School of Computer Science
University College Cork

CS6405: Data Mining CA Assignment Report

Submitted to:

Dr. Alejandro Arbelaez, Department of Computer Science, UCC

Submitted by:

Student Name: Aausuman Deep
Student Number: 119220605

Dated – 3rd April 2020

Table of Contents

Introduction

The tasks listed in the given assignment have been performed in the attached R code file.

The R file consists of basically three parts.

1. Setting up the environment:
    a. Cleaning the environment and setting a variable to the path where the articles of the newsgroups are stored on my local system.
    b. Creating five functions which will take in some input data and return predictions based on five classification algorithms. These five algorithms are:
        i. Naïve Bayes
        ii. KNN
        iii. Random Forest
        iv. Decision Tree
        v. SVM

2. Working on the unprocessed data:
    a. Reading in the bag of words without preprocessing
    b. Exploration tasks
    c. Basic evaluations by calculating accuracies of various classification algorithms on this data. These will include:
        i. Naïve Bayes
        ii. KNN
        iii. Random Forest

3. Working on the preprocessed data:
    a. Reading in the bag of words with preprocessing.
    b. Exploration tasks
    c. Basic evaluations by calculating accuracies of various classification algorithms on this data and comparing these results with the ones we got with unprocessed data for the same. This will include:
        i. Naïve Bayes
        ii. KNN
        iii. Random Forest
    d. Robust evaluations on the data. This will include:
        i. Hold out CV on Random Forest
        ii. Decision Tree Classification
        iii. Support Vector Machine Classification
        iv. Feature Selection by Recursive Feature Elimination using Random Forest
        v. Performing all five classification algorithms on the data reduced by feature selection

Basic Evaluation

While working with unprocessed data, we read in the raw data as a bag of words. Due to memory challenges, I have only taken first two words of each line in every file.

Exploration tasks have been performed on the bag of words, and the respective outputs have been printed out as can be executed in the attached R file.

After splitting the bag of words into a random 70% training set and 30% remaining test set, we perform the three classification techniques on the training set and then check their performance on the test set.

1. Naïve Bayes – This classification does not seem to perform too well. The accuracy is 27.5%

   The confusion matrix for it is as follows:

```
predictions                comp.sys.ibm.pc.hardware sci.electronics talk.politics.guns
  comp.sys.ibm.pc.hardware                       33              33                 24
  sci.electronics                                 0               0                  0
  talk.politics.guns                              0               0                  0
  talk.politics.misc                              0               1                  0

predictions                talk.politics.misc
  comp.sys.ibm.pc.hardware                 29
  sci.electronics                           0
  talk.politics.guns                        0
  talk.politics.misc                        0
```

   The Recall values for the four classes by this classifier are as follows:

```
> recall_class1
[1] 0.2773109

> recall_class2
[1] NaN

> recall_class3
[1] NaN

> recall_class4
[1] 0
```

2. KNN – This classification also does not perform very well, although it is slightly better than Naïve Bayes. The accuracy is 38.3%

3. Random Forest – This classification performs considerably better than the other two techniques on the dataset with an accuracy of 78.33%

   The confusion matrix for it is as follows:

```
predictions                comp.sys.ibm.pc.hardware sci.electronics talk.politics.guns
  comp.sys.ibm.pc.hardware                       29               5                  2
  sci.electronics                                 1              26                  2
  talk.politics.guns                              0               0                 16
  talk.politics.misc                              3               3                  4

predictions                talk.politics.misc
  comp.sys.ibm.pc.hardware                  2
```

```
  sci.electronics                           0
  talk.politics.guns                        4
  talk.politics.misc                       23
```

The Recall values for the four classes by this classifier are as follows:

```
> recall_class1
[1] 0.7631579

> recall_class2
[1] 0.8965517

> recall_class3
[1] 0.8

> recall_class4
[1] 0.6969697
```

Seeing these accuracies together, we can clearly see that none of the classification techniques used here can be easily justified as one to use here. There is a definite need to tune the data further.

```
> # Accuracy of these three classifiers on our unprocessed data

> cbind(accuracy_nb_1, accuracy_knn_1, accuracy_rf_1)
         accuracy_nb_1 accuracy_knn_1 accuracy_rf_1
Accuracy         0.275      0.3833333     0.7833333
```

Robust Evaluation

In this part, first we preprocess the data. The steps followed are as mentioned below:

1. We create the repository of words again, with converting all non-alphabet as a space and then splitting the text on space and converting everything to lower case.

2. Removing the stop words from our repository. The stop words considered by me is only a vector of 7 words ("a", "and", "the", "to", "in", "that", "so"), but we can take more if we want.

3. Next, I am removing all words which have length less than 3, which means that words of length 1 and 2 which are not that meaningful to contribute towards our classification have been removed.

4. With this processed data, we read it in as a bag of words. Due to memory challenges, I have only taken first two words of each line in every file.

Exploration tasks have been performed on this new bag of words, and the respective outputs have been printed out as can be executed in the attached R file.

After performing these exploration tasks, I further decide to remove those variables from our bag which have total frequency across all 400 files, less than 10. This means that words which have occurred less than 10 times combined in these 400 files might not have that impact on classification, and hence have been removed from the bag.

After splitting the bag of words into a random 70% training set and 30% remaining test set, we perform the three classification techniques on the training set and then check their performance on the test set.

1. Naïve Bayes – This classification does improve a little bit than before but still does not seem to perform too well. The accuracy is 29.16%

   The confusion matrix for it is as follows:

```
predictions                comp.sys.ibm.pc.hardware sci.electronics talk.politics.guns
  comp.sys.ibm.pc.hardware                       31              30                 20
  sci.electronics                                 2               3                  3
  talk.politics.guns                              0               0                  1
  talk.politics.misc                              0               1                  0

predictions                talk.politics.misc
  comp.sys.ibm.pc.hardware                 27
  sci.electronics                           1
  talk.politics.guns                        1
  talk.politics.misc                        0
```

   The Recall values for the four classes by this classifier are as follows:

```
> recall_class1
[1] 0.287037

> recall_class2
```

```
[1] 0.3333333
> recall_class3
[1] 0.5

> recall_class4
[1] 0
```

2. KNN – This classification has improved considerably and performs ok. It is much better than Naïve Bayes. The accuracy is 68.3%

3. Random Forest – This classification performs magnificently with an accuracy of 96.67%. This classifier should be the chosen one for this particular dataset.

   The confusion matrix for it is as follows:

```
predictions                comp.sys.ibm.pc.hardware sci.electronics talk.politics.guns
  comp.sys.ibm.pc.hardware                       33               1                  0
  sci.electronics                                 0              32                  0
  talk.politics.guns                              0               0                 24
  talk.politics.misc                              0               1                  0

predictions                talk.politics.misc
  comp.sys.ibm.pc.hardware                  0
  sci.electronics                           0
  talk.politics.guns                        2
  talk.politics.misc                       27
```

   The Recall values for the four classes by this classifier are as follows:

```
> recall_class1
[1] 0.9705882

> recall_class2
[1] 1

> recall_class3
[1] 0.9230769

> recall_class4
[1] 0.9642857
```

Comparing these accuracies with the ones we got for unprocessed data, we clearly see an improvement and conclude that preprocessing this data has had an impact on classification.

```
> cbind(accuracy_nb_1, accuracy_nb_2)
         accuracy_nb_1 accuracy_nb_2
Accuracy         0.275     0.2916667

> cbind(accuracy_knn_1, accuracy_knn_2)
    accuracy_knn_1 accuracy_knn_2
acc      0.3833333      0.6833333

> cbind(accuracy_rf_1, accuracy_rf_2)
     accuracy_rf_1 accuracy_rf_2
[1,]     0.7833333     0.9666667
```

4. Hold Out and Cross validation on our data using Random Forest method.

The rmse for the train data in that case is 0, and similarly 0.0416 for test set. Since these two root mean squared errors are quite close to each other, we can conclude that by the evidence of this cross validation, random forest will perform well for this dataset.

```
> cbind(rmse_train_set, rmse_test_set)
     rmse_train_set rmse_test_set
[1,]              0    0.04166667
```

5. Decision Tree - This classification performs magnificently with an accuracy of 95%. This classifier should be the next choice after random forest for this particular dataset.

6. Support Vector Machine – We run SVM with three different kernel types on our dataset. For 'linear' kernel type, the accuracy is 95%. The confusion matrix is as follows:

```
                          Reference
Prediction                 comp.sys.ibm.pc.hardware sci.electronics talk.politics.guns
  comp.sys.ibm.pc.hardware                       33               0                  0
  sci.electronics                                 1              32                  1
  talk.politics.guns                              0               0                 24
  talk.politics.misc                              0               0                  4
                          Reference
Prediction                 talk.politics.misc
  comp.sys.ibm.pc.hardware                  0
  sci.electronics                           0
  talk.politics.guns                        0
  talk.politics.misc                       25
```

The Recall values for the four classes by this classifier are:

```
> recall_class1
[1] 1

> recall_class2
[1] 0.9411765

> recall_class3
[1] 1

> recall_class4
[1] 0.862069
```

For 'polynomial' kernel type, the accuracy is 20.8%. The confusion matrix is as follows:

```
Prediction                 comp.sys.ibm.pc.hardware sci.electronics talk.politics.guns
  comp.sys.ibm.pc.hardware                        0               0                 33
  sci.electronics                                 0               1                 33
  talk.politics.guns                              0               0                 24
  talk.politics.misc                              0               0                 29
                          Reference
Prediction                 talk.politics.misc
  comp.sys.ibm.pc.hardware                  0
  sci.electronics                           0
  talk.politics.guns                        0
  talk.politics.misc                        0
```

The Recall values for the four classes by this classifier are:

```
> recall_class1
[1] 0
> recall_class2
[1] 0.02941176
> recall_class3
[1] 1
> recall_class4
[1] 0
```

For 'radial' kernel type, the accuracy is 78.3%. The confusion matrix is as follows:

```
                          Reference
Prediction                 comp.sys.ibm.pc.hardware sci.electronics talk.politics.guns
  comp.sys.ibm.pc.hardware                       31               2                  0
  sci.electronics                                10              23                  0
  talk.politics.guns                              0               1                 23
  talk.politics.misc                              0               3                  9
                          Reference
Prediction                 talk.politics.misc
  comp.sys.ibm.pc.hardware                  0
  sci.electronics                           1
  talk.politics.guns                        0
  talk.politics.misc                       17
```

The Recall values for the four classes by this classifier are:

```
> recall_class1
[1] 0.9393939

> recall_class2
[1] 0.6764706

> recall_class3
[1] 0.9583333

> recall_class4
[1] 0.5862069
```

7. Feature Selection by Recursive Feature Elimination – By performing this, we obtain a list of important variables that would be useful in our classification model. This list consists of 400 variables.

So, I remove all the variables from our bag of words except for these 400 variables and our factor response variables, making it a bag of words with 401 columns.

We then perform all the five classification techniques on this bag of words and see a change in the accuracies of our models. I am not attaching all the confusion matrices and recall values, but here is a comparison of all these accuracies with the ones we received for the preprocessed data.

```
> cbind(accuracy_nb_2, accuracy_nb_3)
         accuracy_nb_2 accuracy_nb_3
Accuracy     0.2916667     0.3333333

> cbind(accuracy_knn_2, accuracy_knn_3)
    accuracy_knn_2 accuracy_knn_3
acc      0.6833333      0.7333333

> cbind(accuracy_rf_2, accuracy_rf_2)
     accuracy_rf_2 accuracy_rf_2
[1,]     0.9666667     0.9666667

> cbind(accuracy_dctree, accuracy_dctree_2)
    accuracy_dctree accuracy_dctree_2
acc            0.95              0.95

> cbind(accuracy_svmlinear, accuracy_svmlinear_2)
         accuracy_svmlinear accuracy_svmlinear_2
Accuracy               0.95                0.925

> cbind(accuracy_svmpolynomial, accuracy_svmpolynomial_2)
         accuracy_svmpolynomial accuracy_svmpolynomial_2
Accuracy              0.2083333                     0.375

> cbind(accuracy_svmradial, accuracy_svmradial_2)
```

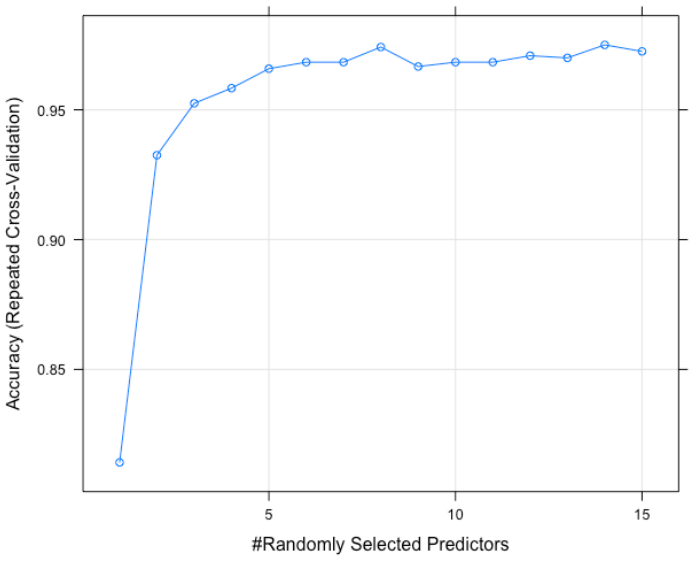
```
          accuracy_svmradial accuracy_svmradial_2
Accuracy           0.7833333            0.8916667
```

As we can see above, the accuracy has majorly increased (while remaining the same in case of decision tree and random forest) and falling slightly with linear kernel type support vector machine.

8. Hyper parameter tuning – The final thing we do in this assignment is applying the hyper parameter tuning. The output of that is as follows.

```
Random Forest

400 samples
400 predictors
  4 classes: 'comp.sys.ibm.pc.hardware', 'sci.electronics', 'talk.politics.guns',
'talk.politics.misc'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 360, 360, 360, 360, 360, 360, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   1    0.8141667  0.7522222
   2    0.9325000  0.9100000
   3    0.9525000  0.9366667
   4    0.9583333  0.9444444
   5    0.9658333  0.9544444
   6    0.9683333  0.9577778
   7    0.9683333  0.9577778
   8    0.9741667  0.9655556
   9    0.9666667  0.9555556
  10    0.9683333  0.9577778
  11    0.9683333  0.9577778
  12    0.9708333  0.9611111
  13    0.9700000  0.9600000
  14    0.9750000  0.9666667
  15    0.9725000  0.9633333

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 14.
```

The plot for the accuracies at each iteration is as follows



Conclusion

I would say that random forest is the best classifier for this dataset considering the accuracy it is giving with and without preprocessing. The classification would also be well performed by decision tree, but the consistency of random forest throughout the analysis is commendable and this is the technique I would recommend for this dataset.

Also, the data without preprocessing is giving very low accuracies with all the basic models which would lead to believe that the words extracted out in raw format for this dataset are absolutely not the way to proceed. Preprocessing done by was definitely necessary.
