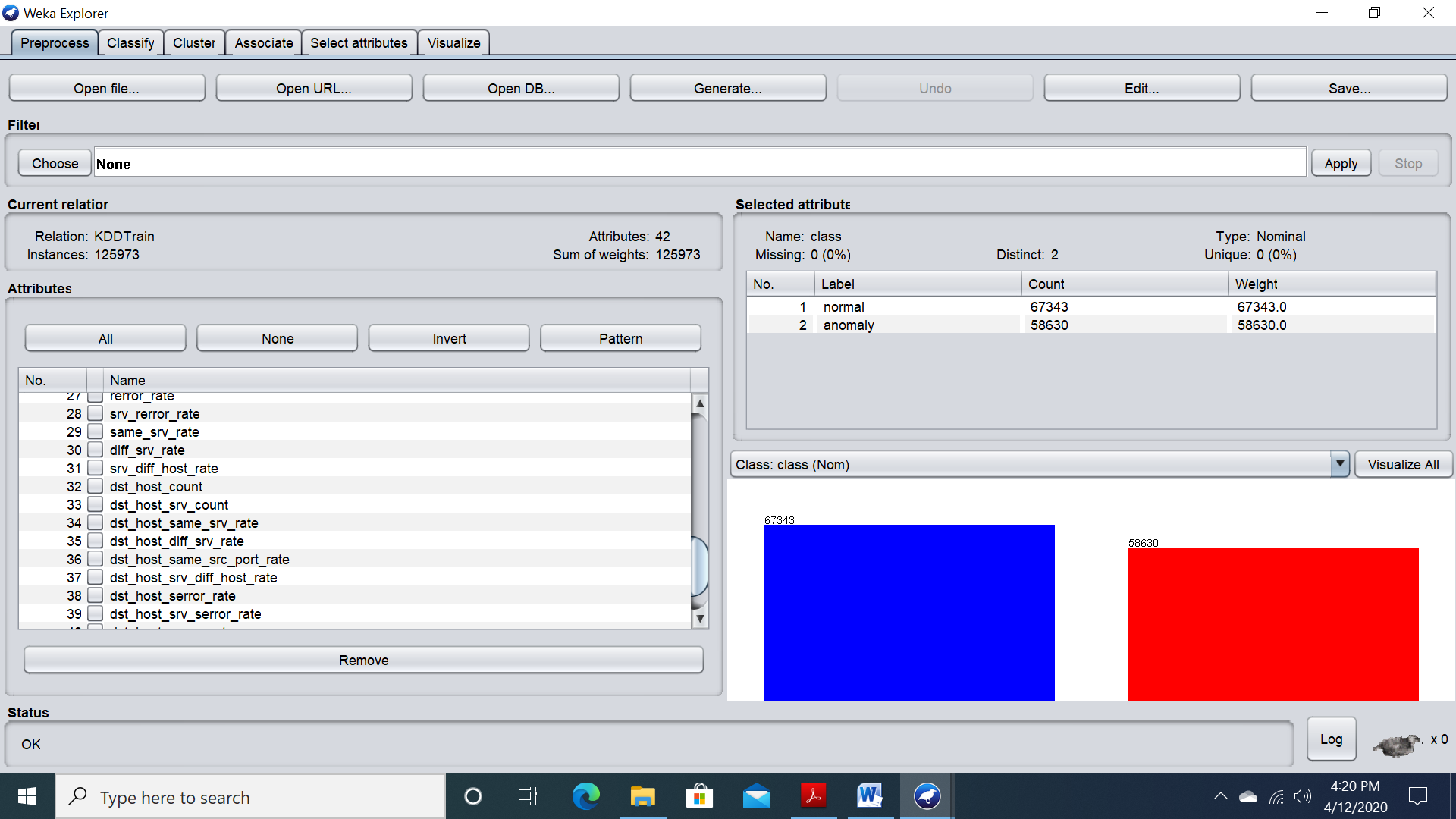
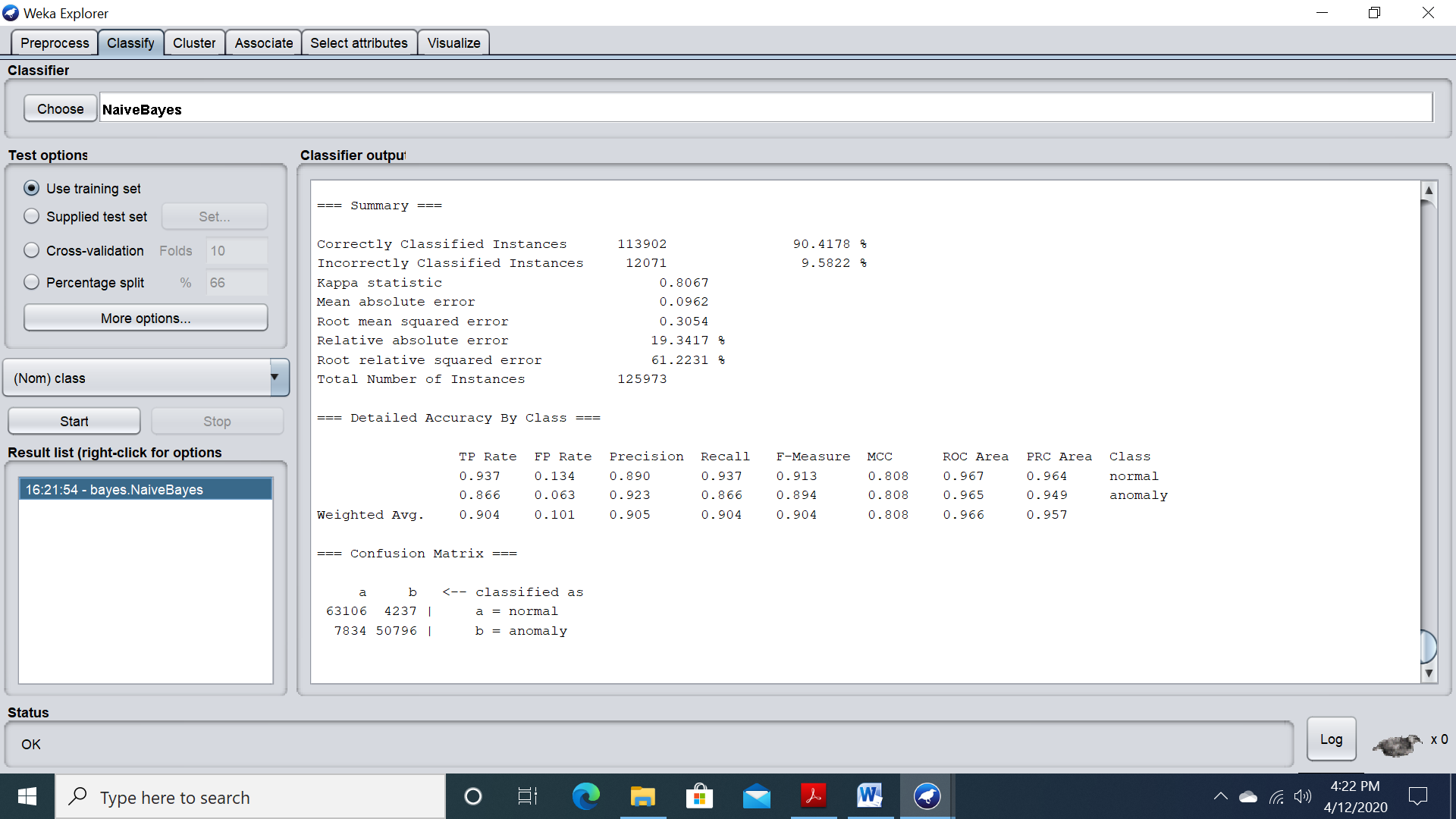
**Task 4.2C- Intrusion detection using supervised learning techniques**

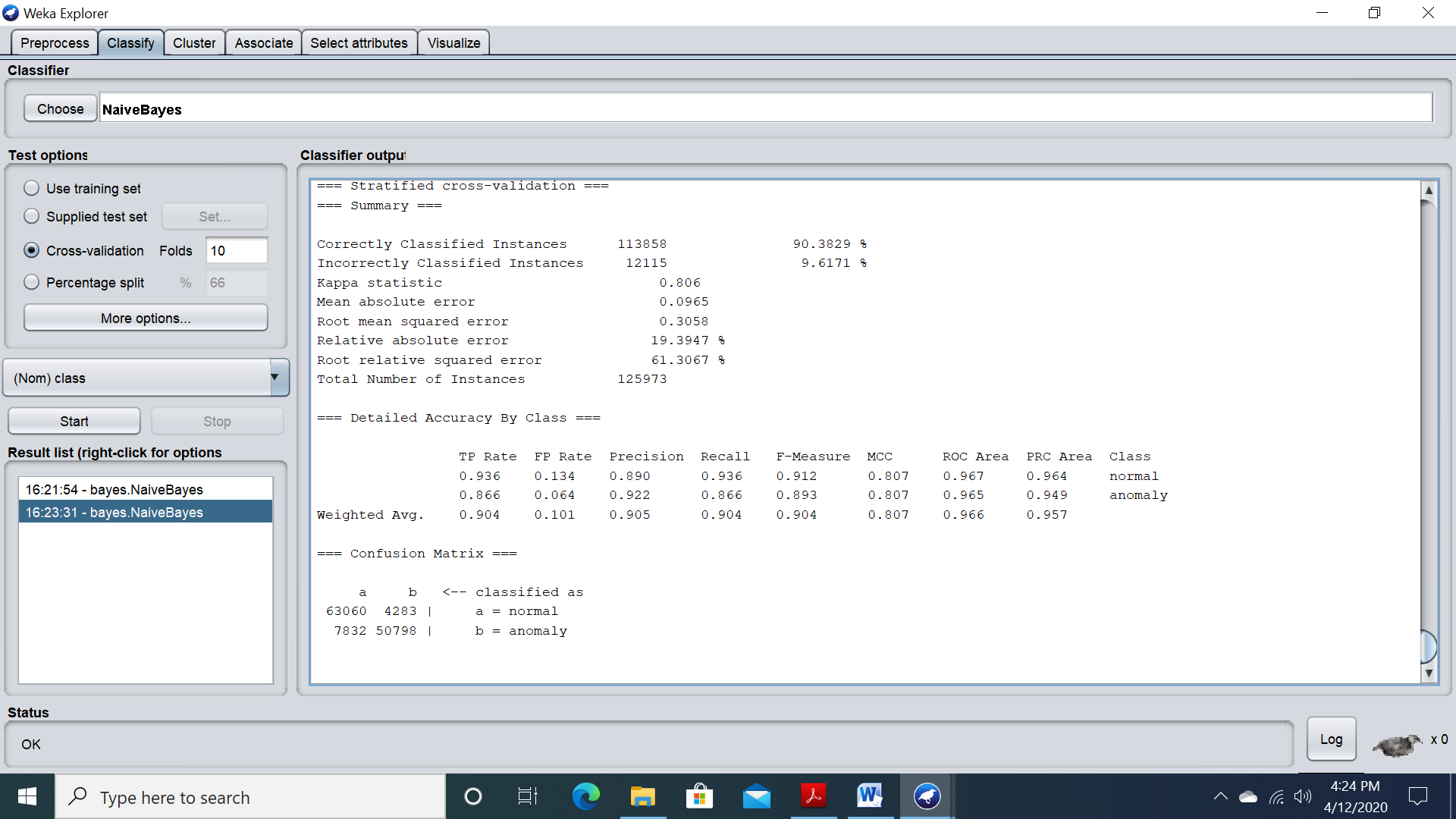
**Step 1: loading the training dataset**



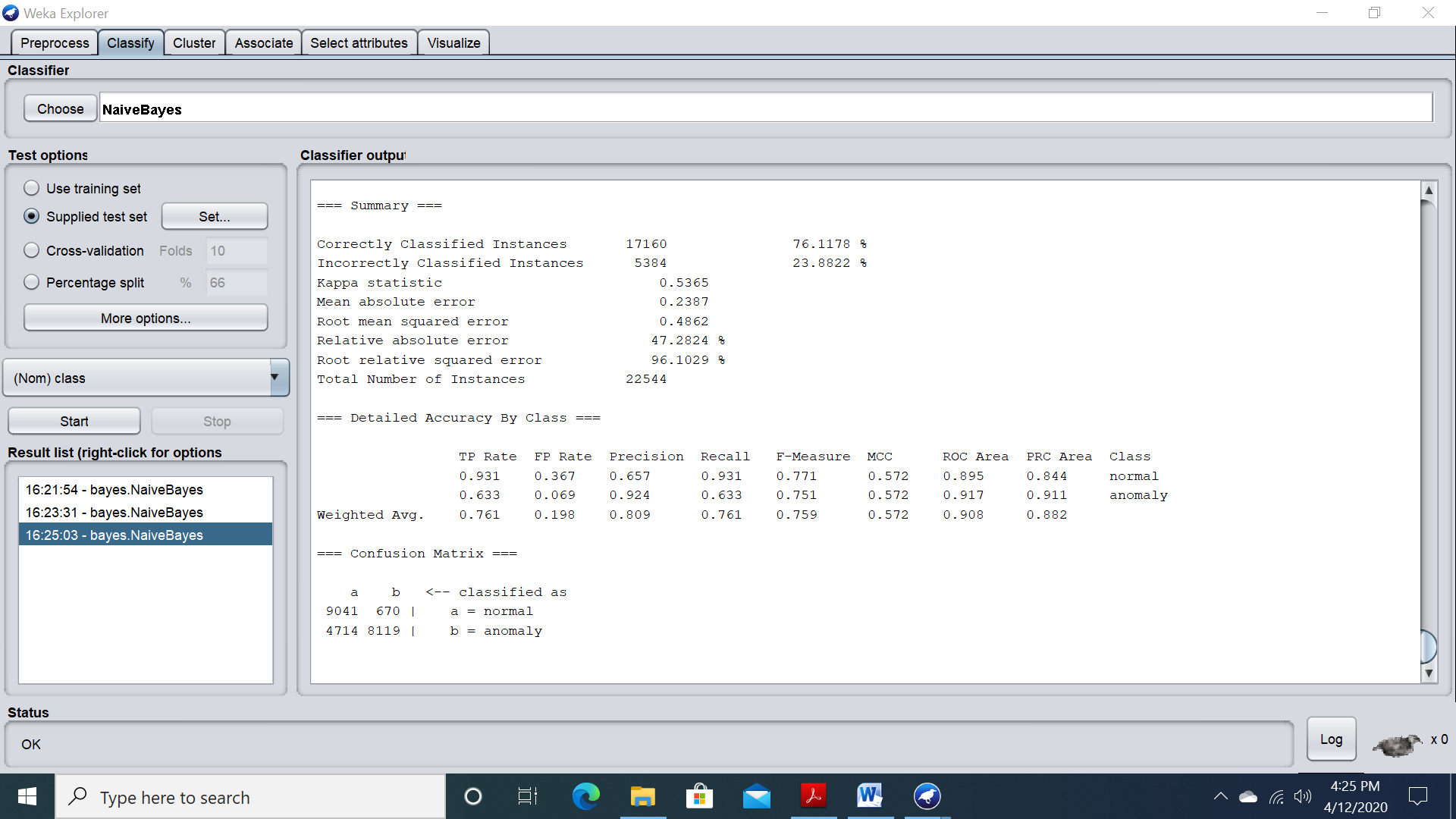
**Step 2: Apply Naïve Bayes Algorithm**



**Step 3: Apply 10-Fold cross-validation**



**Step 4: Apply Test Data-set**



**Step 5: Result comparison**

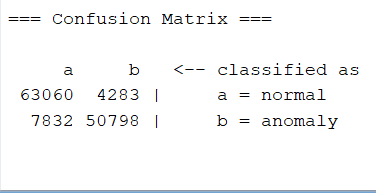


Figure 1: 10 cross-fold

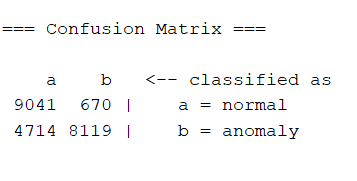
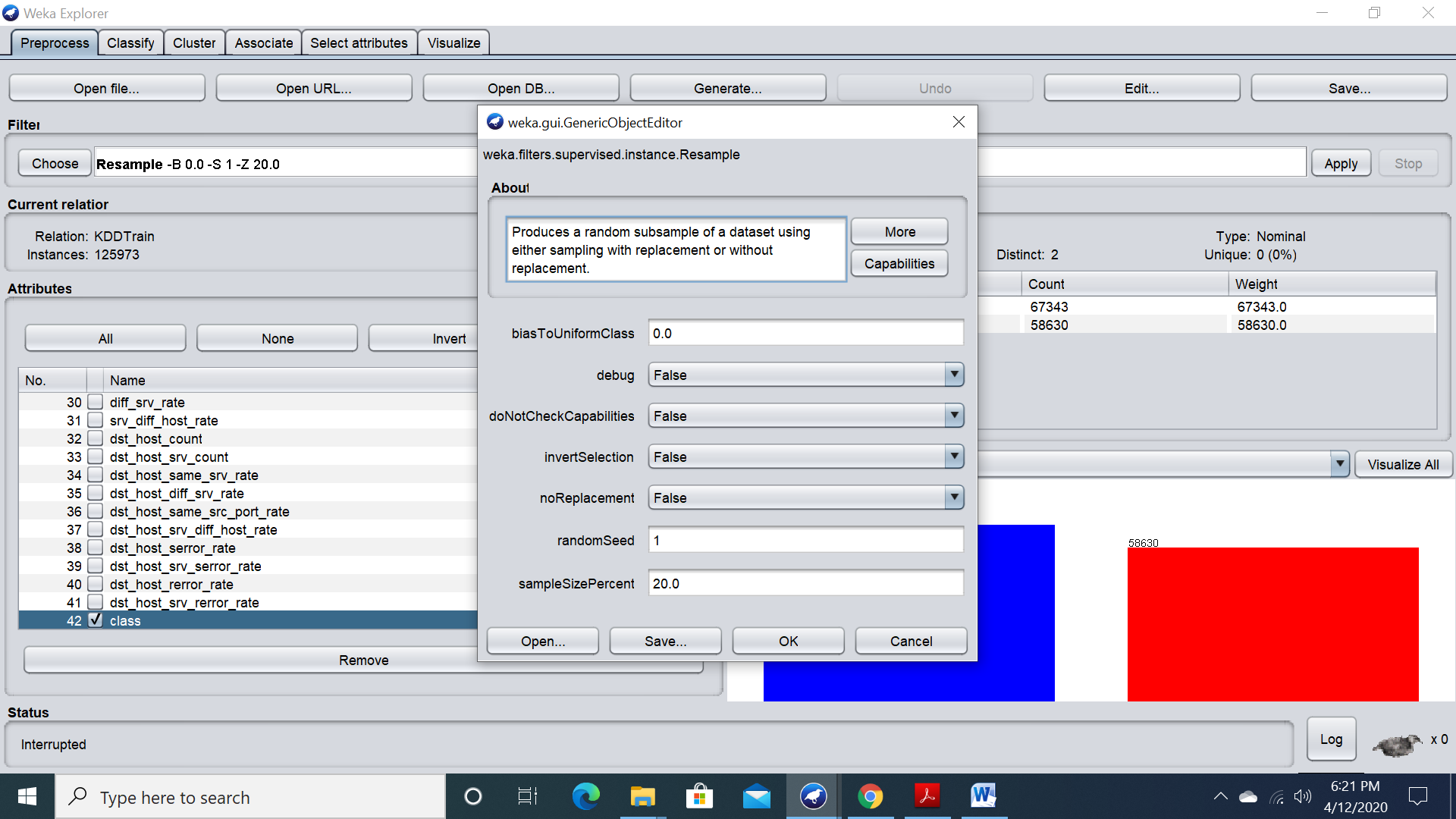


Figure 2: Test set

**Step 6: Applying 5 algorithms**

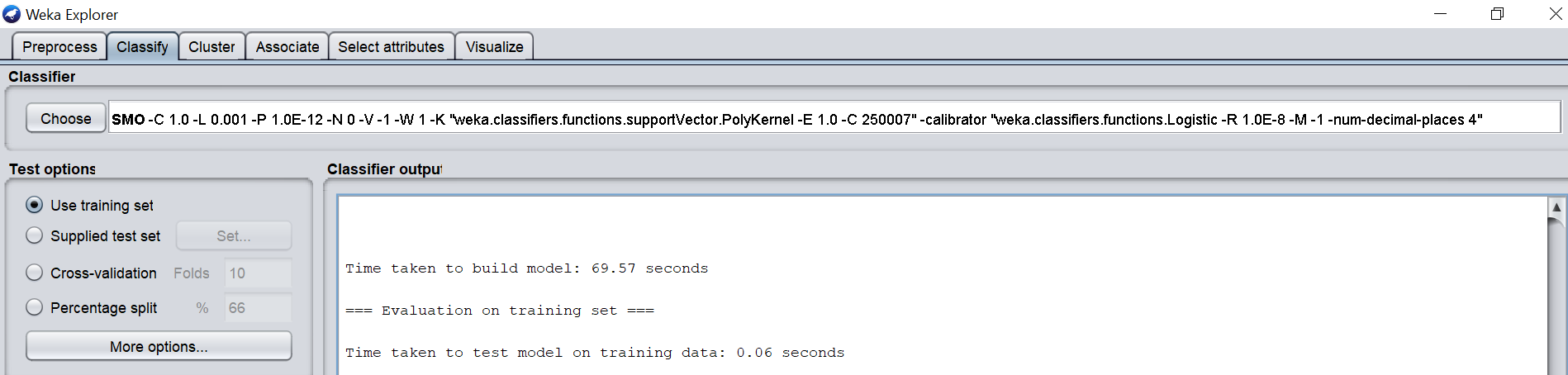
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithms | TP Rate | FP Rate | Precision | Recall | F-measure | MCC | ROC Area | PRC Area |
| Decision Tree Stump | 0.800 | 0.162 | 0.841 | 0.800 | 0.799 | 0.642 | 0.819 | 0.767 |
| Simple Logistic | 0.746 | 0.211 | 0.798 | 0.746 | 0.743 | 0.546 | 0.914 | 0.910 |
| Random Forest | 0.805 | 0.155 | 0.852 | 0.805 | 0.803 | 0.658 | 0.959 | 0.953 |
| J48 | 0.815 | 0.146 | 0.858 | 0.815 | 0.815 | 0.674 | 0.840 | 0.828 |
| Decision Table | 0.726 | 0.214 | 0.814 | 0.726 | 0.718 | 0.545 | 0.949 | 0.944 |

**Step 7: reduce sample size to 20%**

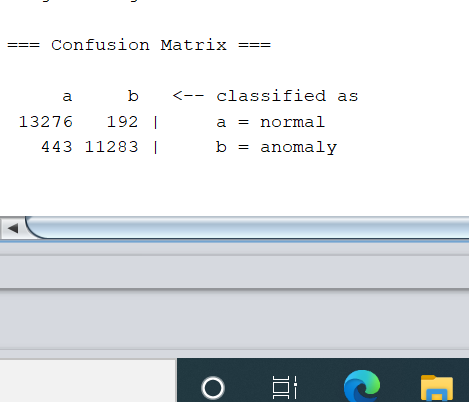


Figure

**Step 8: poly kernel**

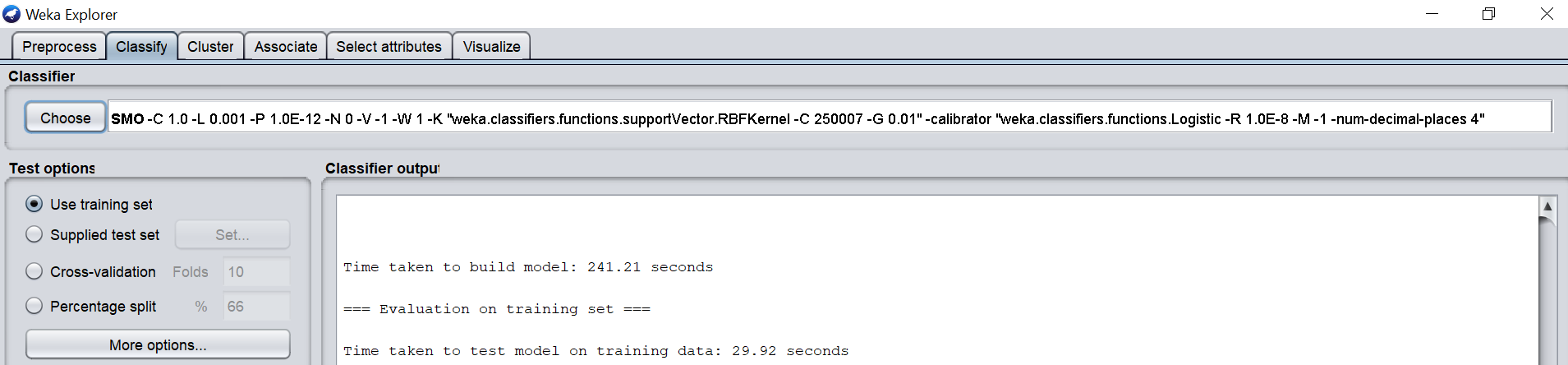


Figure

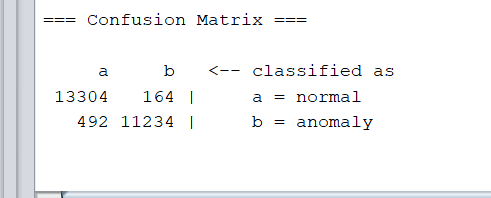


Figure

**Using RBF kernel:**



Figure



Figure

The above figures 1 and 2 provide us with the confusion matrix of both techniques using cross-validation and Test dataset respectively. By comparing the results, we obtain a TP rate of 0.904 by using cross-validation and a TP rate of 0.761 using the test data set making the 10-fold strategy a better measure to predict future values.

By comparing the algorithms stated above the J48 algorithm provides us with the best accuracy giving us a true positive rate of 0.815 in contrast with other algorithms. Moreover, due to the training dataset being of large size, reducing it down to 20 percent of its original size helped algorithms to perform faster due to the large number of attributes present. Furthermore, using the trained data set on the poly kernel and RBF kernel provided with almost the same level of accuracy that were 0.975 and 0.974 respectively. However, the difference between the two types of kernels were its computation times where Poly kernel was less time consuming that amounted to 69.57 seconds and RBF kernel taking more than twice the time of RBF kernel of 241.21 seconds.