Department of Computer Science

CPCS331- Artificial intelligence

Thu 15rd April – 2021

Machine Learning Project

Travel Insurance dataset

Team members

Razan Aljuhani, Haya Alsheikh, Wejdan Alzahrani, Lama Khaled

Table of Contents

Task Assignment *1*

1. *Introduction* *2*
   - *1.1 Purpose of the project* *2*
   - *1.2 Learning objectives* *2*
   - *1.3 Summary of report* *2*
2. *Technical description* *3*
   - *2.1 Dataset* *3*
   - *2.2 Machine Learning algorithms* *4*
3. *Results* *5*
   - *3.1 Experment results* *5*
     - *3.1.1 Logistic regression algorithm* *5*
     - *3.1.2 Decision tree algorithm* *8*
     - *3.1.3 Naïve bayes algorithm* *11*
     - *3.1.4 K-nearest algorithm* *14*
   - *3.2 Result analysis* *18*
4. *Conculosion* *20*
5. *Reference* *20*

Task Assignment

Member Name	*Tasks Performed*
Razan Aljuhani	• Apply Logistic regression algorithm on the dataset. • Writing Report.
Haya Alsheikh	• Apply Decision tree algorithm on the dataset. • Writing Report.
Wejdan Alzahrani	• Apply Naïve bayes algorithm on the dataset. • Writing Report.
Lama Khaled	• Apply K-nearest algorithm on the dataset. • Writing Report.

1. Introduction

Machine Learning is one of the most needed useful methods in data science, Machine Learning is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions. The process of learning begins with observations or data. Everything around us is data, the data can be classified and organized into sets to observe growth, predictions, and to perform experiments on.

1.1 Purpose of project

In this project, we chosen a dataset we retrieved from the given online website. We want to check if based on the stored data of a travel insurance agency, does the individual claim their insurance or not. On Naïve Bayes algorithm, Decision Tree algorithm, and Logistic Regression algorithm and k-Nearest algorithm by running Algorithm in RapidMiner Studio and Weka.

1.2 Learning objectives

- Learn and be able to use AI data programs.
- Test specific algorithms on a dataset using both RapidMiner and Weka.
- Compares the accuracy between algorithms technique.
- Observe how different algorithms handle the data.

1.3 Summary of report

In this report, we chosen some algorithms on the travel insurance agency dataset and applied it by using RapidMiner Studio and Weka this algorithm is Naïve Bayes algorithm, Decision Tree algorithm, and Logistic Regression algorithm, and k-Nearest algorithm. We Performed every algorithm by split validation, and cross validation. At the end of the experiments, we achieved to the accuracy results by analyzed the result.

2. Technical Description

2.1 Dataset

We have chosen the Travel Insurance dataset, a travel insurance servicing company based in Singapore created the dataset we will be experimenting on. The data evaluates whether the insurance for the individual is claimed or not. There exist 63,326 instances, 10 attributes, and 1 target attribute.

Attributes information :

Attribute	Description	Type	Values
Agency	Name of agency	Nominal	'EPX', 'CWT', 'ART', 'CBH', and 19627 more
Agency. Type	Type of travel insurance agency	Nominal	'Travel Agency', 'Airlines'
Distribution. Channel	Distribution channel of agency	Nominal	'Online', 'Offline'
Product. Name	Name of products	Nominal	'Cancellation Plan', 'Rental Vehicle…', and 31538 more
Duration	Duration of travel	Numeric	-2 <= 4881
Destination	Destination of travel	Nominal	Many countries included.
Net. Sales	Number of sales of travel insurance policies	Numeric	-389 <= 810
Commission	Commission received for agency	Numeric	0 <= 284
Gender	Gender of insured	Nominal	'F', 'M', null
Age	Age of insured	Numeric	0 <= 118
Claim. Status	Claim status (target)	Nominal	'Yes', 'No'

Row No.	Claim ↑	Agency	Agency Type	Distribution ...	Product Na...	Duration	Destination	Net Sales	Commision (...	Gender	Age
63318	No	JZI	Airlines	Online	Basic Plan	42	AUSTRALIA	22	7.700	F	25
63319	No	JZI	Airlines	Online	Basic Plan	10	CHINA	35	12.250	M	54
63320	No	JZI	Airlines	Online	Basic Plan	10	CHINA	35	12.250	M	51
63321	No	JZI	Airlines	Online	Basic Plan	5	BRUNEI DAR...	18	6.300	M	27
63322	No	JZI	Airlines	Online	Basic Plan	111	JAPAN	35	12.250	M	31
63323	No	JZI	Airlines	Online	Basic Plan	58	CHINA	40	14	F	40
63324	No	JZI	Airlines	Online	Basic Plan	2	MALAYSIA	18	6.300	M	57
63325	No	JZI	Airlines	Online	Basic Plan	3	VIET NAM	18	6.300	M	63
63326	No	JZI	Airlines	Online	Basic Plan	22	HONG KONG	26	9.100	F	35
24	Yes	C2B	Airlines	Online	Bronze Plan	12	SINGAPORE	94	23.500	M	34
249	Yes	C2B	Airlines	Online	Silver Plan	10	SINGAPORE	43.550	10.890	M	45
314	Yes	EPX	Travel Agency	Online	Cancellation ...	73	THAILAND	16	0	?	36
420	Yes	C2B	Airlines	Online	Silver Plan	11	SINGAPORE	62.250	15.560	M	33
425	Yes	C2B	Airlines	Online	Annual Silver ...	365	SINGAPORE	187.850	46.960	M	32
440	Yes	CWT	Travel Agency	Online	Rental Vehicl...	105	UNITED KIN...	39.600	23.760	?	32
463	Yes	EPX	Travel Agency	Online	2 way Compr...	9	CHINA	87	0	?	36
602	Yes	C2B	Airlines	Online	Silver Plan	16	SINGAPORE	74.250	18.560	M	27
637	Yes	EPX	Travel Agency	Online	2 way Compr...	56	GERMANY	145	0	?	36
781	Yes	C2B	Airlines	Online	Annual Silver ...	364	SINGAPORE	252.850	63.210	M	30
967	Yes	EPX	Travel Agency	Online	2 way Compr...	51	PHILIPPINES	20	0	?	36
1113	Yes	EPX	Travel Agency	Online	2 way Compr...	24	LAO PEOPLE...	21	0	?	36
1781	Yes	C2B	Airlines	Online	Annual Silver ...	431	SINGAPORE	272.300	68.080	F	34
2192	Yes	SSI	Airlines	Online	Ticket Protector	105	SINGAPORE	3.680	1.030	?	48
2213	Yes	C2B	Airlines	Online	Silver Plan	10	SINGAPORE	50	12.500	F	50
2364	Yes	C2B	Airlines	Online	Annual Silver ...	365	SINGAPORE	252.850	63.210	M	61

ExampleSet (63,326 examples, 1 special attribute, 10 regular attributes)

2.2 Machine Learning algorithms

The following is the description about the selected four algorithms :-

1- Logistic regression Algorithm

It's one of the most popular machine learning algorithms, which is used for the classification problems. It's predictive analysis algorithm and based on the concept of probability. It's much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

2- Decision Tree Algorithm

It's belongs to the family of machine learning algorithms. it's highly effective in decision making. its aim to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data. It's able to handle both categorical and numerical data and is resistant to outliers, which means it requires a bit of data preprocessing.

3- Naïve Bayes Algorithm

It's known for being one of the simplest effective classification algorithms which based on Bayes' Theorem with an assumption of independence among predictors. It's performs better in machine learning models for quick fast predictions. Also, it doesn't require much training data and is not sensitive to irrelevant features.

4- K-Nearest algorithm

It's simple algorithm, easy-to-implement machine learning algorithm that can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry.

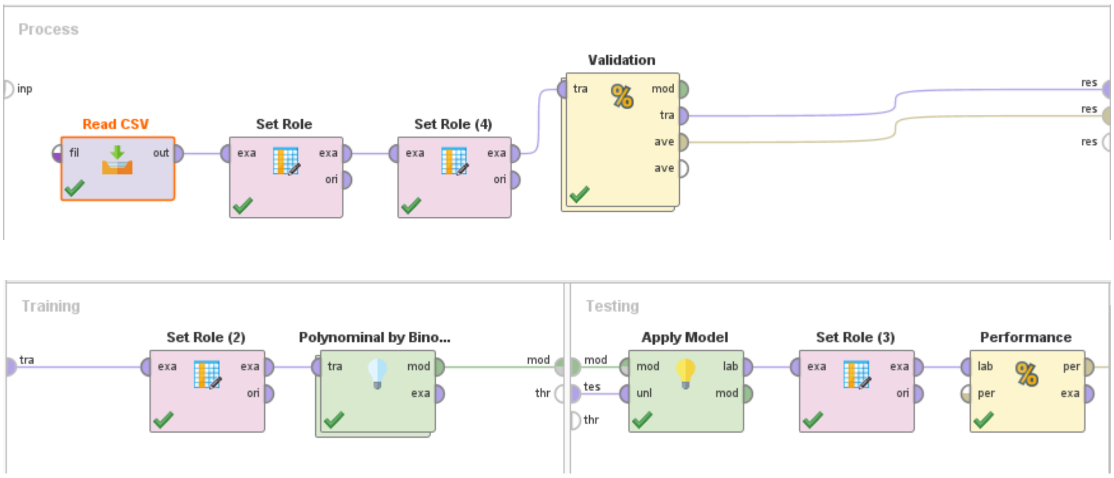
3. Results

3.1 Experiment Result

3.1.1 Logistic Regression Algorithm

- (Split Validation)

In RapidMiner Studio:



(Performance)

accuracy: 98.53%

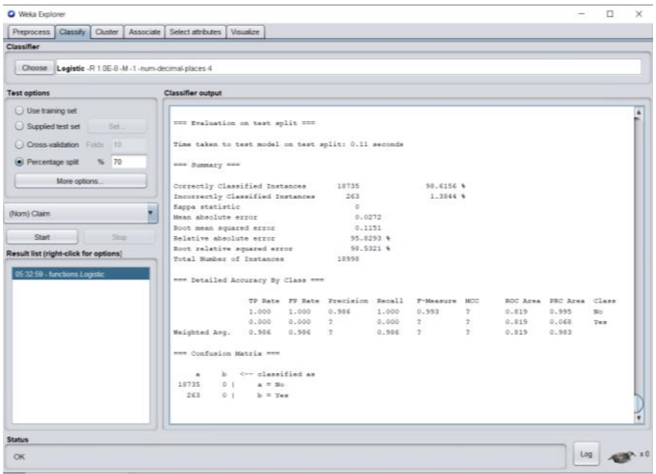
	true No	true Yes	class precision
pred. No	18719	278	98.54%
pred. Yes	1	0	0.00%
class recall	99.99%	0.00%	

(Description)

PerformanceVector

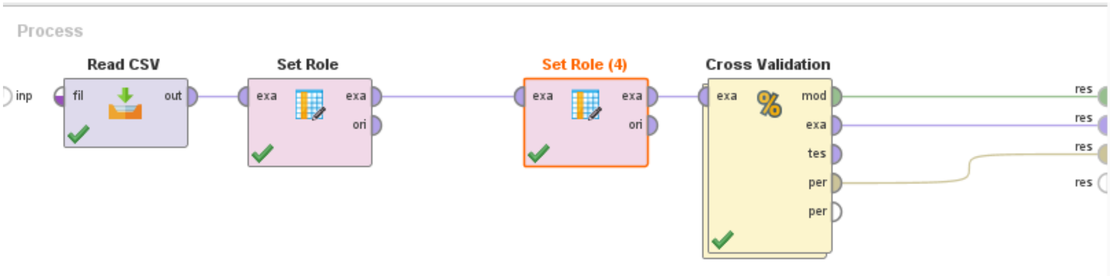
```
PerformanceVector:
accuracy: 98.53%
ConfusionMatrix:
True:   No      Yes
No:     18719   278
Yes:    1       0
precision: 0.00% (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     18719   278
Yes:    1       0
recall: 0.00% (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     18719   278
Yes:    1       0
AUC (optimistic): 0.794 (positive class: Yes)
AUC: 0.794 (positive class: Yes)
AUC (pessimistic): 0.794 (positive class: Yes)
```

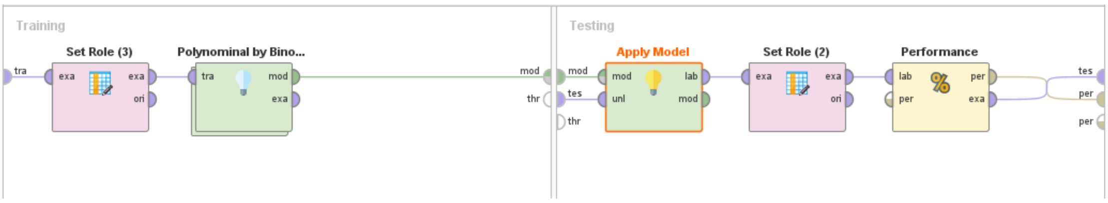
In Weka:



(Cross Validation)

In RapidMiner Studio:





(Performance)

accuracy: 98.53% +/- 0.01% (micro average: 98.53%)

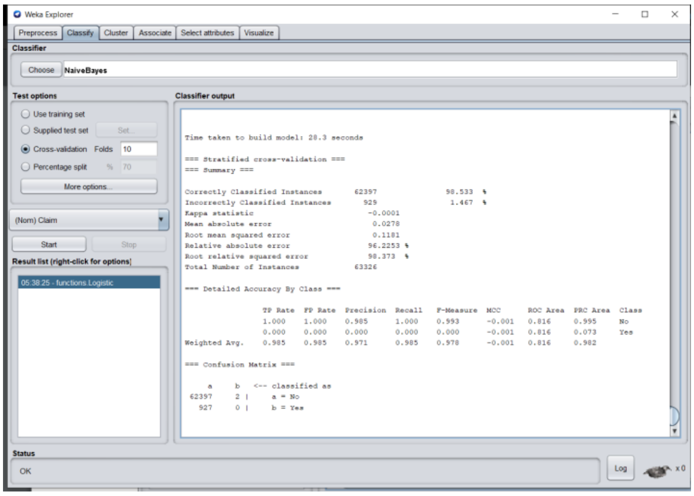
	true No	true Yes	class precision
pred. No	62398	927	98.54%
pred. Yes	1	0	0.00%
class recall	100.00%	0.00%	

(Description)

PerformanceVector

```
PerformanceVector:
accuracy: 98.53% +/- 0.01% (micro average: 98.53%)
ConfusionMatrix:
True:   No      Yes
No:     62398   927
Yes:    1       0
precision: 0.00% (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     62398   927
Yes:    1       0
recall: 0.00% +/- 0.00% (micro average: 0.00%) (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     62398   927
Yes:    1       0
AUC (optimistic): 0.816 +/- 0.015 (micro average: 0.816) (positive class: Yes)
AUC: 0.816 +/- 0.015 (micro average: 0.816) (positive class: Yes)
AUC (pessimistic): 0.816 +/- 0.015 (micro average: 0.816) (positive class: Yes)
```

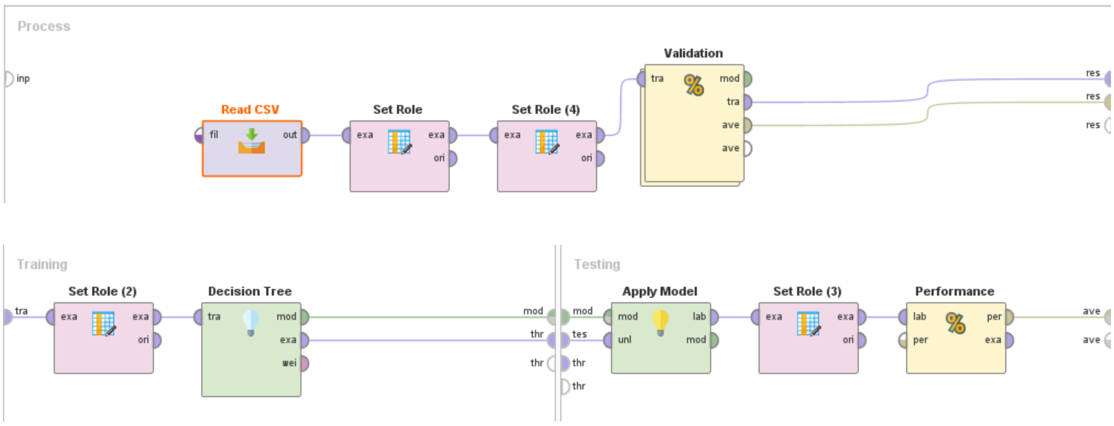
In Weka:



3.1.2 Decision tree algorithm

(Split Validation)

In RapidMiner Studio:



(Performance)

accuracy: 98.53%

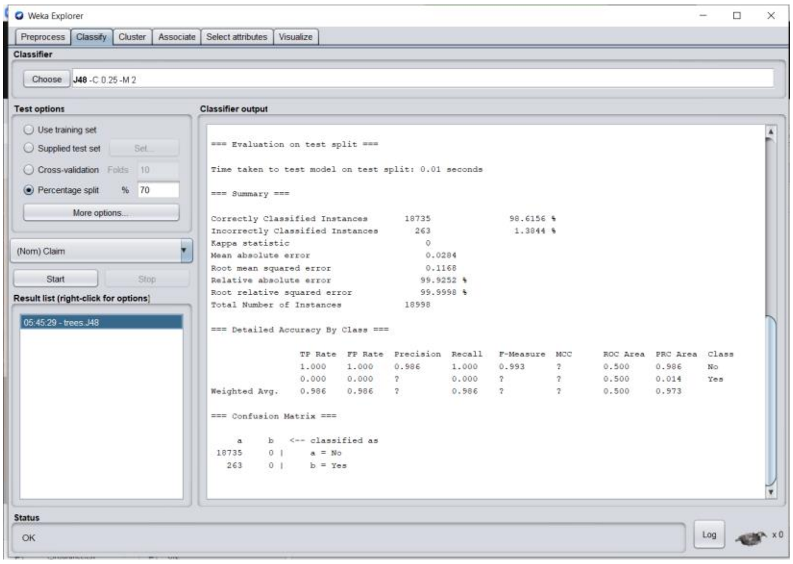
	true No	true Yes	class precision
pred. No	18719	278	98.54%
pred. Yes	1	0	0.00%
class recall	99.99%	0.00%	

(Description)

PerformanceVector

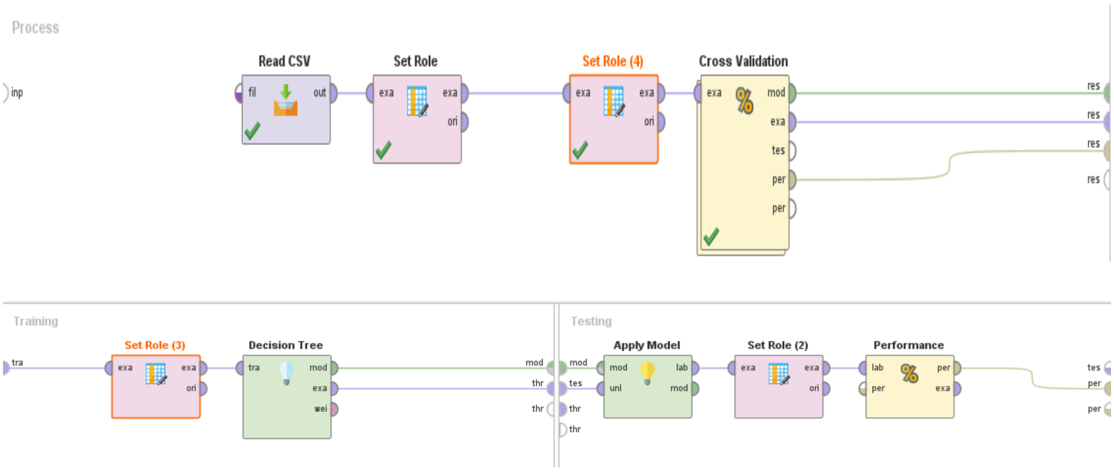
```
PerformanceVector:
accuracy: 98.53%
ConfusionMatrix:
True:   No      Yes
No:     18719   278
Yes:    1       0
precision: 0.00% (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     18719   278
Yes:    1       0
recall: 0.00% (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     18719   278
Yes:    1       0
AUC (optimistic): 0.989 (positive class: Yes)
AUC: 0.541 (positive class: Yes)
AUC (pessimistic): 0.093 (positive class: Yes)
```

In Weka:



(Cross Validation)

In RapidMiner Studio:



(Performance)

accuracy: 98.52% +/- 0.03% (micro average: 98.52%)

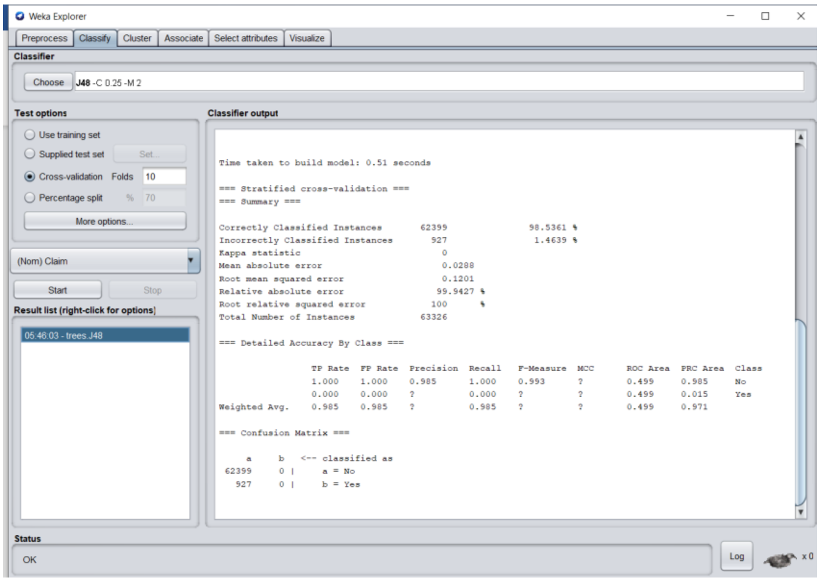
	true No	true Yes	class precision
pred. No	62389	926	98.54%
pred. Yes	10	1	9.09%
class recall	99.98%	0.11%	

(Description)

PerformanceVector

```
PerformanceVector:
accuracy: 98.52% +/- 0.03% (micro average: 98.52%)
ConfusionMatrix:
True:   No      Yes
No:     62389   926
Yes:    10      1
precision: 9.09% (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     62389   926
Yes:    10      1
recall: 0.11% +/- 0.34% (micro average: 0.11%) (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     62389   926
Yes:    10      1
AUC (optimistic): 0.982 +/- 0.010 (micro average: 0.982) (positive class: Yes)
AUC: 0.561 +/- 0.015 (micro average: 0.561) (positive class: Yes)
AUC (pessimistic): 0.139 +/- 0.037 (micro average: 0.139) (positive class: Yes)
```

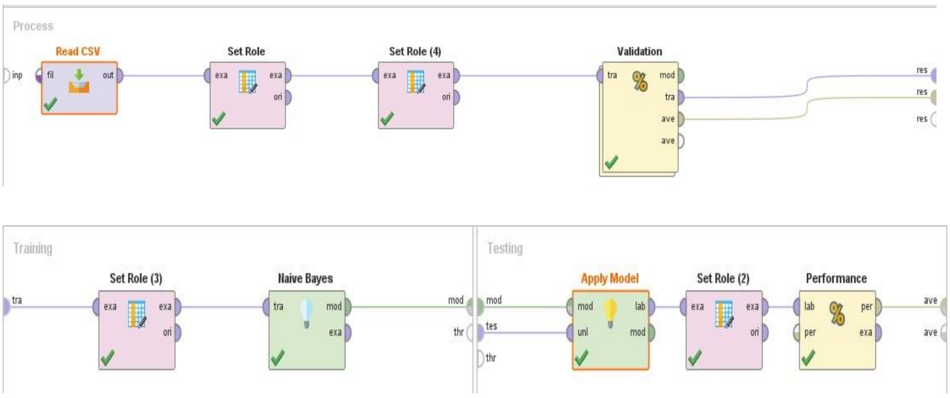
In Weka:



3.1.3 Naïve Bayes Algorithm

(Split Validation)

In RapidMiner Studio:

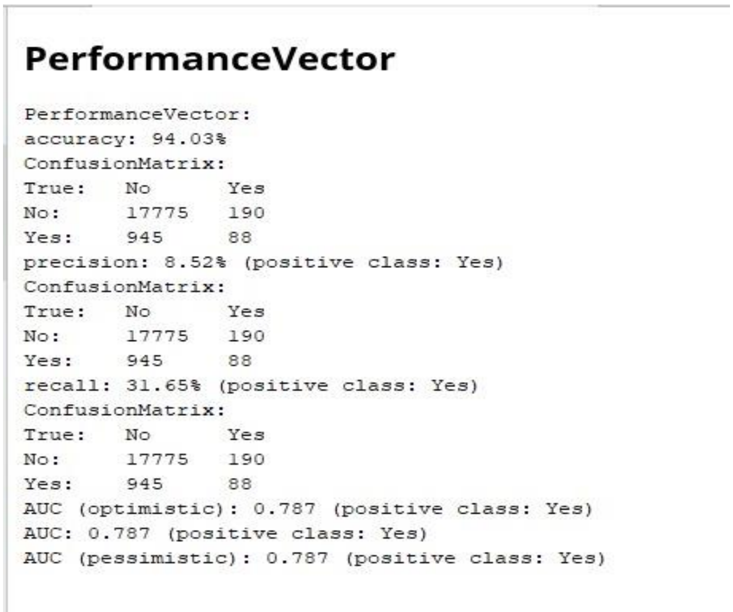


(Performance)

accuracy: 94.03%

	true No	true Yes	class precision
pred. No	17775	190	98.94%
pred. Yes	945	88	8.52%
class recall	94.95%	31.65%	

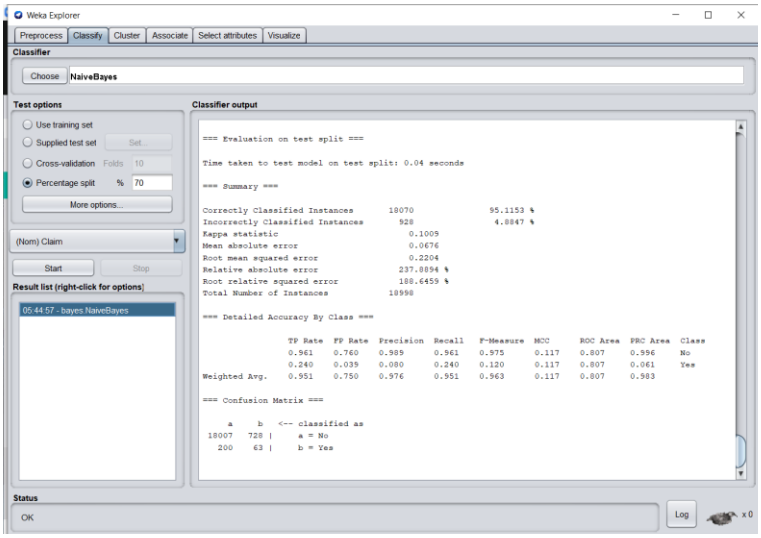
(Description)



PerformanceVector

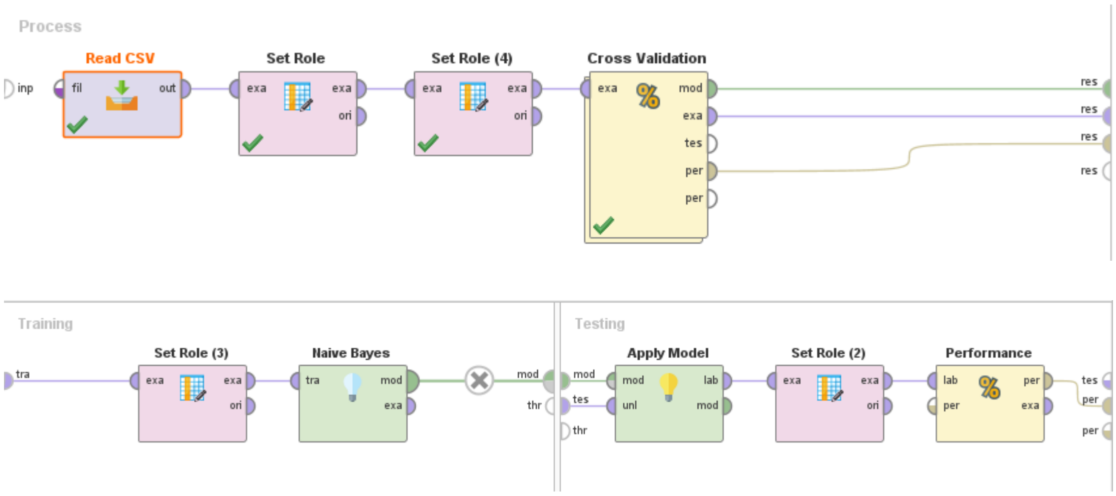
```
PerformanceVector:
accuracy: 94.03%
ConfusionMatrix:
True:   No      Yes
No:     17775   190
Yes:    945     88
precision: 8.52% (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     17775   190
Yes:    945     88
recall: 31.65% (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     17775   190
Yes:    945     88
AUC (optimistic): 0.787 (positive class: Yes)
AUC: 0.787 (positive class: Yes)
AUC (pessimistic): 0.787 (positive class: Yes)
```

In Weka:



(Cross Validation)

In RapidMiner Studio:



(Performance)

accuracy: 93.78% +/- 0.42% (micro average: 93.78%)

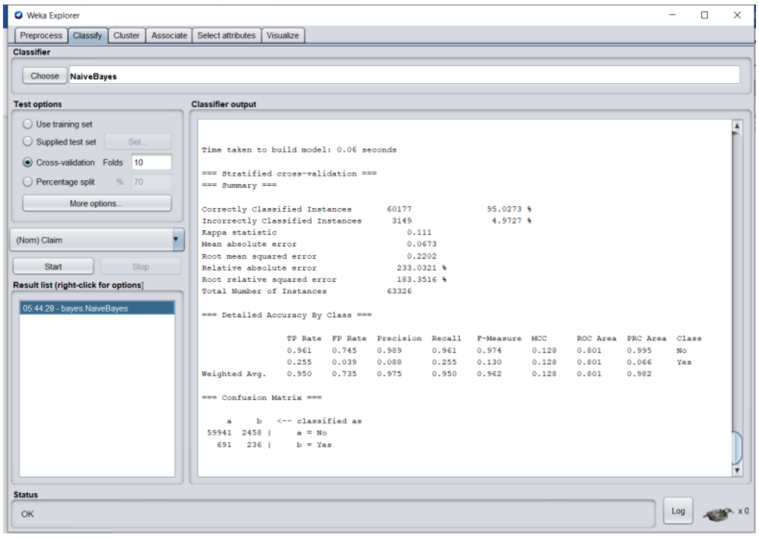
	true No	true Yes	class precision
pred. No	59075	616	98.97%
pred. Yes	3324	311	8.56%
class recall	94.67%	33.55%	

(Description)

PerformanceVector

```
PerformanceVector:
accuracy: 93.78% +/- 0.42% (micro average: 93.78%)
ConfusionMatrix:
True:   No      Yes
No:     59075   616
Yes:    3324    311
precision: 8.59% +/- 1.05% (micro average: 8.56%) (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     59075   616
Yes:    3324    311
recall: 33.54% +/- 3.65% (micro average: 33.55%) (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     59075   616
Yes:    3324    311
AUC (optimistic): 0.794 +/- 0.021 (micro average: 0.794) (positive class: Yes)
AUC: 0.794 +/- 0.021 (micro average: 0.794) (positive class: Yes)
AUC (pessimistic): 0.794 +/- 0.021 (micro average: 0.794) (positive class: Yes)
```

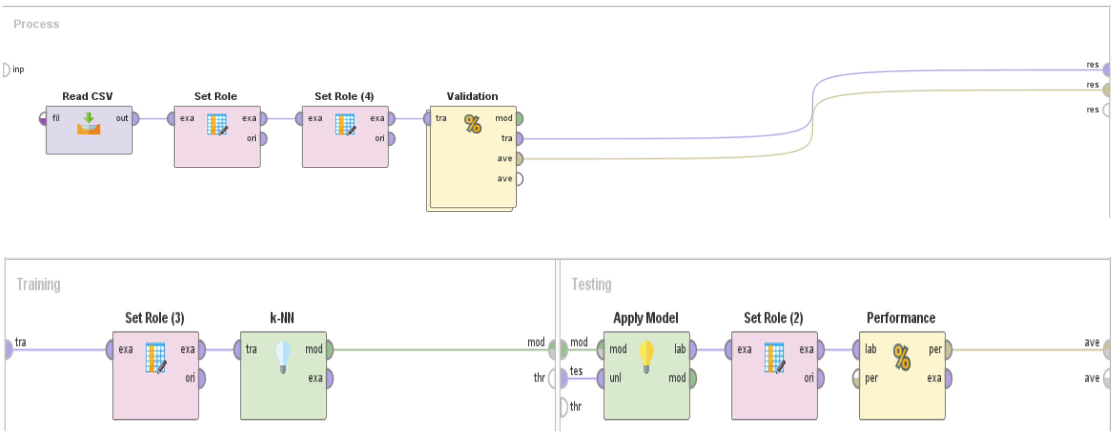
In Weka:



3.1.4 K-nearest Algorithm

(Split Validation)

In RapidMiner Studio:



(Performance)

accuracy: 98.49%

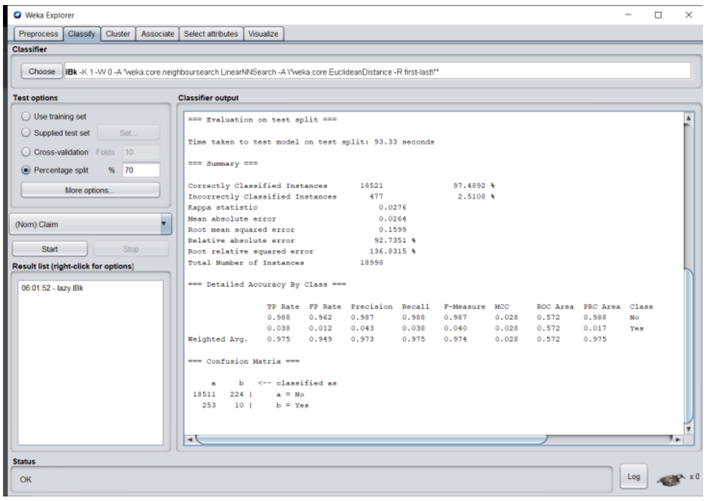
	true No	true Yes	class precision
pred. No	18710	276	98.55%
pred. Yes	10	2	16.67%
class recall	99.95%	0.72%	

(Description)

PerformanceVector

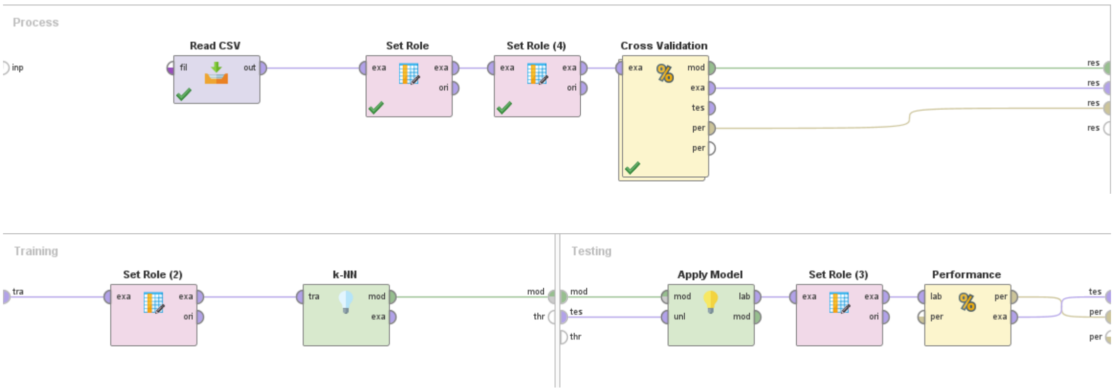
```
PerformanceVector:
accuracy: 98.49%
ConfusionMatrix:
True:   No      Yes
No:     18710   276
Yes:    10      2
precision: 16.67% (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     18710   276
Yes:    10      2
recall: 0.72% (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     18710   276
Yes:    10      2
AUC (optimistic): 0.947 (positive class: Yes)
AUC: 0.596 (positive class: Yes)
AUC (pessimistic): 0.245 (positive class: Yes)
```

In Weka:



(Cross Validation)

In RapidMiner Studio:



(Performance)

accuracy: 98.48% +/- 0.02% (micro average: 98.48%)

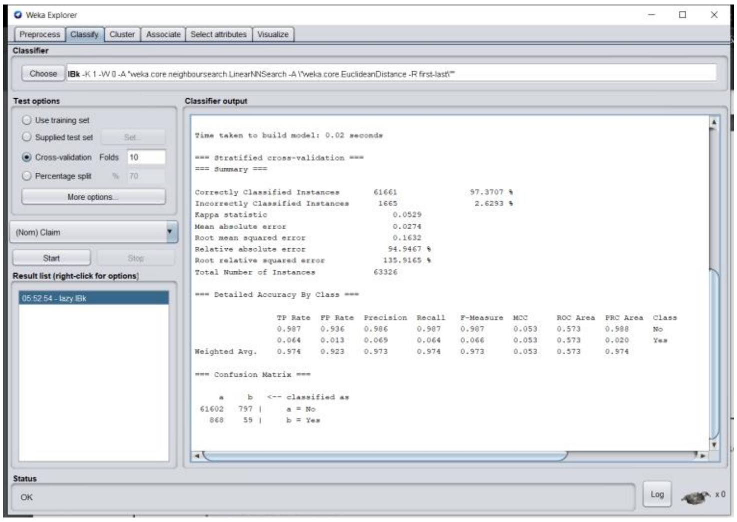
	true No	true Yes	class precision
pred. No	62359	920	98.55%
pred. Yes	40	7	14.89%
class recall	99.94%	0.76%	

(Description)

PerformanceVector

```
PerformanceVector:
accuracy: 98.48% +/- 0.02% (micro average: 98.48%)
ConfusionMatrix:
True:   No      Yes
No:     62359   920
Yes:    40      7
precision: 15.97% +/- 16.80% (micro average: 14.89%) (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     62359   920
Yes:    40      7
recall: 0.76% +/- 0.73% (micro average: 0.76%) (positive class: Yes)
ConfusionMatrix:
True:   No      Yes
No:     62359   920
Yes:    40      7
AUC (optimistic): 0.946 +/- 0.003 (micro average: 0.946) (positive class: Yes)
AUC: 0.594 +/- 0.016 (micro average: 0.594) (positive class: Yes)
AUC (pessimistic): 0.242 +/- 0.031 (micro average: 0.242) (positive class: Yes)
```

In Weka:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 70
More options...
(Nom) Claim
Start
Stop
Result list (right-click for options)
05:52:54 - lazy.IBk
Classifier output
Time taken to build model: 0.02 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 61661 97.3707 %
Incorrectly Classified Instances 1665 2.6293 %
Kappa statistic 0.0529
Mean absolute error 0.0274
Root mean squared error 0.1632
Relative absolute error 94.9467 %
Root relative squared error 135.9165 %
Total Number of Instances 63326
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.987 0.936 0.986 0.987 0.987 0.053 0.573 0.988 No
0.064 0.013 0.069 0.064 0.066 0.053 0.573 0.020 Yes
Weighted Avg. 0.974 0.923 0.973 0.974 0.973 0.053 0.573 0.974
=== Confusion Matrix ===
a b <-- classified as
61602 797 | a = No
868 59 | b = Yes
Status
OK
Log
x 0

3.2 Result analysis

Split Validation

Split Percentage = 70%			
Program	Algorithm	Accuracy	Confusion Matrix
RapidMiner Studio	Logistic regression	98.53%	True: NO Yes No: 18719 278 Yes: 1 0
	Decision Tree Algorithm	98.53%	True: NO Yes No: 18719 278 Yes: 1 0
	Naïve Bayes Algorithm	94.03%	True: NO Yes No: 17775 190 Yes: 945 88
	K-Nearest algorithm	98.49%	True: NO Yes No: 18710 276 Yes: 10 2
Weka	Logistic regression	98.61%	a b <-- classified 18735 0| a = No 263 0| b = Yes
	Decision Tree Algorithm	98.61%	a b <-- classified 18735 0| a = No 263 0| b = Yes
	Naïve Bayes Algorithm	95.11%	a b <-- classified 18007 728 | a = No 200 63 | b = Yes
	K-Nearest algorithm	97.48%	a b <-- classified 18511 224 | a = No 253 10 | b = Yes

After applying the algorithms by split validation on the two software, we notice different results despite the same dataset. In RapidMiner software, the accuracy is the same for decision tree, logistic regression, a slight difference for k-nearest and Naïve bayes has the lowest accuracy out of the four algorithms. Also, in Weka software all algorithms have high percentages above 90% and the results higher than the other software. So, we conclude that for the split validation Weka generated better and more accurate results.

Cross Validation

Number of Folds = 10			
Program	Algorithm	Accuracy	Confusion Matrix
RapidMiner Studio	Logistic regression	98.53%	True: NO Yes No: 62389 926 Yes: 10 0
	Decision Tree	98.52%	True: NO Yes No: 62389 926 Yes: 10 0
	Naïve Bayes	93.78%	True: NO Yes No: 59075 616 Yes: 3324 311
	K-Nearest	98.48%	True: NO Yes No: 62359 920 Yes: 40 7
Weka	Logistic regression	98.53%	a b <-- classified 62397 2 | a = No 927 0 | b = Yes
	Decision Tree	98.53%	a b <-- classified 62399 0| a = No 927 0 | b = Yes
	Naïve Bayes	95.02%	a b <-- classified 59941 2458 | a = No 691 236 | b = Yes
	K-Nearest	97.37%	a b <-- classified 61602 797 | a = No 868 59 | b = Yes

Here we applying the four algorithms by cross validation on the two software, they also delivered different results on the same dataset. they only shows how the implementation of the four algorithms is different on each program. Again, naïve bayes had the lowest accuracy, decision tree, logistic regression and k-nearest has almost similar results. As weka software the all algorithms produced high percentages of accuracy. Since decision tree and logistic regression has same accuracy results when run on both RapidMiner Studio and Weka. So, we agree that based on naïve bayes results, Weka is the better program for cross validation.

4. Conclusion

In conclusion, the Result analysis showed that we retrieve slightly different results from each program, even when we use the same dataset, algorithm, and type of validation. The four algorithms delivered high accuracy percentages all above 90% with naïve bayes being the lowest but produced slightly better results in weka . Both of the decision tree and logistic regression algorithms produced similar results in both programs and we could not decide which program is better for them, k nearest produced slightly better results in RapidMiner Studio. we learned to use tow machine learning program, RapidMiner Studio and Weka and both of them are good machine learning programs.

5. References

- https://www.kaggle.com/mhdzahier/travel-insurance
- https://www.javatpoint.com/logistic-regression-in-machine-learning