Data Mining Project

Name: **Digant Kumar**
Student No: **119220141**
Lecturer: Dr. Alejandro Arbelaez

Declaration:

By submitting this assignment. I agree to the following:

"I have read and understand the UCC academic policy on plagiarism and I agree to the requirements set out thereby in relation to plagiarism and referencing. I confirm that I have referenced and acknowledged properly all sources used in preparation of this assignment. I declare that this assignment is entirely my own work based on my personal study. I further declare that I have not engaged the services of another to either assist me in, or complete this assignment"

Objectives:

Implementing machine learning techniques namely KNN, Naïve Bayes, Random Forest, Decision Trees and SVM on the Newsgroup dataset, performing feature selection to get the most important features, and optimizing the machine learning tools using the same features to yield better results on the testing set.

EXPLORATION OF THE DATASET:

The table below shows the difference in the top 200 words before and after filtering the tokens by length. The left hand-side shows the **first 10** most popular vector of words and its frequency **before filtering by length**, whereas the **right-hand** side of the table shows the **top 10** most occurring words **after filtering by length**. Only one word**('that')** from the left hand-side has made it into the filtered words section, rest all are filtered as the were of length less than 3.

	words	Freq	filtered_words	Freq
1	the	5212	that	1548
2	to	3021	have	824
3	of	2351	with	637
4	a	2309	this	512
5	and	1859	they	459
6	is	1566	Subject:	454
7	that	1548	From:	409
8	I	1482	Date:	407
9	in	1267	Lines:	405
10	>	1200	Newsgroups:	405

BASIC EVALUATION:

Results from KNN, Random Forest and Naïve Bayes:

Confusion Matrix:

In KNN, the majority of the test set are predicting class 2.
In Random forest, all the classes are being predicted equally, and hence is performing better than Naïve Bayes and KNN.
In Naïve Bayes, most of the test set variables are getting predicted in class 1

```
> knn.confmat$table
          Reference
Prediction  1  2  3  4
         1  9  4  5  5
         2 14 18 12 11
         3  3  6 12  5
         4  3  2  3  8
```

```
> rf.confmat$table
          Reference
Prediction  1  2  3  4
         1 28  3  1  1
         2  1 25  2  4
         3  0  0 26  3
         4  0  2  3 21
```

```
> nb.confmat$table
          Reference
Prediction  1  2  3  4
         1 25  1  0  0
         2 14 14  1  0
         3 20 10  1  1
         4 24  7  0  2
```

Accuracy:

```
> knn.confmat$overall[1]
 Accuracy
0.3916667
```

```
> rf.confmat$overall[1]
 Accuracy
0.8333333
```

```
> nb.confmat$overall[1]
Accuracy
    0.35
```

Precision:

	KNN	*Ran. Forest*	*Naïve Bayes*
	Precision	Precision	Precision
Class: 1	0.3913043	0.8484848	0.96153846
Class: 2	0.3272727	0.7812500	0.48275862
Class: 3	0.4615385	0.8965517	0.03125000
Class: 4	0.5000000	0.8076923	0.06060606

Recall:

	KNN	*RF*	*Naïve Bayes*
	Recall	Recall	Recall
Class: 1	0.3103448	0.9655172	0.3012048
Class: 2	0.6000000	0.8333333	0.4375000
Class: 3	0.3750000	0.8125000	0.5000000
Class: 4	0.2758621	0.7241379	0.6666667

F1 Score:

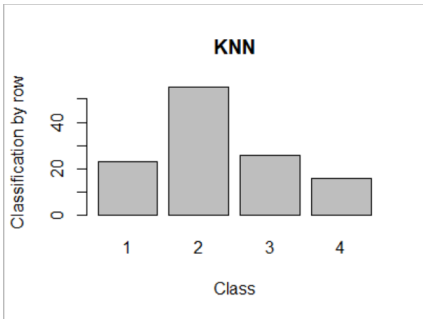
	KNN	*RF*	*Naïve Bayes*
	F1	F1	F1
Class: 1	0.3461538	0.9032258	0.45871560
Class: 2	0.4235294	0.8064516	0.45901639
Class: 3	0.4137931	0.8524590	0.05882353
Class: 4	0.3555556	0.7636364	0.11111111

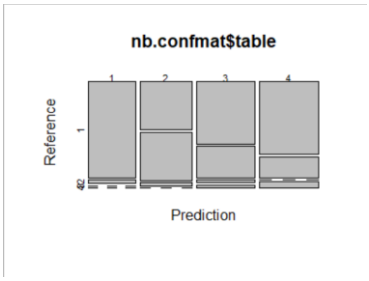
Plots:

The plot shows the confusion matrix table graphically. In KNN, Class 2 is being predicted more than 50 times and hence the big horizontal line, while all the others are being predicted equally and therefore is leading to low accuracy.

In the random forest, all the classes are being predicted almost equally, hence the higher accuracy.

In Naïve Bayes, the majority of the test set variables are predicted as being in class 1.





Conclusion from Basic Evaluation Methods:

Random Forest clearly outperforms KNN and Naïve Bayes on the raw dataset as can be seen from the accuracy and performance metrics.

ROBUST EVALUATION:

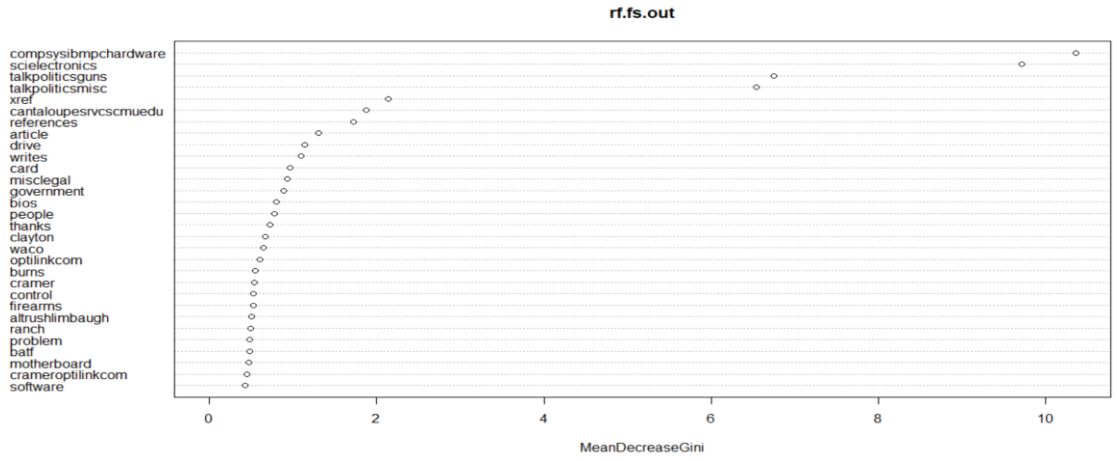
Methods employed for cleaning the dataset:

Converting all the words to lower case, removing the punctuations, numbers, stop-words, white-spaces and words of length less than 3. Creating bag of words of this new updated dataset.

Feature Selection:

Using **Random Forest** variable importance for feature selection. The features are selected in the decreasing order of mean impurity over all the trees. Taking the **first 1000 most significant features** as selected and defining it as the new dataset.

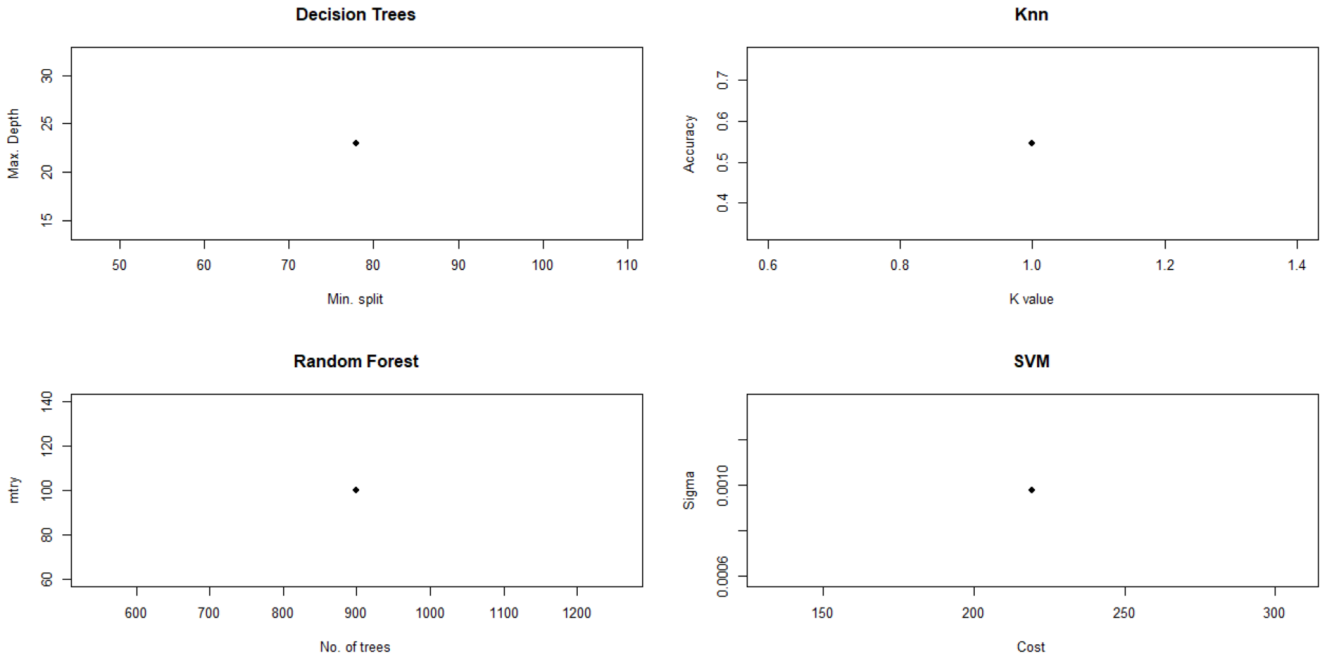
The following plot shows the 30 most important features selected by Random Forest.



*The **top 4 most important features** in our dataset are the names of the folders within our Newsgroup folders.*

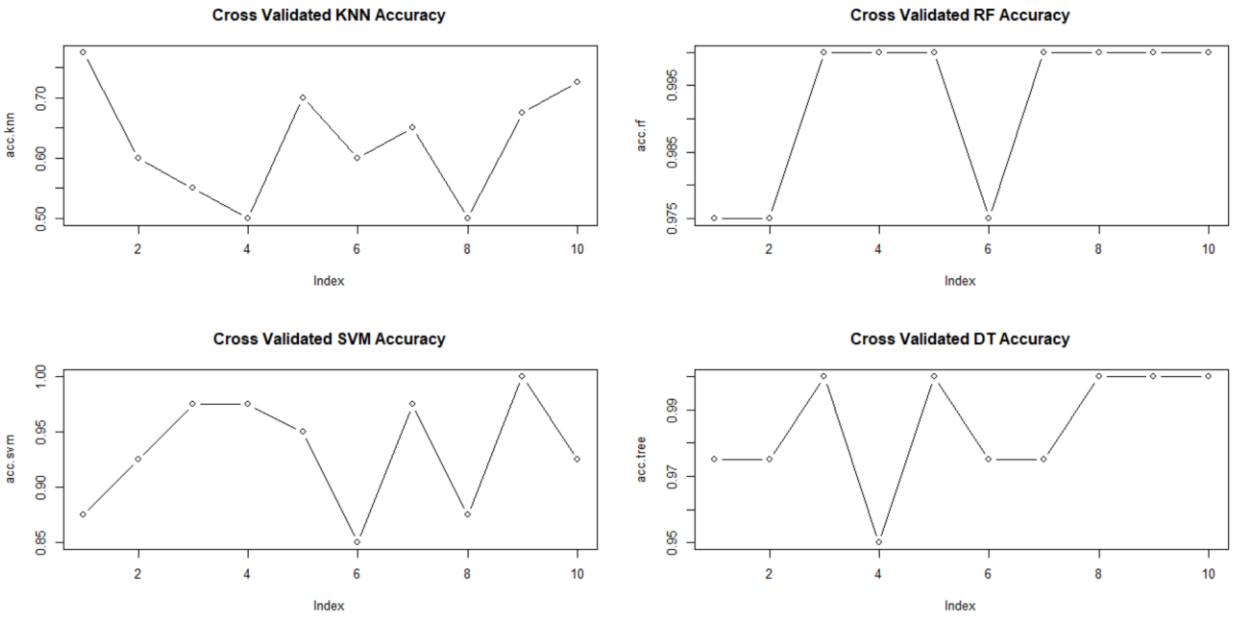
Hyperparameter Tuning:

Implementing **Exhaustive Grid Search** for Hyperparameter tuning for all the methods.
Optimal Parameters for Trees/Parameters Optimize: Min. split = 78, max. depth = 23, cp = 0.005
Optimal Parameters for KNN/Parameters Optimized = K = 1
Optimal Parameters for Random Forest/Parameters Optimized = No. of trees = 900, mtry = 100
Optimal Parameters for SVM/Parameters Optimized = Cost = 219, sigma = 0.000977



Cross Validation:

Performing 10-fold cross validation on the test set. The plot below shows the accuracy for all the 10 iterations in all the methods. Random Forest continues to perform best with a mean accuracy of 99.25%, closely followed by decision trees at 98.5%, followed by SVM at 93.25% and KNN at 62.75%.



Hold-out:

Diving the dataset into 70% training and 30% testing set and then making predictions on the test set using the optimal parameters obtained from Hyperparameter tuning of the respective dataset.

Conclusions from Holdout:

Confusion Matrix:

```
> tree.confmat$table
          Reference
Prediction  1  2  3  4
         1 30  0  0  0
         2  1 34  0  0
         3  0  0 28  0
         4  0  0  1 26
```

```
> svm.confmat$table
          Reference
Prediction  1  2  3  4
         1 30  0  0  0
         2  1 33  0  1
         3  0  0 28  0
         4  0  0  1 26
```

```
> knn2.confmat$table
          Reference
Prediction  1  2  3  4
         1 12 16  0  2
         2  1 33  0  1
         3  1  5 21  1
         4  0 10  2 15
```

```
> rf2.confmat$table
          Reference
Prediction  1  2  3  4
         1 30  0  0  0
         2  1 34  0  0
         3  0  0 28  0
         4  0  0  0 27
```

Accuracy:

```
> cbind(tree.confmat$overall[1], svm.confmat$overall[1], knn2.confmat$overall[1],
+       rf2.confmat$overall[1]) ## Comparing the final accuracies
             [,1]  [,2]  [,3]      [,4]
Accuracy 0.9833333 0.975 0.675 0.9916667
```

Precision:

	Trees	*SVM*	*KNN*	*RF*
	Precision	Precision	Precision	Precision
Class: 1	1.0000000	1.0000000	0.4000000	1.0000000
Class: 2	0.9714286	0.9428571	0.9428571	0.9714286
Class: 3	1.0000000	1.0000000	0.7500000	1.0000000
Class: 4	0.9629630	0.9629630	0.5555556	1.0000000

Recall:

	Recall	Recall	Recall	Recall
Class: 1	0.9677419	0.9677419	0.8571429	0.9677419
Class: 2	1.0000000	1.0000000	0.5156250	1.0000000
Class: 3	0.9655172	0.9655172	0.9130435	1.0000000
Class: 4	1.0000000	0.9629630	0.7894737	1.0000000

F1 Score:

	F1	F1	F1	F1
Class: 1	0.9836066	0.9836066	0.5454545	0.9836066
Class: 2	0.9855072	0.9705882	0.6666667	0.9855072
Class: 3	0.9824561	0.9824561	0.8235294	1.0000000
Class: 4	0.9811321	0.9629630	0.6521739	1.0000000

Impact of cleaning the data:

Comparing the knn, and random forest accuracy before and after cleaning the dataset, we can see that cleaning the dataset(improved bag of words, hyper-parameter tuning) has improved the classification rate massively by around 20% for knn, and of random forest by around 12%.

Difference between Basic Evaluation and Robust Evaluation:

Under the basic evaluation, we have taken the entire bag of words formed from the Newsgroup data, and, made predictions on this dataset. The accuracy obtained from the basic evaluation techniques are 38% for knn, 83% for random forest and 35% for naïve bayes.
Under the robust evaluation, we have first cleaned the dataset, by removing stop-words, punctuations, numbers, words of length 3, and converted all the words in lower case. Further, we have performed feature selection using Random Forest and selected the top 1000 important features, which now becomes our updated dataset. Applying machine learning techniques on this dataset yielded a much improved accuracy in both knn and random forest. We further applied, decision trees and SVM on the updated dataset.

Best Model:

From the accuracy in the holdout and cross-validation, we can see that **Random Forest and Decision trees** do the best job in classifying the words on this dataset while KNN does the worst job of classification this dataset.
However, most of the models are prone to overfitting as the top 4 most important variables are the folder names, and hence might not perform best on an unseen data.