IST 707 Applied Machine Learning

HW8: Lie Detection and Sentiment Classification with Text Mining

**Scenario:** Some people claim that machine learning algorithms can figure out whether a person is lying or not. Do you believe that? To test this claim, we have assembled a collection of customer reviews — some true and some false — on which you are going to test how good Naïve Bayes and SVMs can be for **fake review detection**. This data set also has sentiment label for each review. You will also compare NB and SVMs performance in **sentiment classification**.

For both tasks, try different tuning parameters and report the results for the **best model of each type** in a table like the following.

1. Explain the data and the pre-processing steps you took to prepare for each classification task.

The given data set contained 93 rows and 3 columns. Each row gives information about restaurant review left by customer, their overall sentiment about their experience, and whether or not the review was true or a lie.

Step:

* I first started with removing first row which contained metadata about the column after importing the dataset
* Rename the column and denote them with appropriate description
* Check for null values
* Convert the sentiment column from character to factor
* Convert the lie column from character to factor
* Tokenization: converting every word to a token
* Removing stop words:
* Convert each word to smaller case.
* Remove words containing smaller case letter
* Remove blank entries
* Conclude similar words with lemmatization
* Make each unique word as a column and provide their sentiment, lie status.
* Make partition and divide dataset to test and train.
* Remove Index and Sentiment column from lie train and test set. Consequently removing Index and lie column from Sentiment train and test set.

1. Explain your initial parameter tuning strategy — which parameter to tune, to what option, and the theoretical foundation for your choice. Does your strategy help you get better results?

* For SVM I used degree, scale and C:
* Degree which denotes degree of polynomial kernel I used. Values I used were 1,2,3
* Scale ensures that my predictors contribute equally in predicting my dependant variable (lie and sentiment).
* C helps to prevent overfitting. Values range from 0.1 to 2 with 10 even difference between values
* For Naïve Bayes, usekernel, laplace and adjust.
* Usekernel has two values TRUE and false, which tells us to use kernel or not
* Laplace, 0 and 1. Laplace tells us to use smoothing or not.
* Ajust : provides information about degree of smoothing. (0,1,2)

1. Compare performance differences in sentiment classification and lie detection and tell us which task is harder. Try to explain why one may be harder than the other.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Parameter Settings | Accuracy  Lie Detection | Precision  Lie Detection | Recall  Lie Detection | Accuracy Sentiment | Precision Sentiment | Recall Sentiment |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SVM | **degree** = c(1, 2, 3)  **scale** = c(0.001, 0.01, 0.1, 1.0)  **C** = seq(0.1, 2, length = 10) | 0.5 | 0.5 | 0.4545 | 0.8182 | 1 | 0.6364 |
| Naïve Bayes | **usekernel** = c(TRUE, FALSE),  **laplace** = c(0, 1),  **adjust** = c(0,1,2) | 0.5 | 0.5 | 1 | 0.5 | 0.5 | 1 |

For detection of sentiment, SVM provided higher accuracy of 81.82%. whereas for detection of lie, accuracy was same for both SVM and Naïve Bayes. The only segment Naïve Bayes was better than SVM for lie detection was in recall percentage with 100%.

It was harder to detect sentiment with naïve bayes mainly due to existing hyperparameters that I used for tuning.

For lie detection Both model had similar output except that Naïve bayes had better recall percentage. I can improve SVM model mainly by adjust tuning parameters and creating different partition between train and test set. One more thing I can try is using linear method instead of poly method while training.