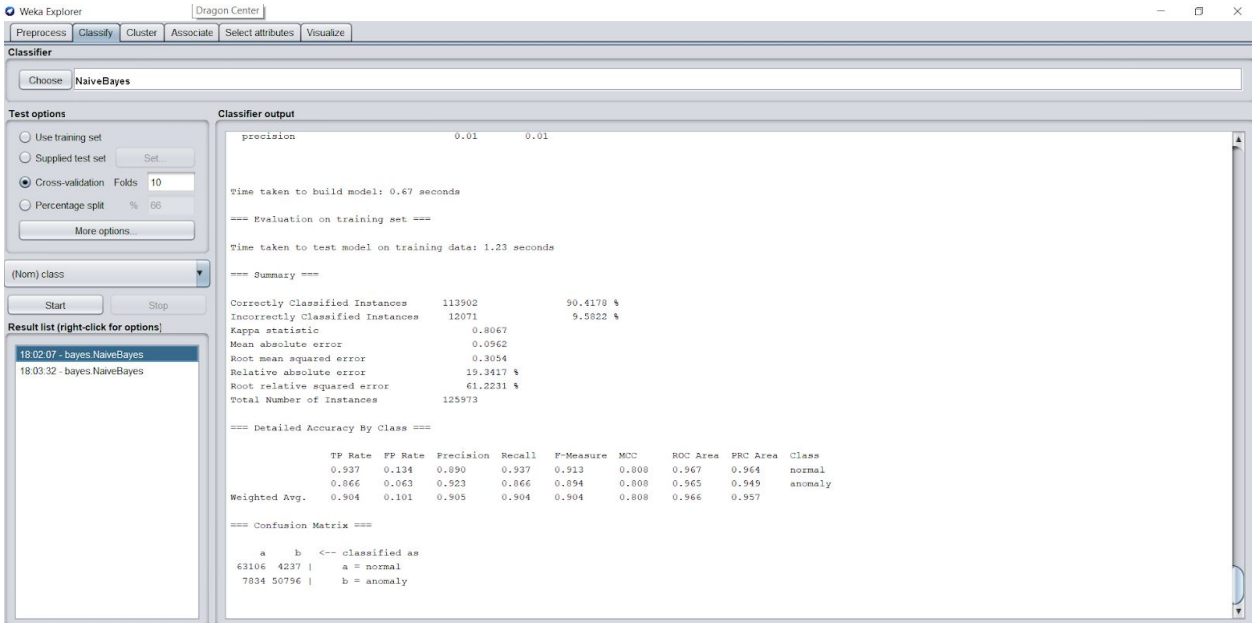
SIT719 Security and Privacy Issues in Analytics

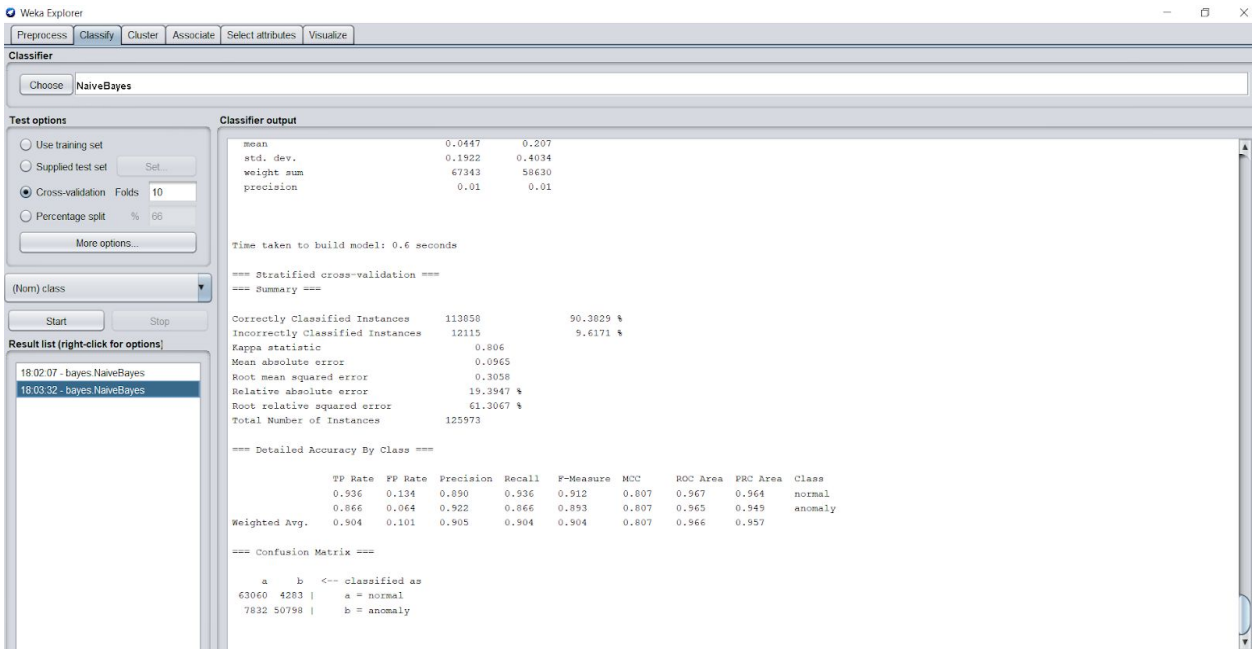
Credit Task 4.2: Intrusion Detection using Supervised Learning Techniques

---

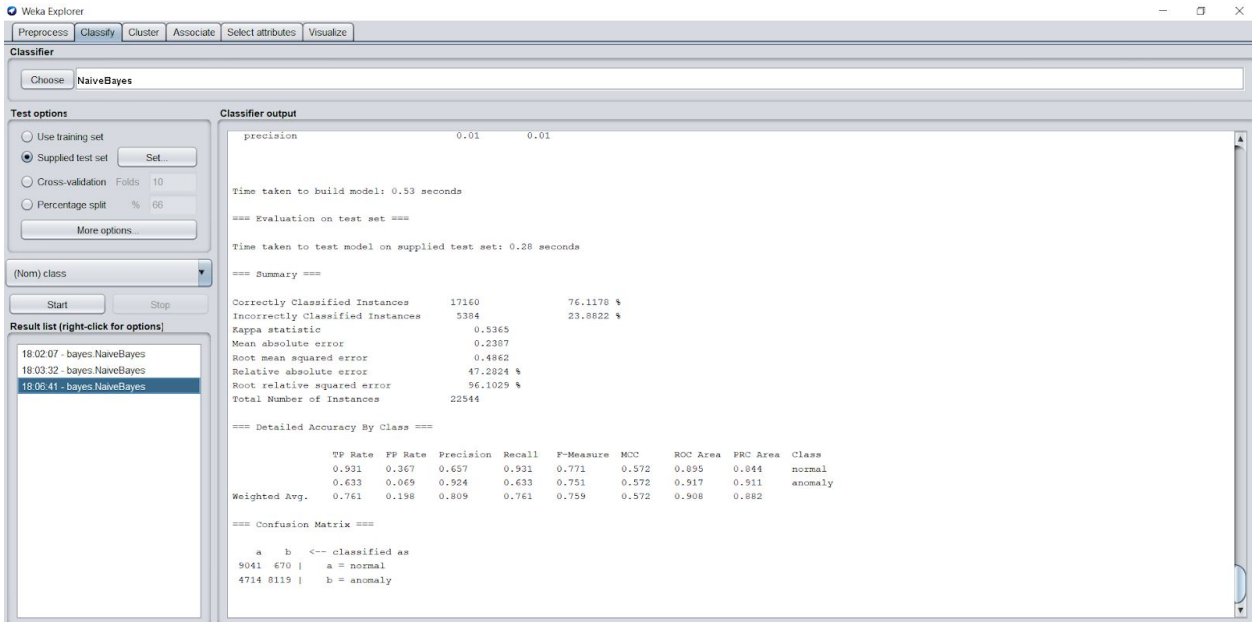
“Naïve Bayes” classification:



10-fold cross-validation:



Using test dataset:



Comparing the results between 10-fold cross-validation and the one obtained using the test dataset. Using the confusion matrix to explain the results:

!. The correctly classified instance of 10 fold cross-validation is higher [90.3%] compared to that of the test data set [76.1%].
2.Confusion matrix of 10 cross-validation
=== Confusion Matrix ===

```
    a     b   <-- classified as
63060  4283 |     a = normal
 7832 50798 |     b = anomaly
```

3. Confusion matrix of the test data set
=== Confusion Matrix ===

```
   a    b   <-- classified as
9041  670 |    a = normal
4714 8119 |    b = anomaly
```

As observed from both of these confusion matrices the TP, TN is considerably higher than FP and FN comparatively between these matrices 2 & 3 which are directly proportionate to the precision and accuracy score.

10 fold cross-validation:

Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.

In 10-fold cross-validation, the original sample is randomly partitioned into 10 equal size subsamples. Of the 10 subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 10-1 subsamples are used as training data. The cross-validation process is then repeated 10 times (the folds), with each of the 10 subsamples used exactly once as the validation data. The 10 results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.

6.)

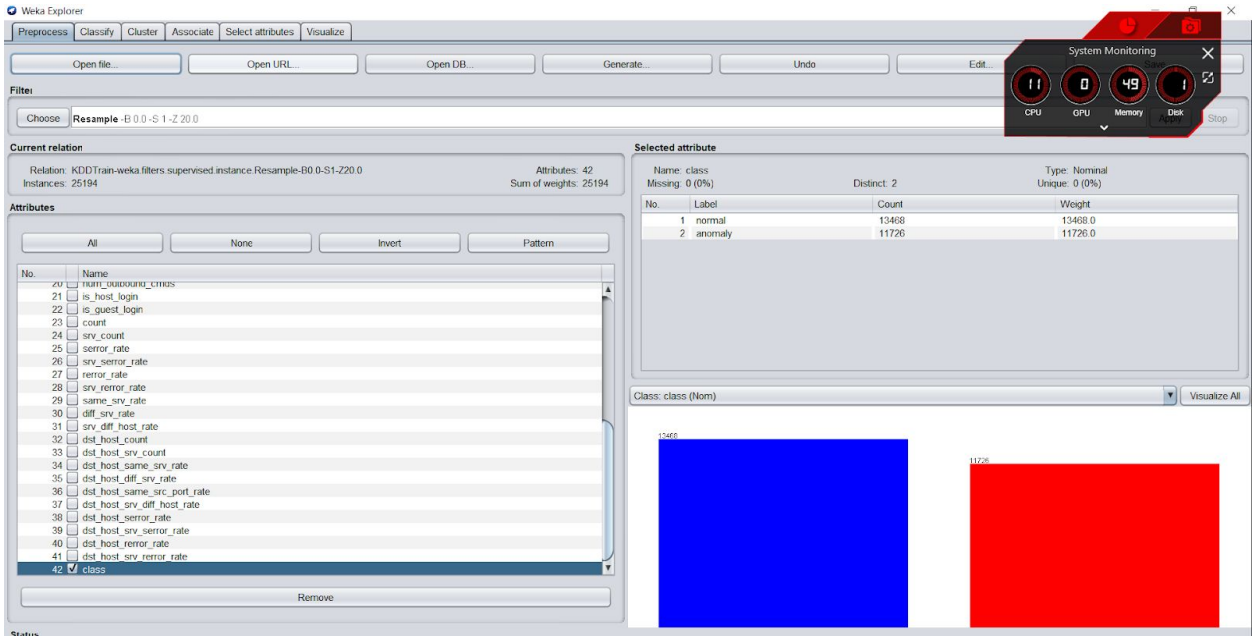
For Train dataset:

Algorithms	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
lazy.IBK	1	1	1	1	1	1
trees.DecisionStump	0.92	0.079	0.922	0.922	0.922	0.922
trees.RandomTree	1.0	1.0	1.0	1.0	1.0	1.0
bayes.BayesNet	0.972	0.031	0.973	0.972	0.972	0.998
rules.OneR	0.964	0.032	0.966	0.964	0.964	0.966

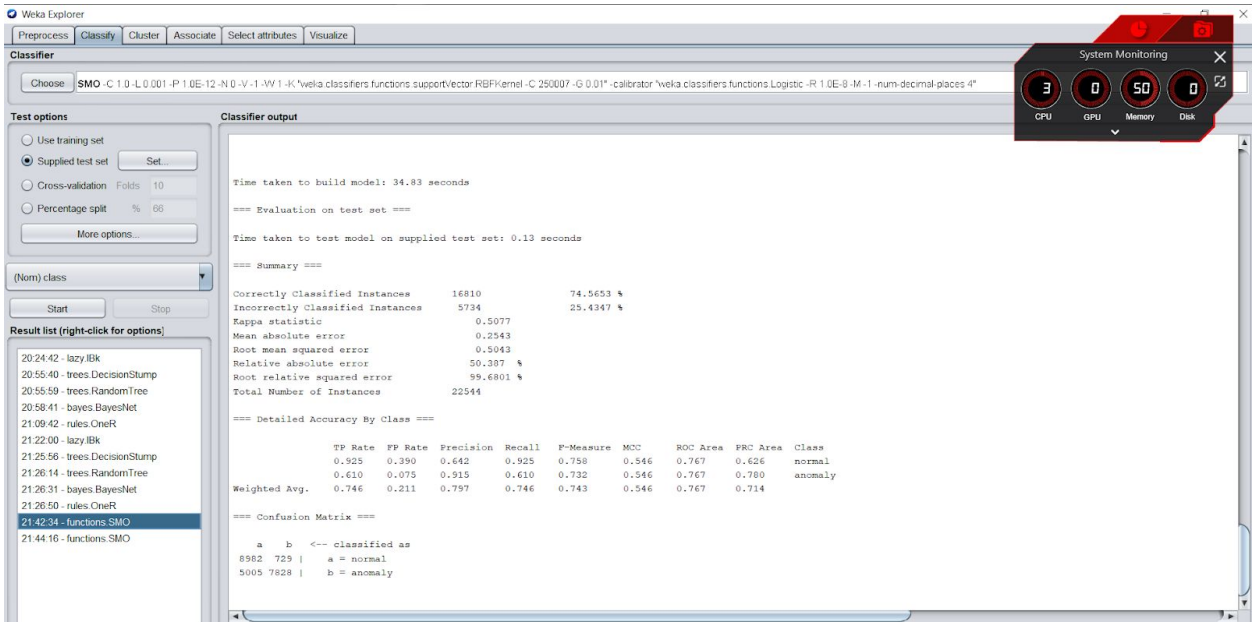
For Test dataset

Algorithms	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
lazy.IBK	0.794	0.165	0.841	0.794	0.792	0.814
trees.DecisionStump	0.800	0.162	0.841	0.800	0.799	0.819
trees.RandomTree	0.814	0.160	0.837	0.814	0.814	0.827
bayes.BayesNet	0.744	0.200	0.822	0.744	0.739	0.945
rules.OneR	0.814	0.151	0.851	0.814	0.814	0.831

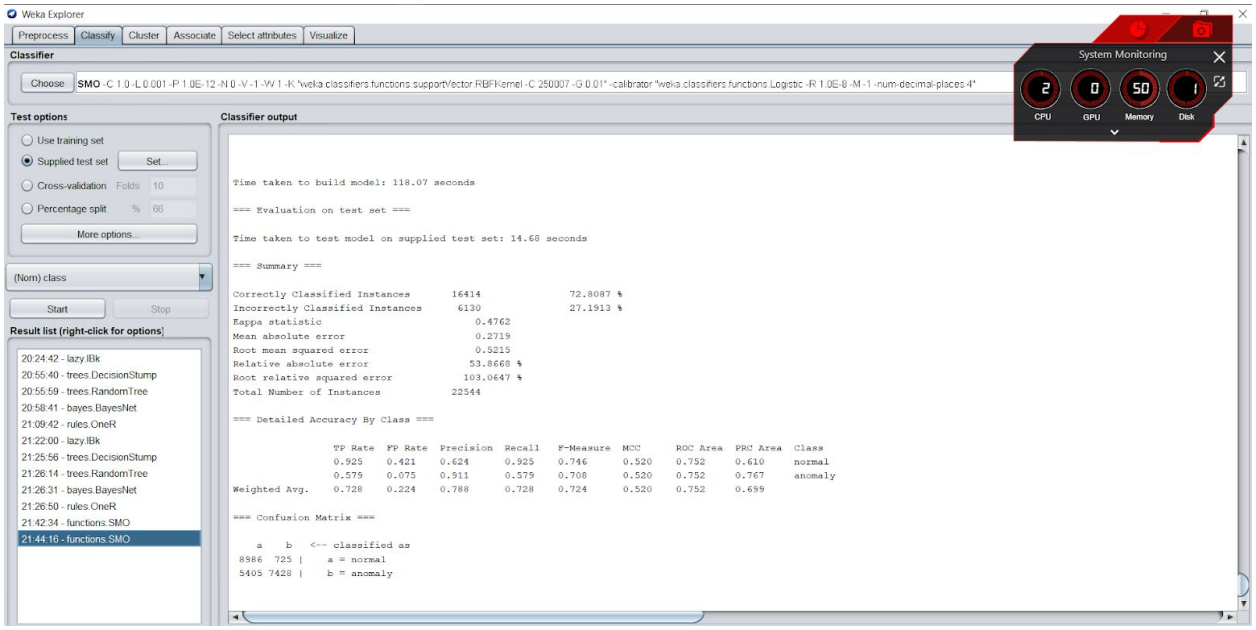
Resample at data size of 20%:



SVM classifier (SMO) using POLY:



SVM classifier (SMO) using RBF:



Confusion matrix and computation time of SVM classifier (SMO) using POLY:
=== Confusion Matrix ===

```
   a    b   <-- classified as
 8982  729 |    a = normal
 5005 7828 |    b = anomaly
```

Time taken to build model: 34.83 seconds
Time taken to test model on supplied test set: 0.13 seconds

Confusion matrix and computation time of SVM classifier (SMO) using POLY:
=== Confusion Matrix ===

```
   a    b   <-- classified as
 8986  725 |    a = normal
 5405 7428 |    b = anomaly
```

Time taken to build model: 118.07 seconds
Time taken to test model on supplied test set: 14.68 seconds

Even though the confusion matrix between the two aren't showing much difference there is surely a notable difference in the computation time taken by both on the same machine.