Text Mining

Sudhanshu Kulkarni

## **Introduction**

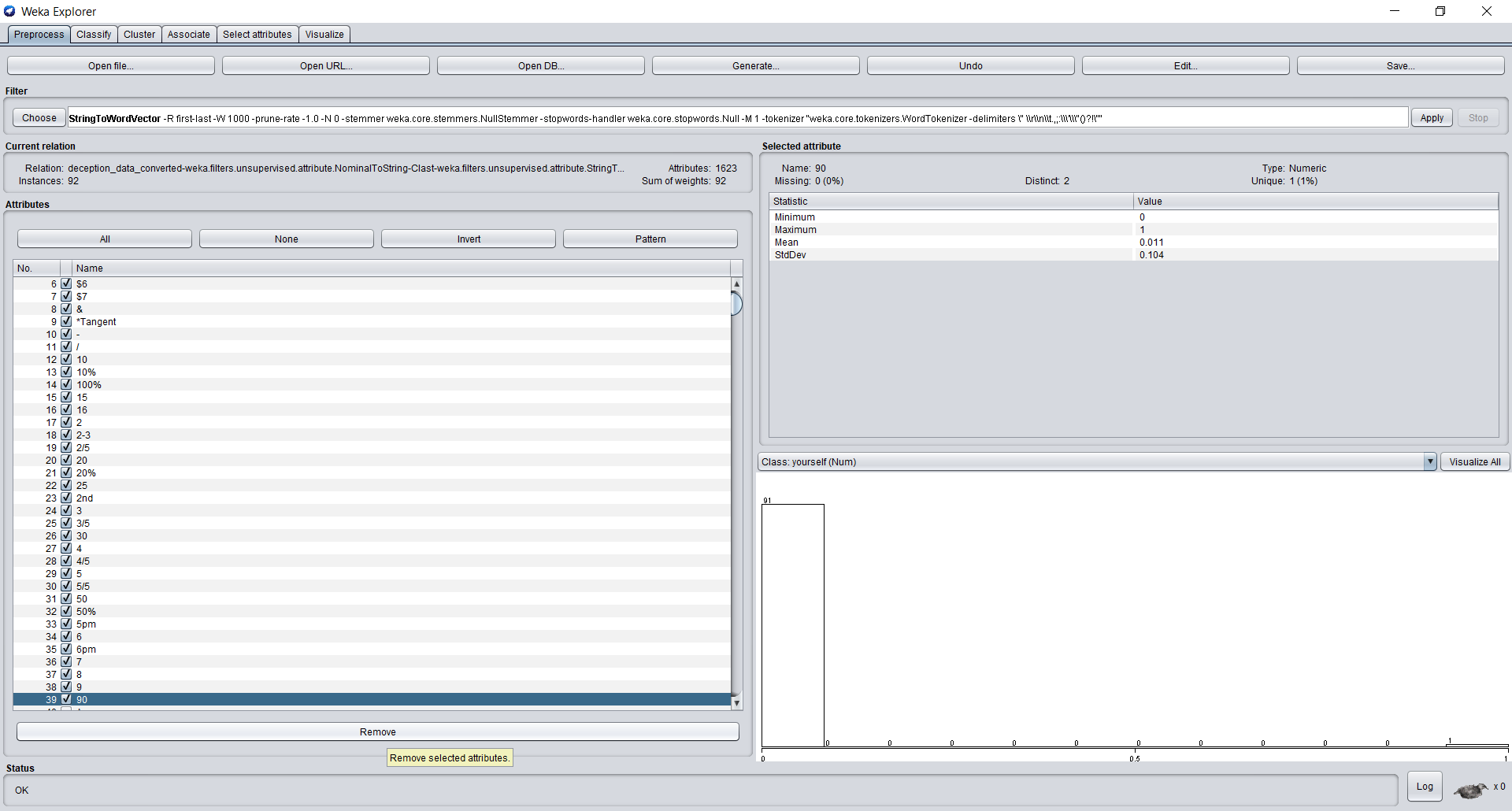
This data set is a collection of customer reviews, some true and some fake. Also, we have a sentiment label for each of the reviews. Using machine learning algorithms like Multinomial Naïve Bayes and Support Vector machine, we have to determine the truthfulness of a review. For this, I have used Weka.

## **Importing Libraries**

We install the package ChiSquaredAttributeEval to rank the features using chi2.

## **Data Preprocessing**

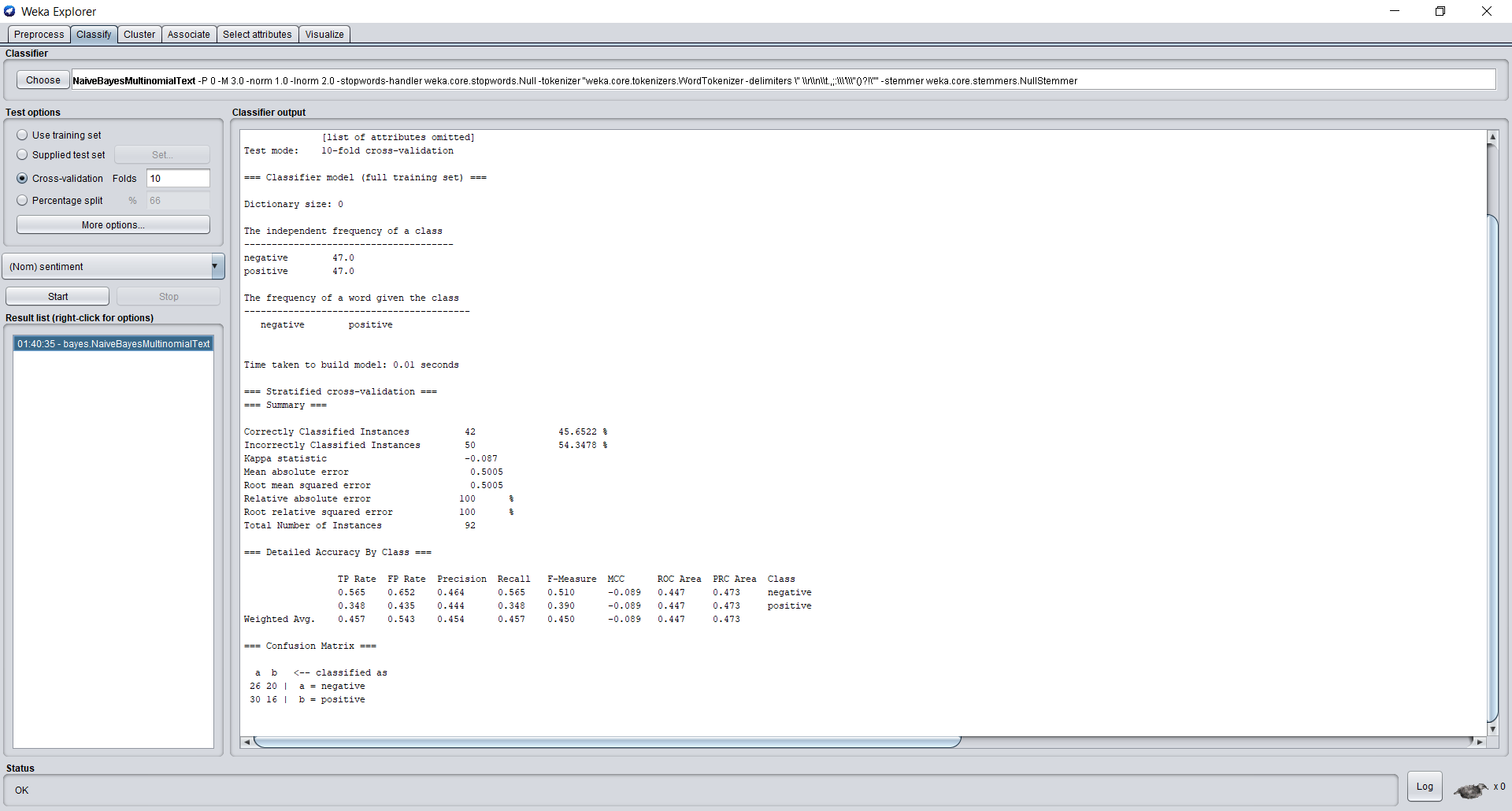
After loading the data, we vectorize the string i.e. we will convert them from string to word vector. Then, we get rid of all the number and special characters as they are not important.

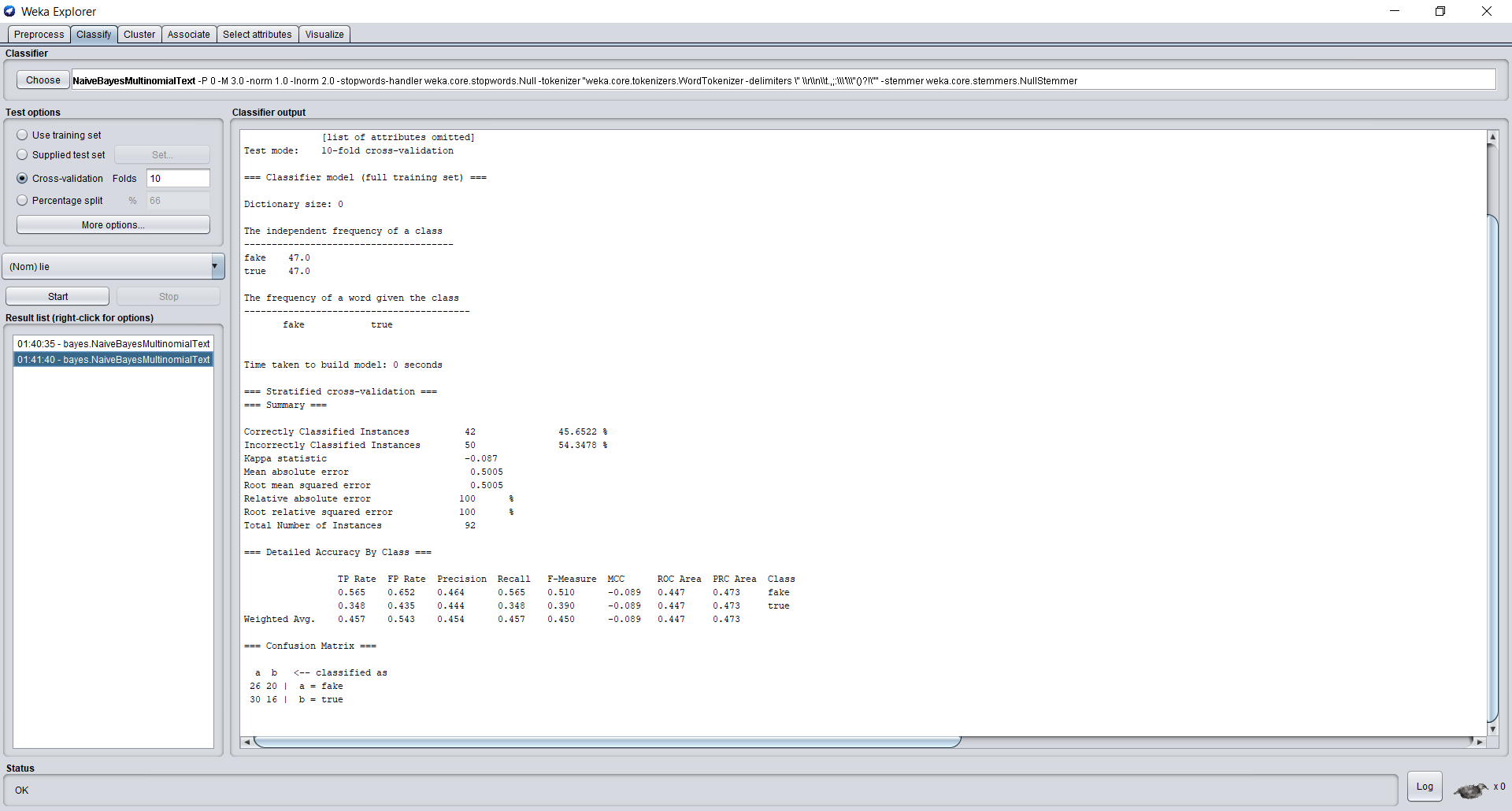


## **Multinomial Naïve Bayes**

We have used Multinomial Naïve Bayes as it is a simple classification method.

**For Sentiment**

  
**For Lie**



Despite tuning the factors, we do not observe any change in the results for sentiment and lie. Below table summarizes the statistics and findings.

Table - MNB comparison for Sentiment

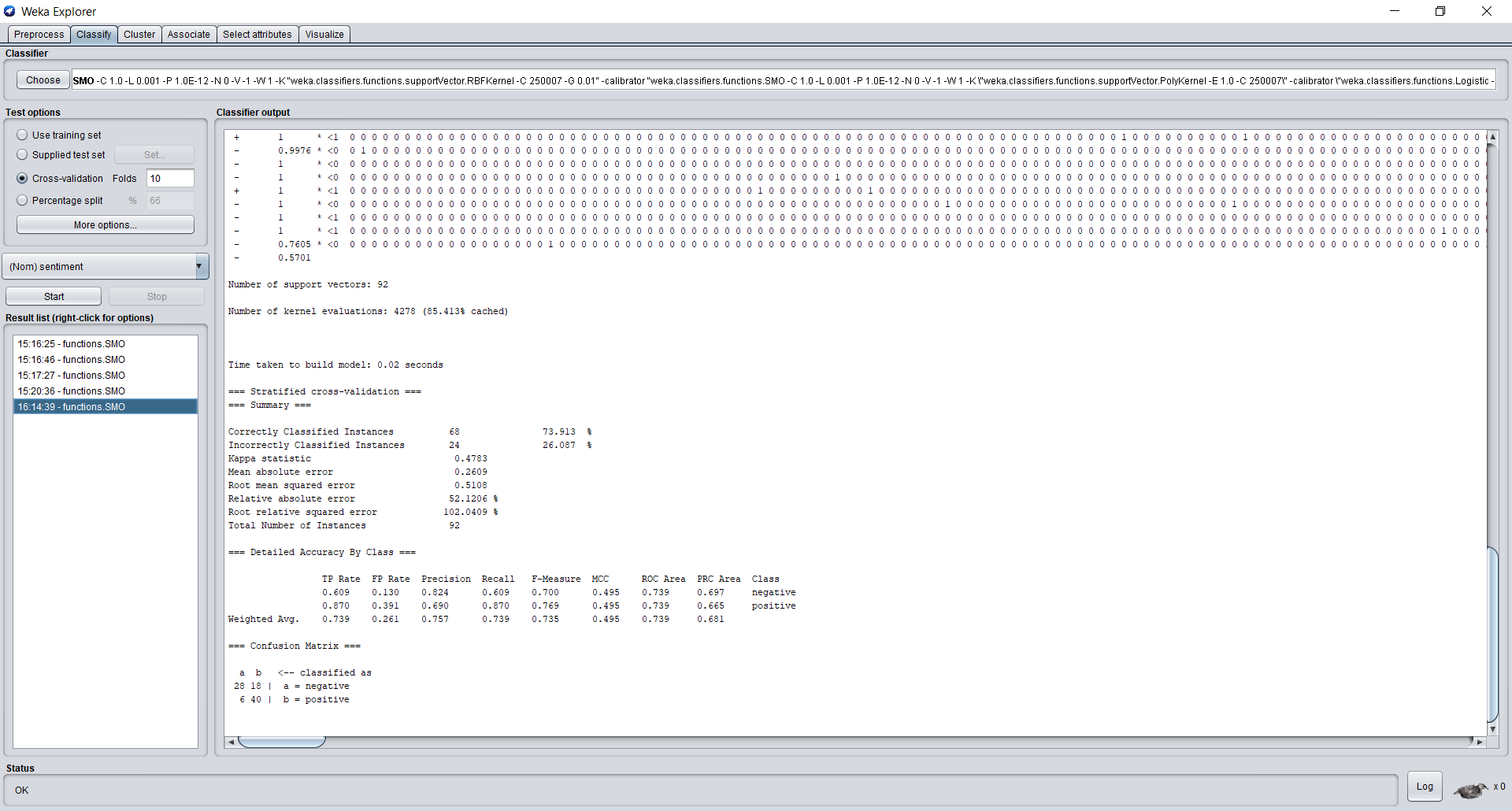
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter setting** | **Overall accuracy** | **Precision in Category 1** | **Recall in Category 1** | **Precision in Category 2** | **Recall in Category 2** |
| Default | 45.65% | 0.464 | 0.565 | 0.444 | 0.348 |
| Lowercasetokens = true  Normalizedoclength = true  Stemmer = lovins stemmer | 45.65% | 0.464 | 0.565 | 0.444 | 0.348 |

Table – MNB comparison for Lie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter setting** | **Overall accuracy** | **Precision in Category 1** | **Recall in Category 1** | **Precision in Category 2** | **Recall in Category 2** |
| Default | 45.65% | 0.464 | 0.565 | 0.444 | 0.348 |
| Lowercasetokens = true  Normalizedoclength = true  Stemmer = lovins stemmer | 45.65% | 0.464 | 0.565 | 0.444 | 0.348 |

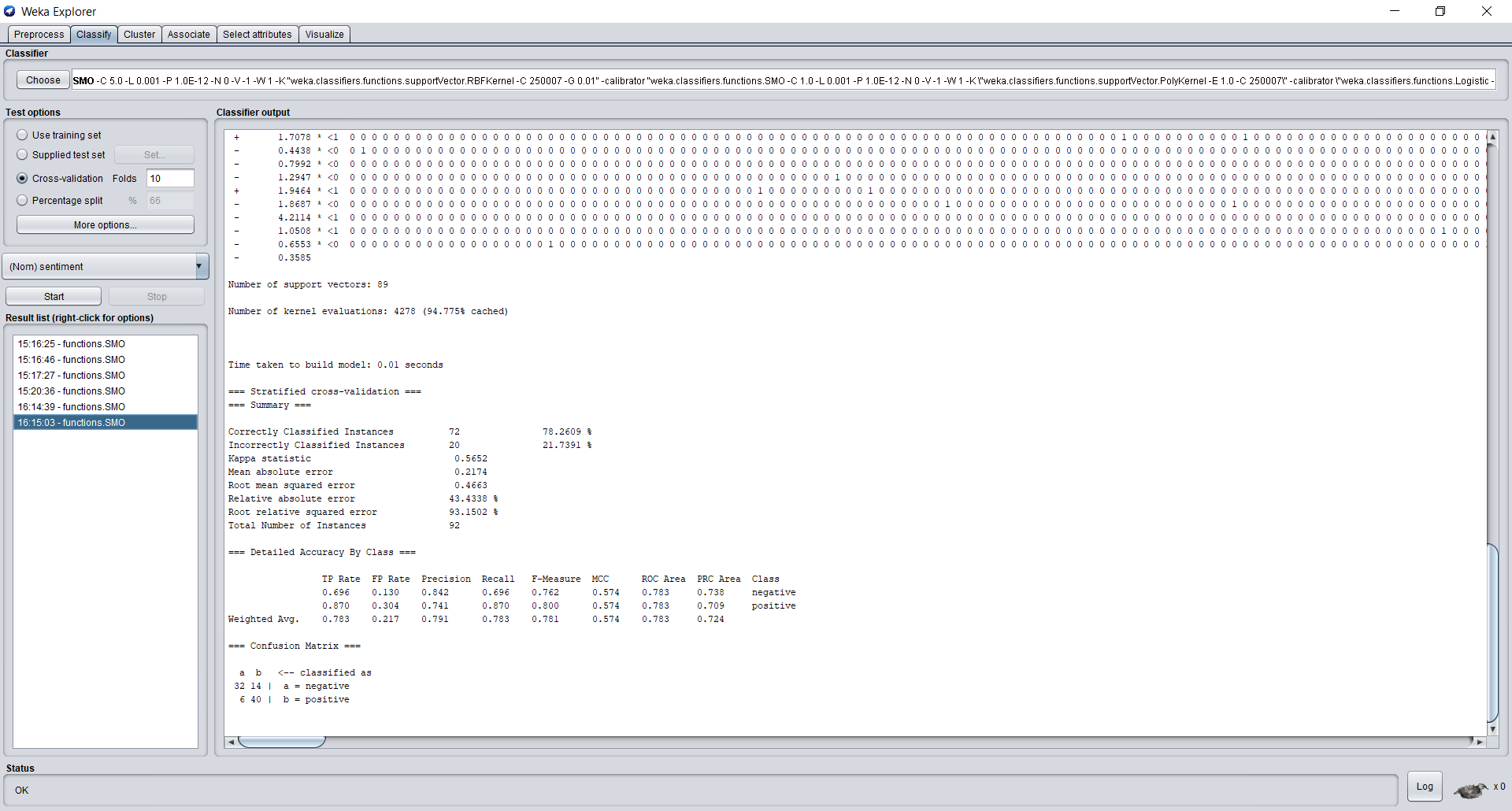
## **SVM**

**For Sentiment – Using default parameters**



**For sentiment – Tuning Parameters**

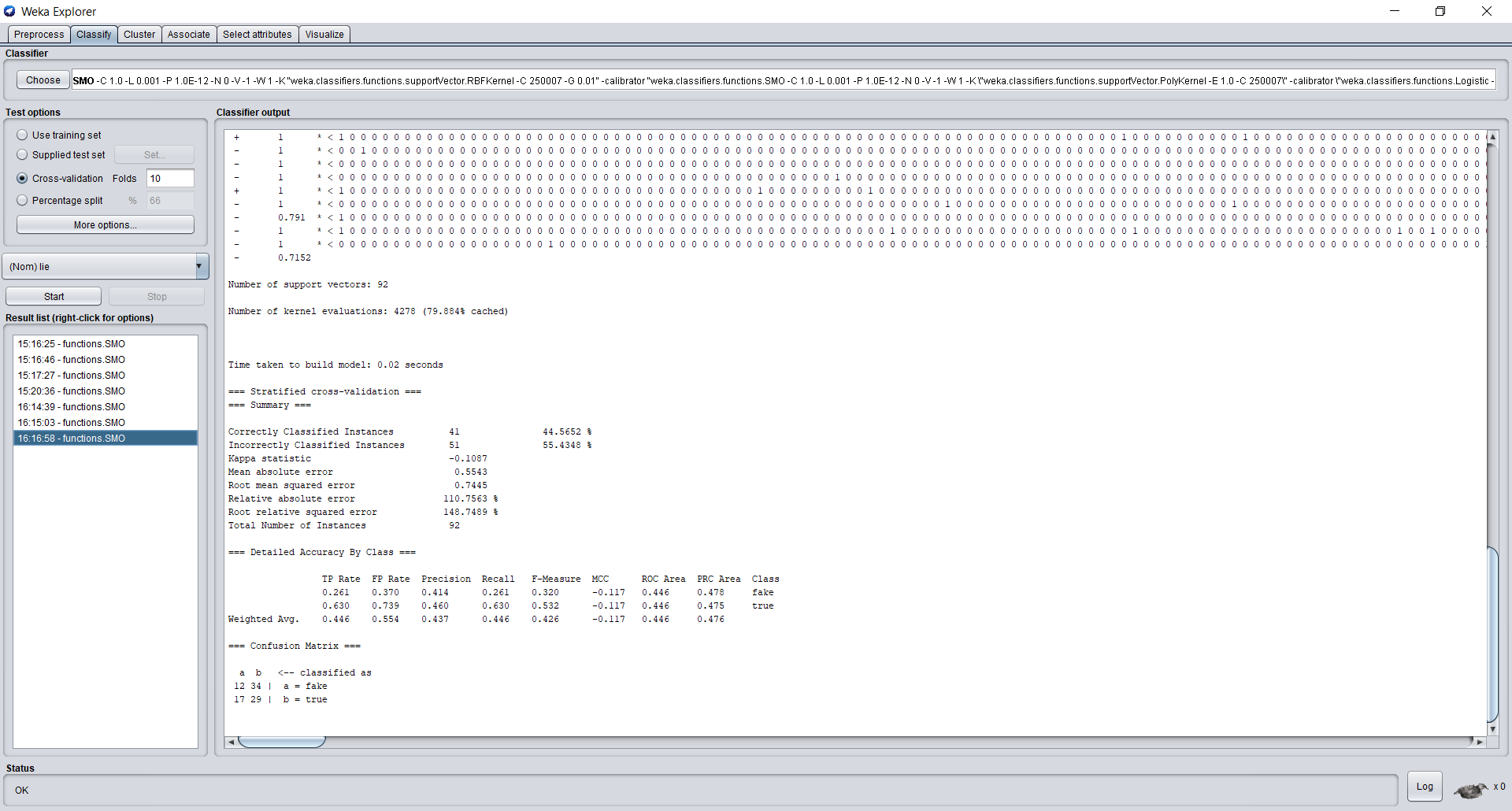
Here we changed the Calibrator to SMO and the Kernel value to RBFKernel.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter setting** | **Overall accuracy** | **Precision in Category 1** | **Recall in Category 1** | **Precision in Category 2** | **Recall in Category 2** |
| Default | 73.91% | 0.824 | 0.609 | 0.690 | 0.870 |
| C = 5  Calibrator = SMO  Kernel = RBFKernel | 78.26% | 0.842 | 0.696 | 0.741 | 0.870 |

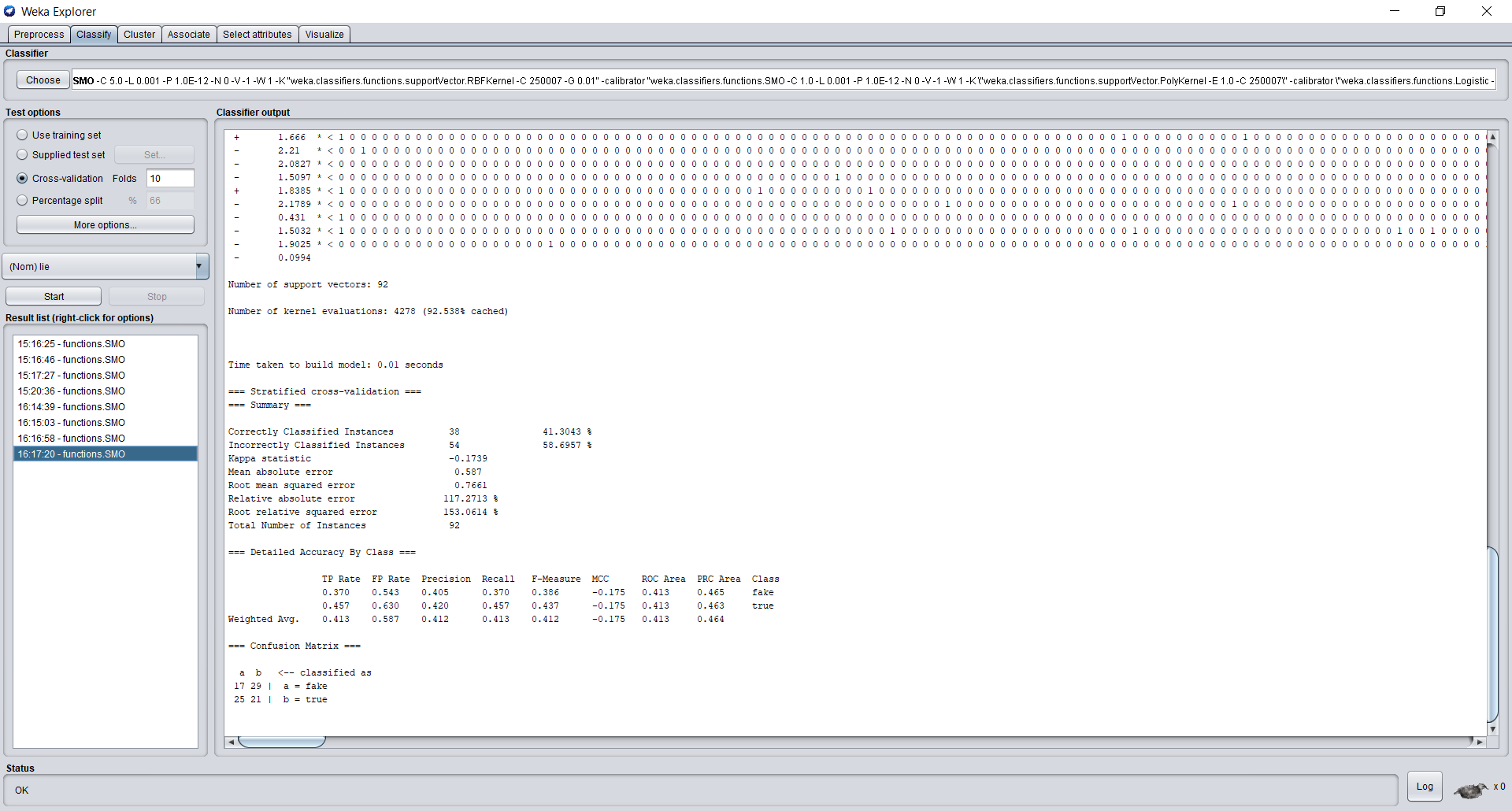
After tuning parameters, we observed that the accuracy went up considerably.

**For Lie**



**For Lie – Tuning Parameters**

Here we have changed the C value to 5, Calibrator to SMO and the Kernel value to RBFKernel.



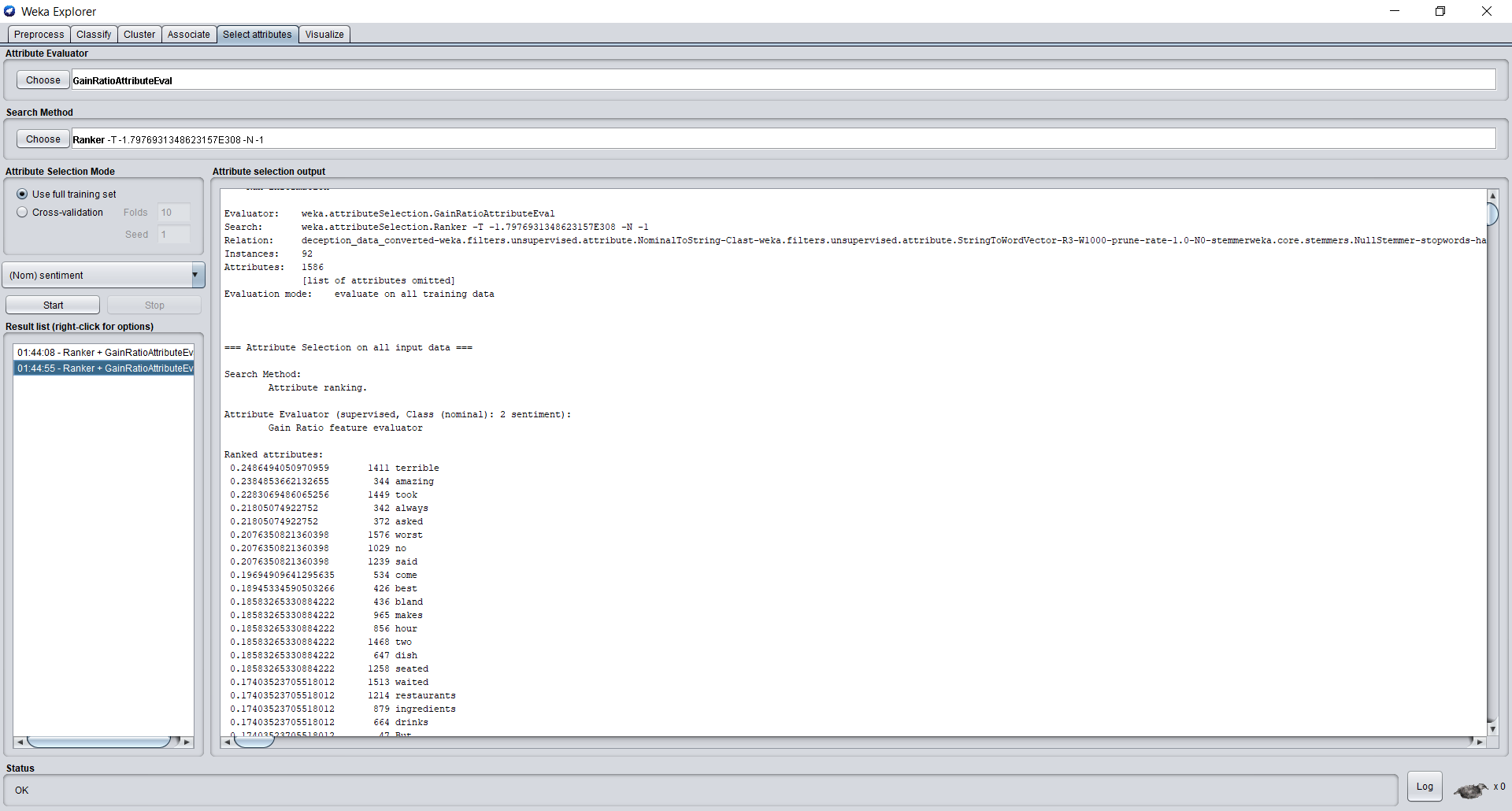
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter setting** | **Overall accuracy** | **Precision in Category 1** | **Recall in Category 1** | **Precision in Category 2** | **Recall in Category 2** |
| Default | 44.565% | 0.414 | 0.261 | 0.460 | 0.630 |
| C = 5  Calibrator = SMO  Kernel = RBFKernel | 41.30% | 0.405 | 0.370 | 0.420 | 0.457 |

After tuning the parameters, we have observed that the accuracy has gone down.

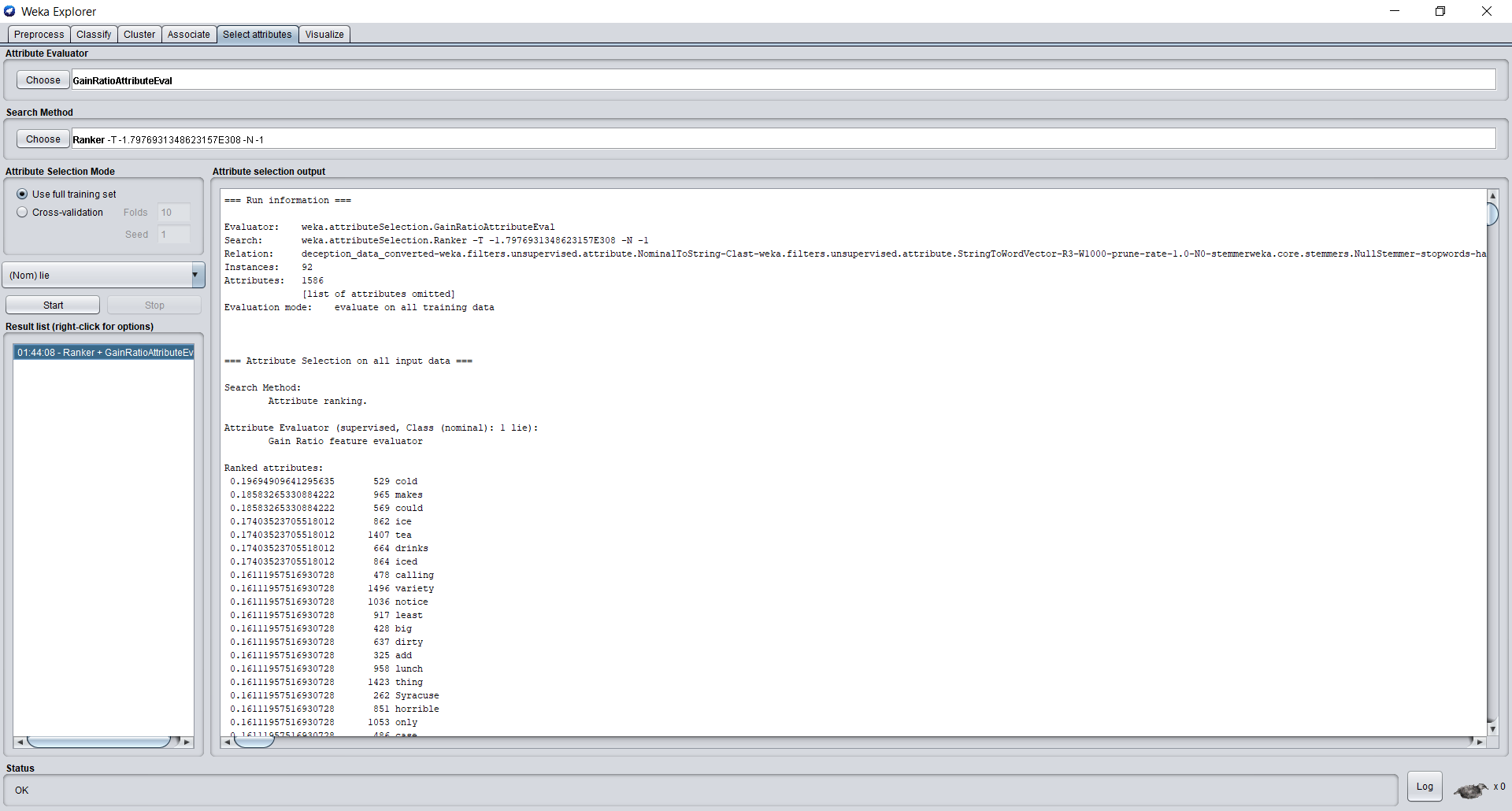
## **Gain Ratio**

Using Gain Ratio, we rank the top 20 features for sentiment and lie.

**For Sentiment**



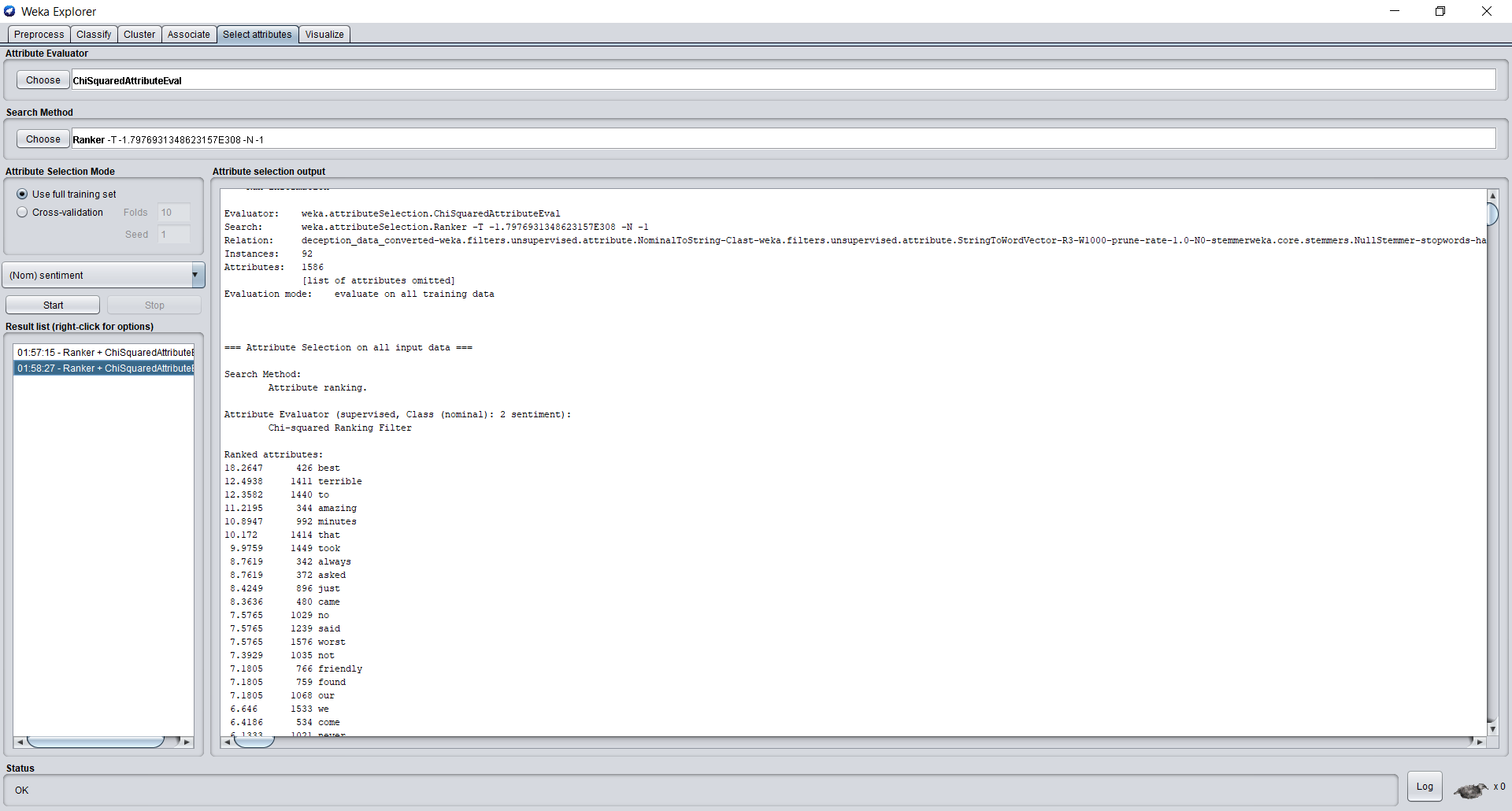
**For Lie**



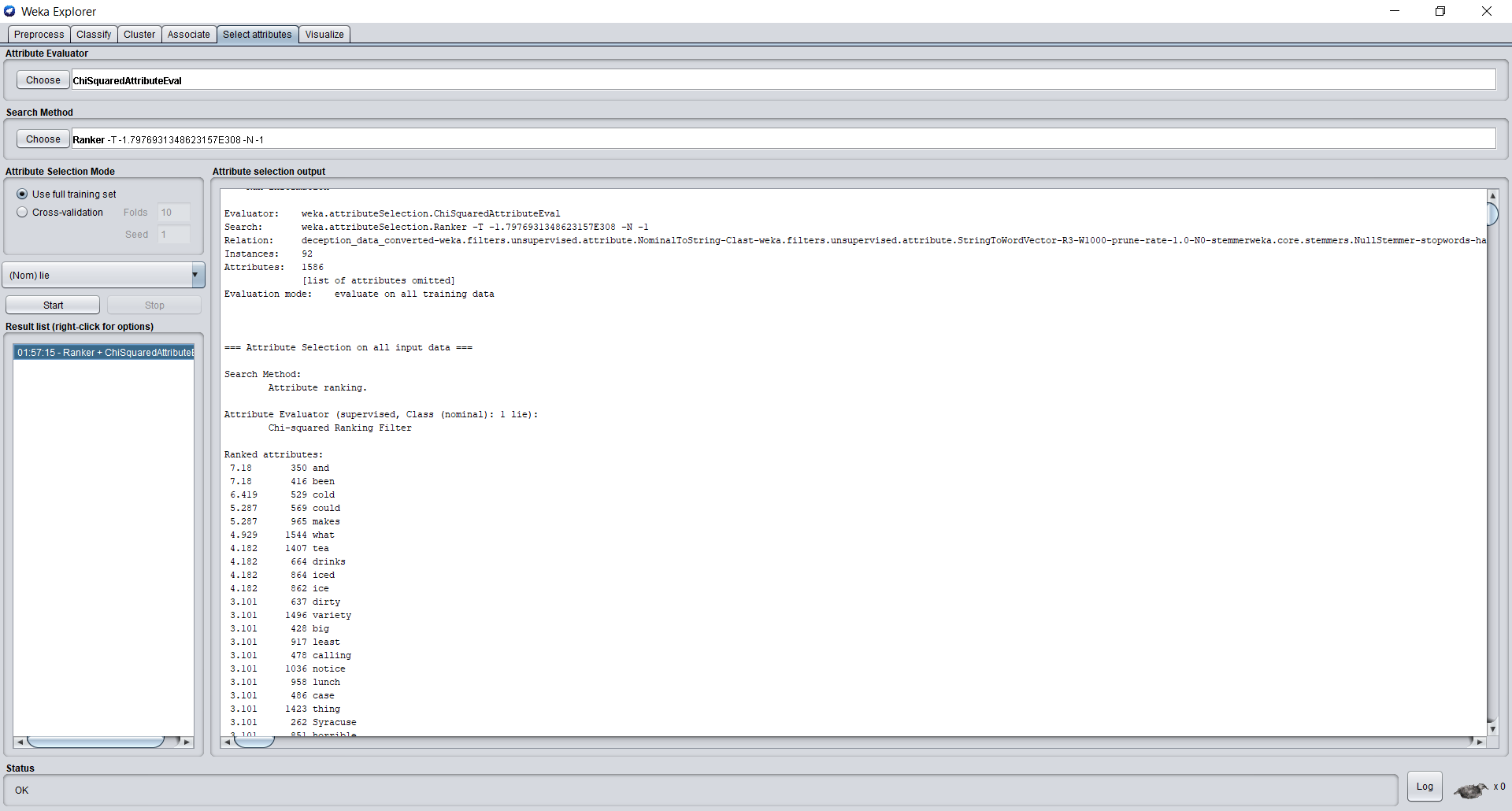
## **Chi2**

Using chi2, we rank the top 20 features for sentiment and lie.

**For Sentiment**



**For Lie**



## **Conclusion**

After doing the analysis, we get to know that classification of sentiment is easier as compared to lie detection. Also, looking at the gain ratio and Chi2 evaluation method, we can witness that for sentiment analysis the algorithm is trying to identify the difference in peoples’ feelings, for eg. Words like amazing and terrible and best and worst. For lie detection, we do not observe any such pattern. Hence, it is difficult to detect lie.