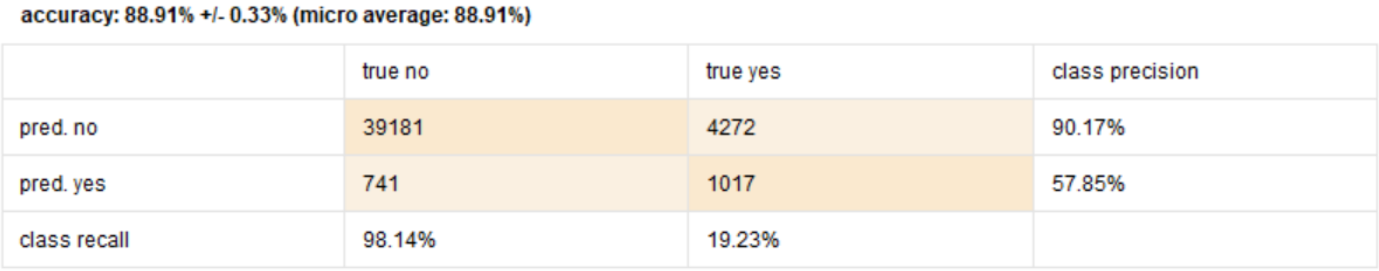


```
=== Evaluation on test split ===

Time taken to test model on test split: 3.22 seconds

=== Summary ===

Correctly Classified Instances        7113               65.5516 %
Incorrectly Classified Instances      3738               34.4484 %
Kappa statistic                          0.2357
Mean absolute error                      0.3445
Root mean squared error                  0.5869
Relative absolute error                166.1979 %
Root relative squared error            181.7712 %
Total Number of Instances            10851

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.625    0.115    0.976      0.625    0.762      0.332    0.755     0.941     no
                 0.885    0.375    0.240      0.885    0.378      0.332    0.755     0.226     yes
Weighted Avg.    0.656    0.146    0.889      0.656    0.716      0.332    0.755     0.856

=== Confusion Matrix ===

    a    b   <-- classified as
 5978 3590 |    a = no
  148 1135 |    b = yes
```

6.2.3 RapidMiner Cross Validation

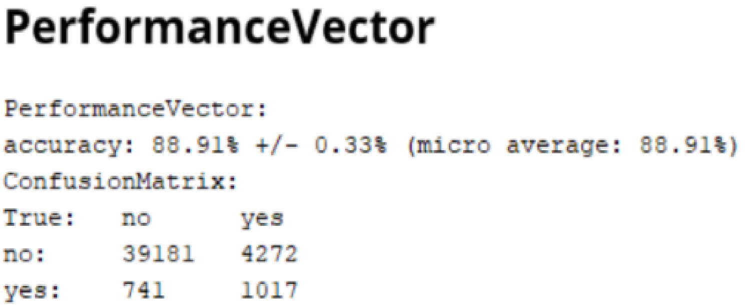






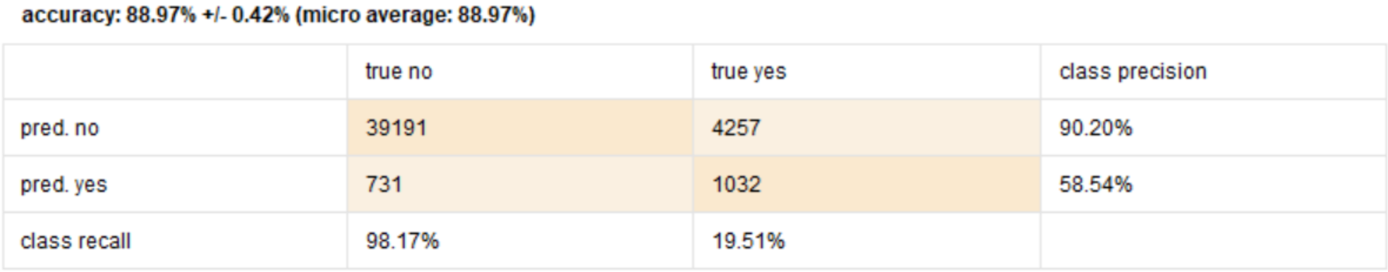
accuracy: 88.91% +/- 0.33% (micro average: 88.91%)

	true no	true yes	class precision
pred. no	39181	4272	90.17%
pred. yes	741	1017	57.85%
class recall	98.14%	19.23%	



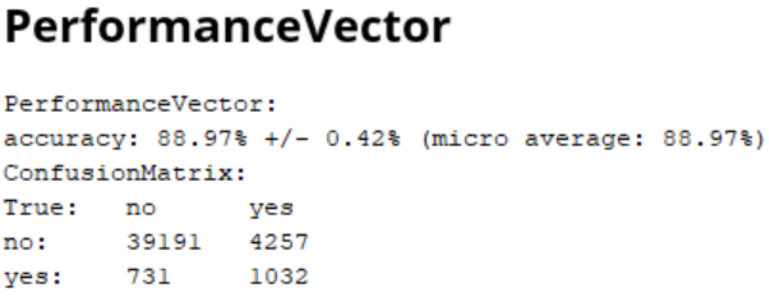
PerformanceVector

```
PerformanceVector:
accuracy: 88.91% +/- 0.33% (micro average: 88.91%)
ConfusionMatrix:
True:   no      yes
no:     39181   4272
yes:    741     1017
```



accuracy: 88.97% +/- 0.42% (micro average: 88.97%)

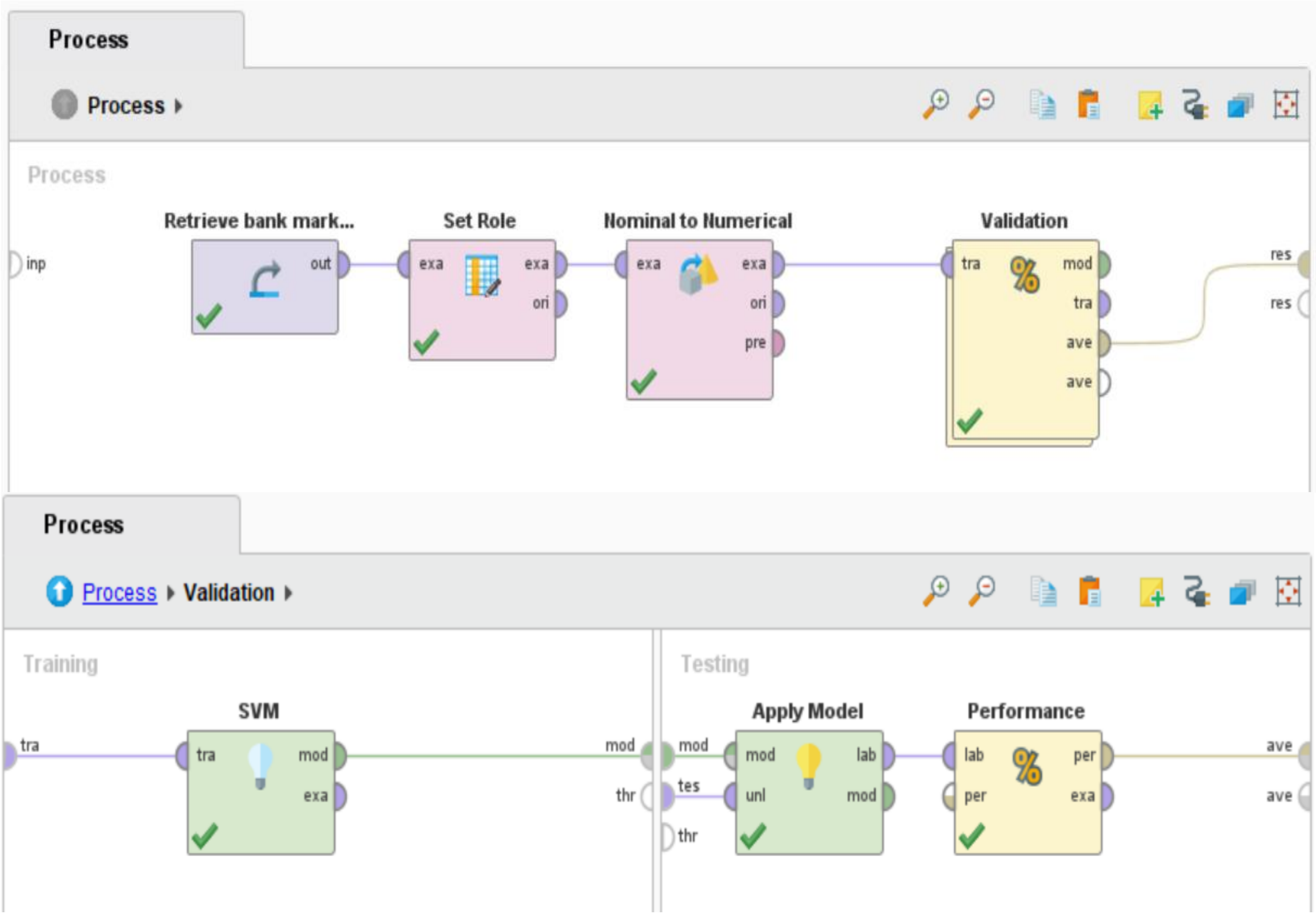
	true no	true yes	class precision
pred. no	39191	4257	90.20%
pred. yes	731	1032	58.54%
class recall	98.17%	19.51%	

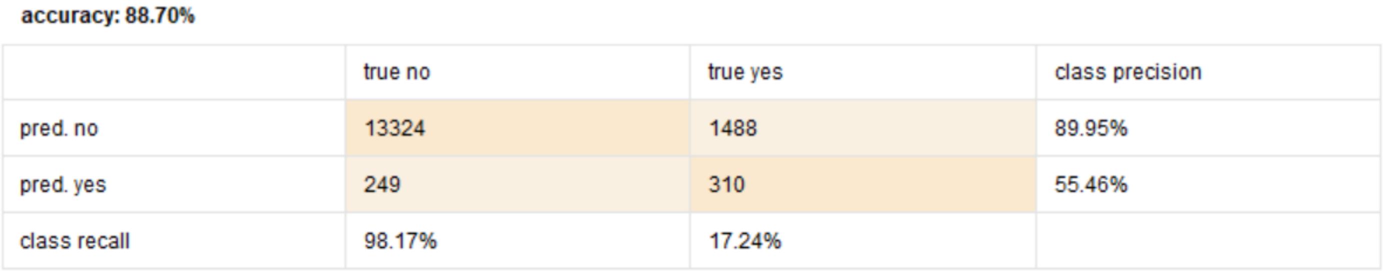


PerformanceVector

```
PerformanceVector:
accuracy: 88.97% +/- 0.42% (micro average: 88.97%)
ConfusionMatrix:
True:   no      yes
no:     39191   4257
yes:    731     1032
```

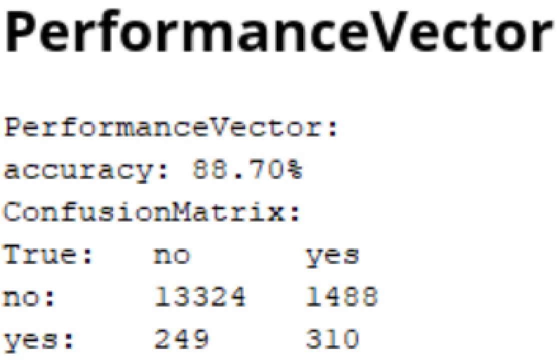
6.2.4 RapidMiner Split Validation





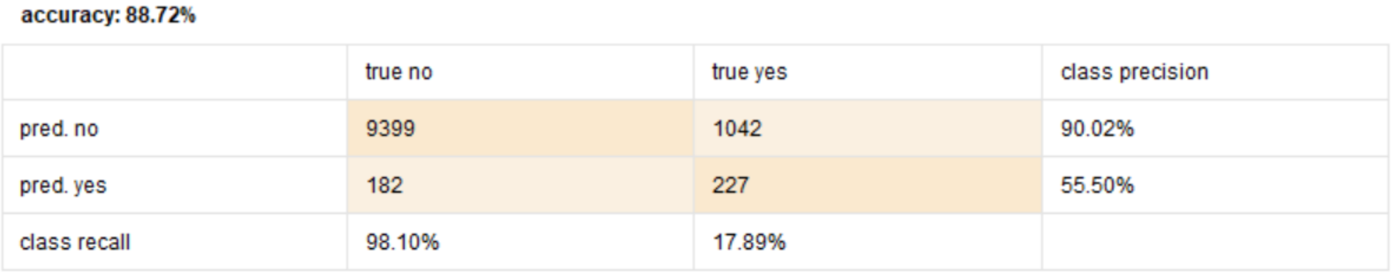
accuracy: 88.70%

	true no	true yes	class precision
pred. no	13324	1488	89.95%
pred. yes	249	310	55.46%
class recall	98.17%	17.24%	



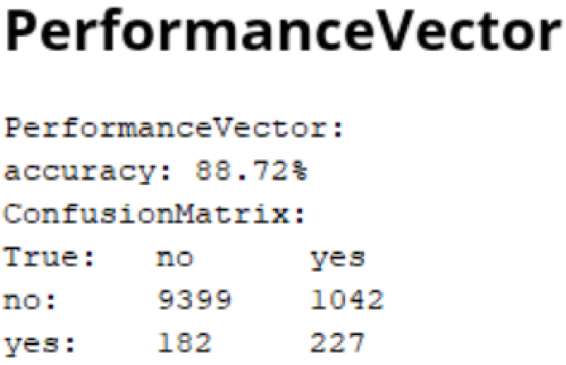
PerformanceVector

```
PerformanceVector:
accuracy: 88.70%
ConfusionMatrix:
True:   no      yes
no:     13324   1488
yes:    249     310
```



accuracy: 88.72%

	true no	true yes	class precision
pred. no	9399	1042	90.02%
pred. yes	182	227	55.50%
class recall	98.10%	17.89%	



PerformanceVector

```
PerformanceVector:
accuracy: 88.72%
ConfusionMatrix:
True:   no      yes
no:     9399    1042
yes:    182     227
```