**Home Work – 8  
Karan Ashar**

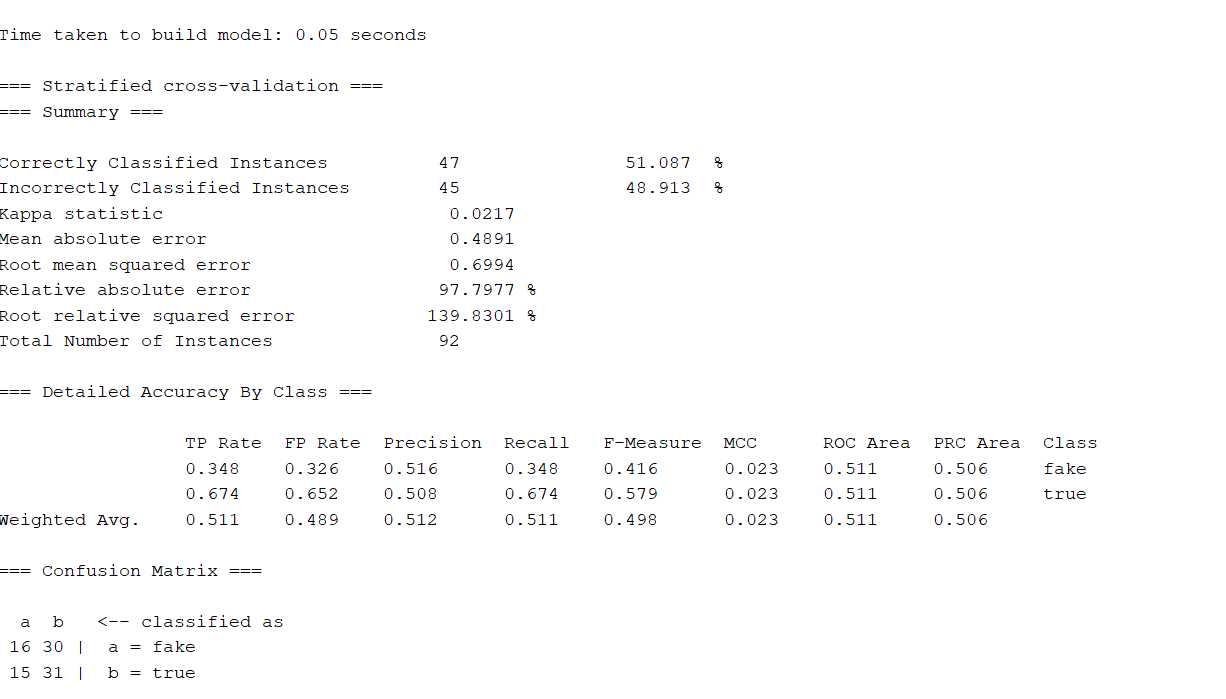
**Initial Parameter Strategy:**

Since this is a text mining task the first thing we need to do is convert the words to vectors. The following preprocessing steps can be done – Stemming, Removing Stop words, Selecting only those words with a minimum frequency, Tokenize the words in a specific manner (e.g. Uni-gram, Bi-gram)

These steps are really important, and they help us increase accuracy.

**Task 1 – Lie Detector**

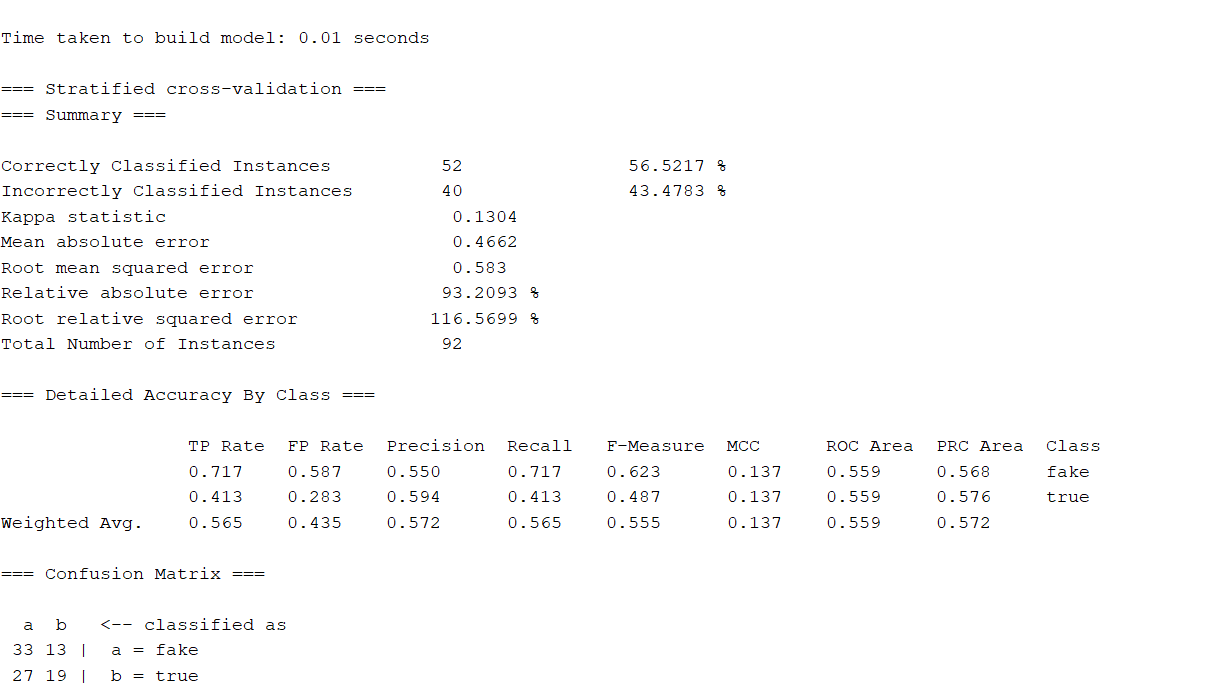
**Model – SVM**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter setting | Overall accuracy | Precision in category I | Recall in category I | Precision in category II | Recall in category II |
| 1.C=0.01  2.minTermFreq=2  3.stopwordsHandler = WordsFromFile  5.tokenizer= WordTokenizer | 51.087% | 0.522 | 0.261 | 0.507 | 0.761 |

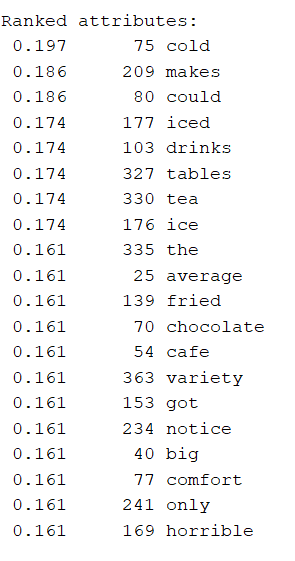
**Model – Multinomial Naïve Bayes**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter setting | Overall accuracy | Precision in category I | Recall in category I | Precision in category II | Recall in category II |
| 1.TFTransform=True  2.minTermFreq=1  3.stopwordsHandler = WordsFromFile  4.tokenizer=WordTokenizer | 56.5217% | 0.55 | 0.717 | 0.594 | 0.413 |



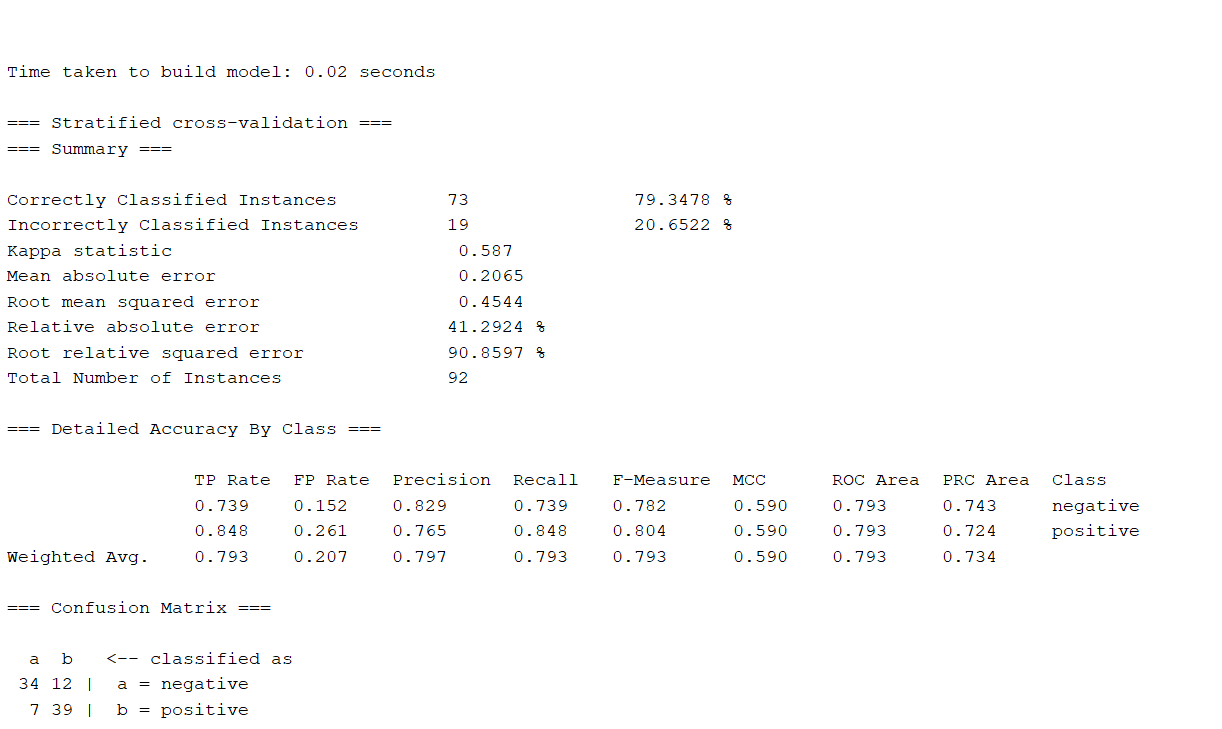
**Top Features**

These are the top 20 features used to make decisions for this task. These are selected based on the Gain Ratio.



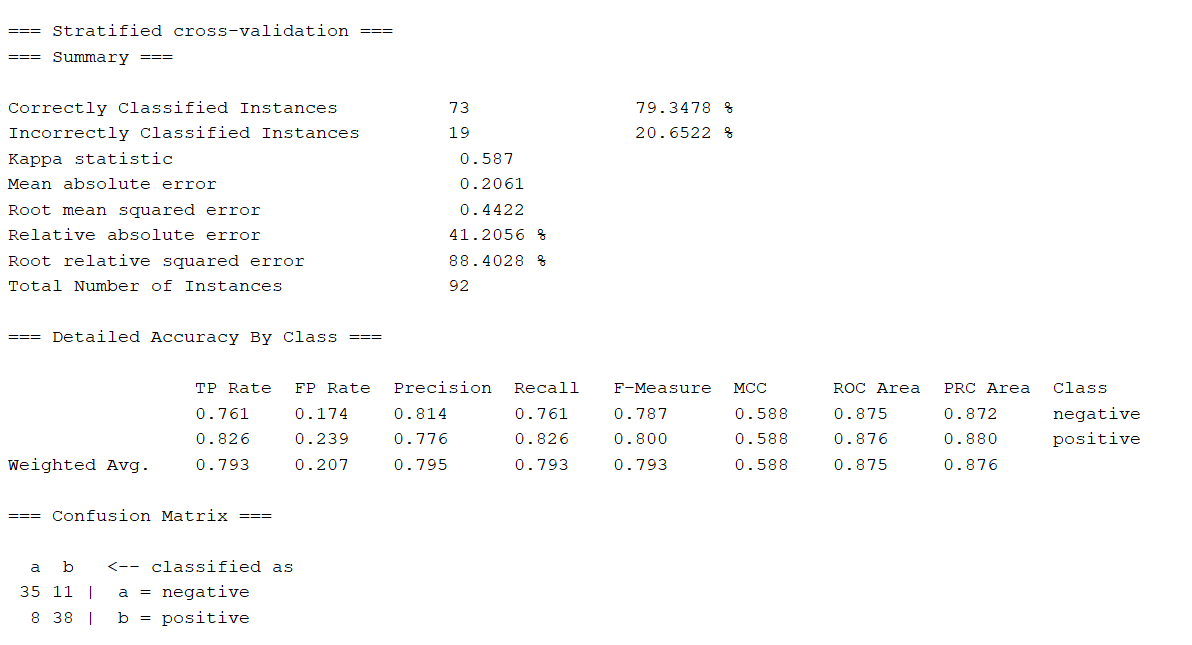
**Task 2 – Sentiment Classifier**

**Model – SVM**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter setting | Overall accuracy | Precision in category I | Recall in category I | Precision in category II | Recall in category II |
| C=0.1  IDFTransform=False  TFTransform=True  lowerCaseTokens=False  minTermFreq=1  stemmer = NullStemmer  stopwordsHandler=WordsFromFile  tokenizer=WordTokenizer  wordsToKeep=1000 | 79.3478% | 0.829 | 0.739 | 0.765 | 0.848 |

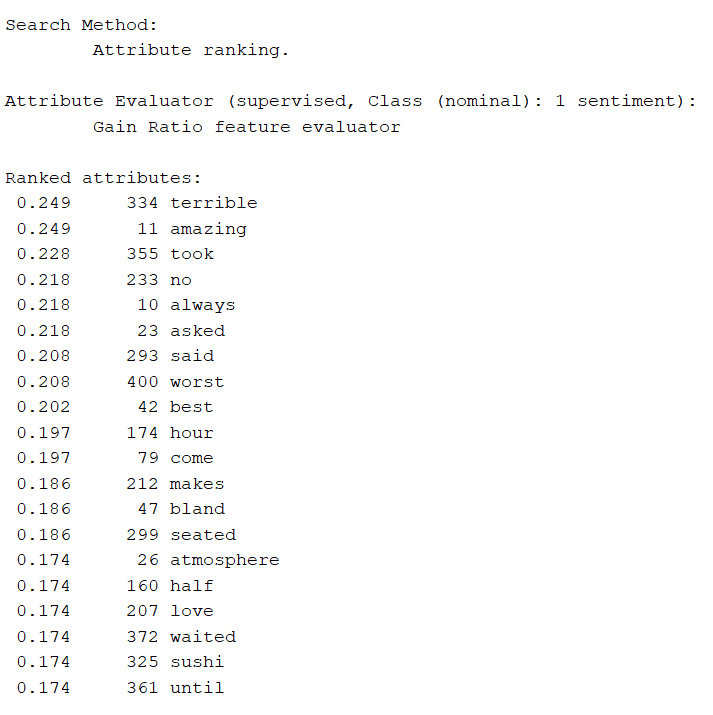
**Model – Multinomial Naïve Bayes**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter setting | Overall accuracy | Precision in category I | Recall in category I | Precision in category II | Recall in category II |
| IDFTransform=True  TFTransform=False  lowerCaseTokens=False  minTermFreq=1  stemmer = NullStemmer  stopwordsHandler=WordsFromFile  tokenizer=WordTokenizer | 79.3478% | 0.814 | 0.761 | 0.776 | 0.826 |

**Top Features**

These are the top 20 features used to make decisions for this task. These are selected based on the Gain Ratio.



**Conclusion**

Both these tasks are completely different. I personally feel that the Lie Detection task is way difficult than the Sentiment Analysis task. This is because, it is very difficult for a human to also say if the person is lying or no. It would be more difficult for the model. Identifying sentiment is easier. This is because the presence of few words would be enough to identify the sentences. A lie can be said by different people in many ways. It would be very difficult to find patterns in lies. But its much easier to identify a sentiment based on the sentence.   
I think the lie detection task would work if there would be some other data as well.

Top Features -   
For the Lie Detection task the top features appear to be random, and no pattern could be detected.

For the sentiment analysis task we can see strong words signifying a positive or negative sentiment. For example, words contributing to negative sentiments could be – terrible, worst. The words contributing to a positive sentiment would be – amazing, best, love.