SAINT MARY'S UNIVERSITY SINCE 1802

One University. One World. Yours.

Data and Text Mining

MCDA5580

Team Members:

NAME	A#	Email
Manoj Bandaru	A00433174	Manoj.bandaru@smu.ca
Aditya Tandon	A00432835	Aditya.Tandon@smu.ca
Khagesh Pandya	A00431429	Khagesh.pandya@smu.ca

Table of Contents

1. Executive Summary:

The main aim of this report is performing classification on a data which is about car with certain feature values. Based on that, cars are classified in different categories to identify whether car should buy or not. In this report, analysis is performed to classify car category for the purchase. For the classification, we used decision tree and random forest and the most significant variables are been picked.

2. Data Summary:

The Data is about car features and it contains seven columns.

- **Price:** low, med, high, vhigh
- **Maintenance:** low, med, high, vhigh
- **Doors:** 2,3,4
- **Safety:** low, med, high
- **Seats:** 2, 4, more
- **Storage:** small, med, big
- **Should Buy:** unacc, acc, good, vgood

From this we picked up dependent and Independent Features which are:

Dependent Features: should Buy

Independent Features: price, Maintenance, Doors, Safety, Seats, Storage, should Buy

3. Method:

Decision tree is supervised machine learning algorithm. For the analysis, data is split into two different categories.

1 Train Set: 75%

2. Test Set: 25%

4.Decision Tree:

Decision tree is tree-based algorithm which takes occurrence of certain values into account and forms rules to classify unseen data and predict the output.

There are multiple algorithms which can be used to create decision tree. Most widely used algorithms for decision tree are[1]

1. ID3 (Entropy measurement)

2. Gini Index

4.1 Classification using ID3 Without Parameter Tuning:

The Initial step in decision tree has been done based entropy measurement and the below are results of it:

Rules:

```
1) root 1296 384 unacc (0.22299383 0.03703704 0.70370370 0.03626543)
   2) safetylow< 0.5 864 384 unacc (0.33449074 0.05555556 0.55555556 0.05439815)
     4) seats4>=0.5 284 138 acc (0.51408451 0.09507042 0.30985915 0.08098592) *
     5) seats4< 0.5 580 188 unacc (0.24655172 0.03620690 0.67586207 0.04137931)
      10) seatsmore>=0.5 288 145 acc (0.49652778 0.07291667 0.34722222 0.0833333)
        20) maintenancevhigh< 0.5 210  93 acc (0.55714286 0.10000000 0.22857143
        0.11428571) *
        21) maintenancevhigh>=0.5 78  26 unacc (0.33333333 0.00000000 0.66666667
       0.0000000) *
      11) seatsmore< 0.5 292   0 unacc (0.00000000 0.00000000 1.00000000 0.000) *
3) safetylow>=0.5 432   0 unacc (0.00000000 0.00000000 1.00000000 0.00000000) *
```

```
          Reference
Prediction acc good unacc vgood
     acc    85   21    54    18
     good    0    0     0     0
     unacc  10    0   244     0
     vgood   0    0     0     0

Overall Statistics

               Accuracy : 0.7616
                 95% CI : (0.7185, 0.801)
    No Information Rate : 0.6898
    P-Value [Acc > NIR] : 0.0005934

                  Kappa : 0.5267
```

Figure 1 Accuracy of ID3 Decision tree without tuning

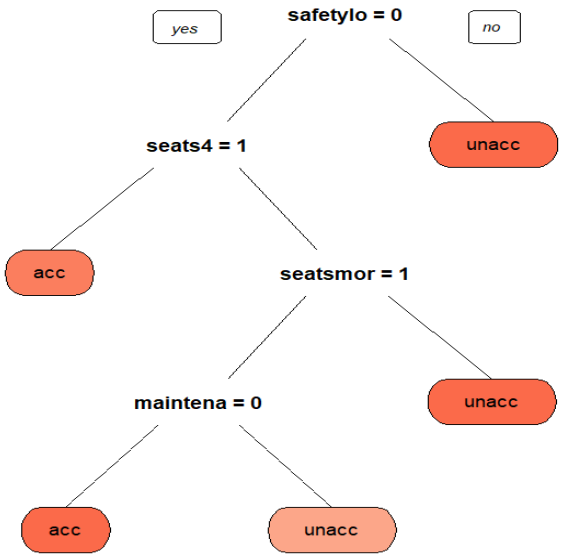

Figure 2 ID3 Decision Tree without tuning

Above results are generated without parameter tuning. To improve accuracy and model efficiency parameter tuning is required. In decision tree there are certain methods can be used to do parameter tuning.

Grid Search: Automatic Grid:

In this caret package automatically tunes model to improve performance. In this method it specifies number of different values for each algorithm parameter. This can be achieved by specifying value for tune Length.

Grid Search: Manual Grid:

In this we need to manually specify grid value. After passing grid object to tunGrid parameter in function it uses all possible combinations to produce best results.

4.1.1 Classification using ID3 After Parameter Tuning:

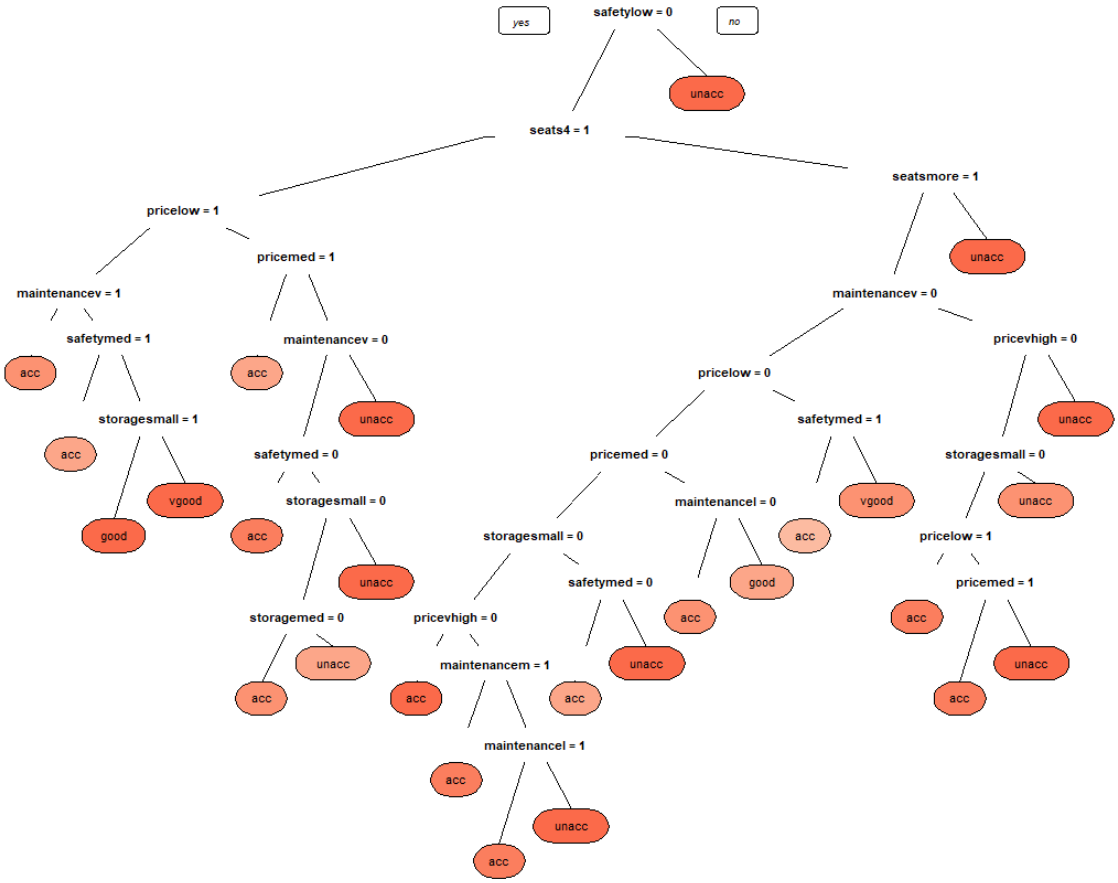

Figure 3 ID3 After parameter Tuning

```
          Reference
Prediction acc good unacc vgood
     acc    86    9    19    10
     good    1    7     0     2
     unacc   7    0   279     0
     vgood   1    5     0     6

Overall Statistics

               Accuracy : 0.875
                 95% CI : (0.8401, 0.9047)
    No Information Rate : 0.6898
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7384
```

Figure 4 Accuracy table for ID3 after tuning

- After tuning parameters, accuracy is increased by 10% with confidence level 95%

4.2 Decision Tree using Gini Index Without Tuning:

- To avoid overfitting in decision tree minsplit value can be used to make model more generalize. Minsplit cut downs decision tree when tree grows to that in size. When in accuracy there is significant drop in accuracy that value can be considered as final min split value[3].

```
          Reference
Prediction acc good unacc vgood
     acc    90    0    13     0
     good    4   16     1     0
     unacc   0    0   284     0
     vgood   1    5     0    18

Overall Statistics

               Accuracy : 0.9444
                 95% CI : (0.9185, 0.9641)
    No Information Rate : 0.6898
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8865
```

Figure 6 Decision Tree with Gini Index

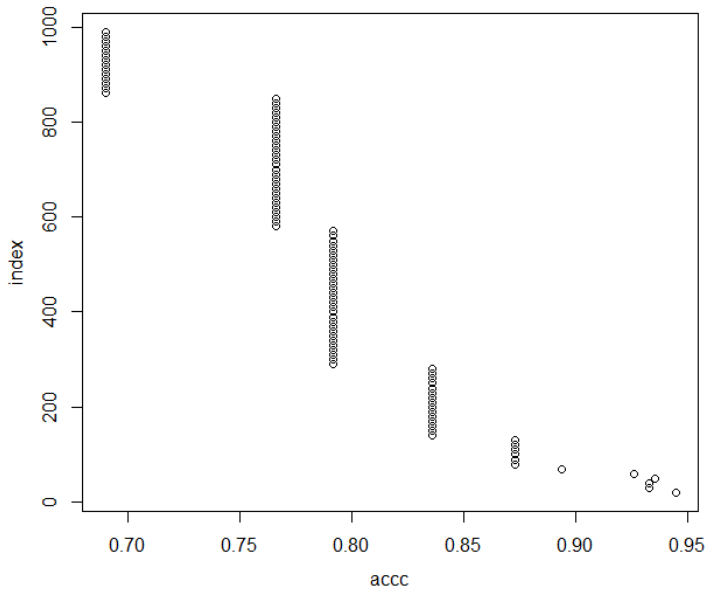

Figure 5 Minsplit node vs Accuracy

```
          Reference
Prediction acc good unacc vgood
     acc    77    0    14     0
     good    8   16     1     0
     unacc   9    0   283     0
     vgood   1    5     0    18

Overall Statistics

                Accuracy : 0.912
                  95% CI : (0.8813, 0.937)
    No Information Rate : 0.6898
    P-Value [Acc > NIR] : < 2.2e-16

                   Kappa : 0.8176
```

Figure 7 Decision Tree accuracy table based on gini index

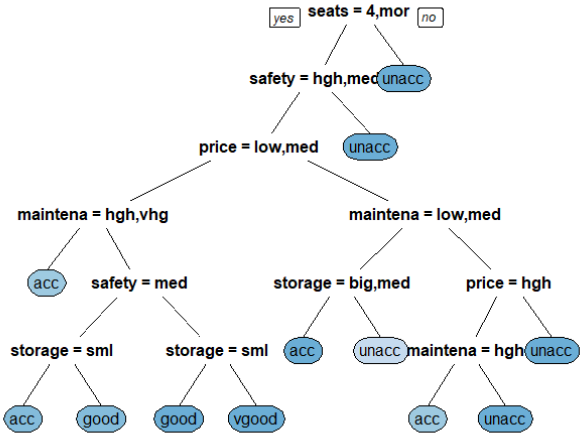

Figure 8 Decision Tree based on gini index

4.4 Decision Tree after Parameter Tuning:

After taking minsplit=85 this is the result which indicates the model accuracy is decreased, and kappa value also get decreased.

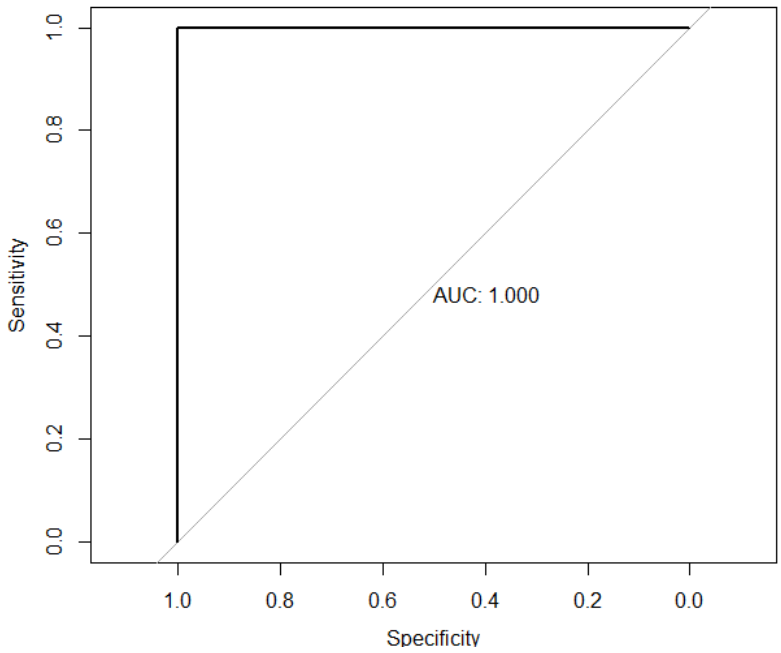

Figure 9 AUC ROC Curve

- Area under the curve of ROC graph is measurement how well model can identify two categories.

- As this is multiclass classification it multiclass.roc function is used to plot overall roc plot for all features.

- **AUC Interpretation:**

  0.0<AUC<0.5:- Wrong classification.
  AUC=0.5: Not able to classify.
  0.5<AUC<1.0: Able to classify.
  AUC=1.0 : Perfectly identify categories.

5. Random Forest:

Random Forest is an ensemble method which creates many decision trees under the hood and then takes an average value if it is quantitative or it takes votes if it is qualitative. It trains a lot of weak learners while training the data and then combine their result to build the exact set of rules and thus, is a much better process than the Decision Tree.

Just like we can see the rules in a Decision Tree, and can visualize it, we can't do the same in the case of random forest as there are multiple decision trees that are made and there is no way we can visualize it. So, the question comes, how can we extract what sort of features were the most important in determining the class.

For that, random forest provides us with feature importance graph, which we can plot and see, which feature has been given the most importance.

Random forest is a strong bagging algorithm which achieves better performance and higher accuracy than decision tree.

5.1 Random Forest Without Tuning:

The most important part of the machine learning algorithms is to tweak and play with the parameters, also known as model's hyper parameters.

Without tuning the parameters of the Random Forest algorithm, we get a pretty good result but not the best which is 95.6.

The default parameters were,

- mtry = 6.
- ntree = Remains 500 by default.

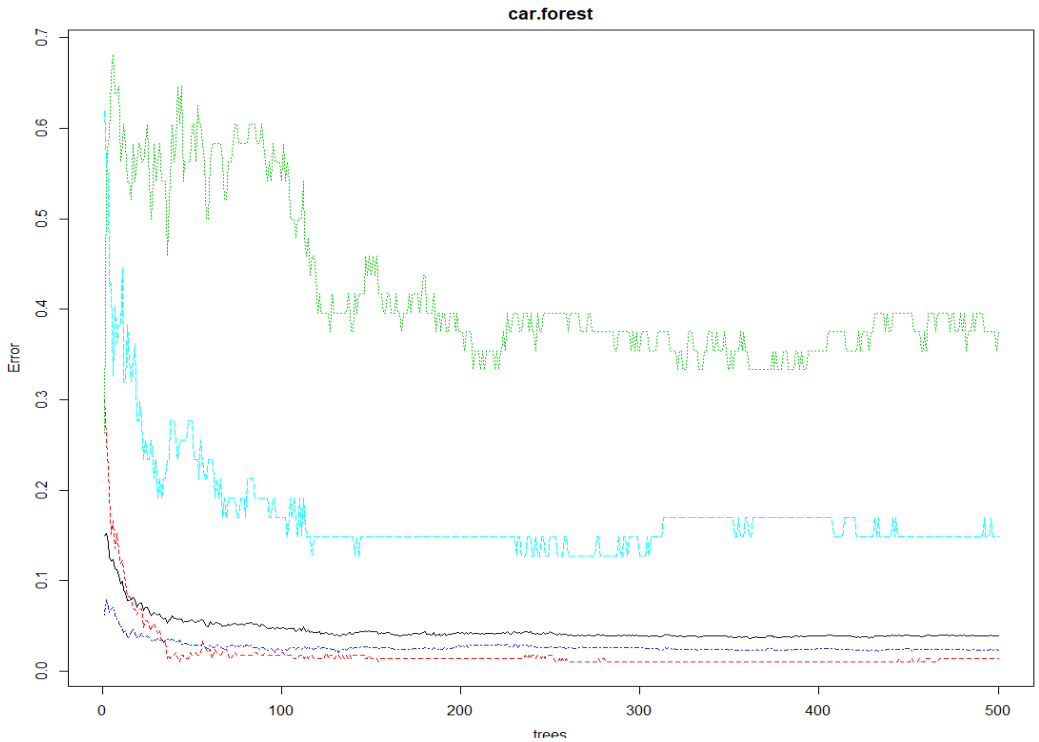

Figure 10 Random Forest Accuracy without Tuning

- It is using all the six parameters and is giving us about 96.06% accuracy.
- Higher kappa value indicates perfect agreement between features.

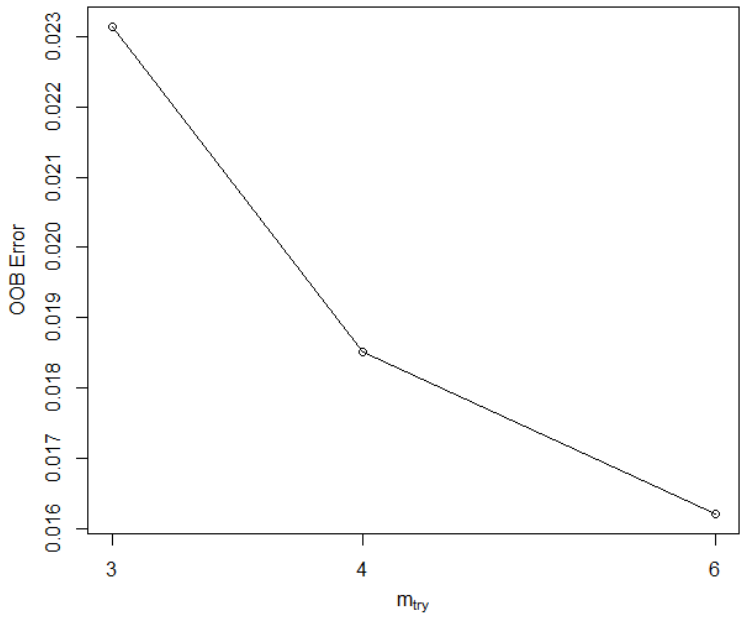

Figure 11 OOB Error Vs Number Of trees

- Black line indicates overall error while colored lines indicates error values of different predicted categories. OOB rate withour tuning is 1.62%.
- It is evident that after 100 trees, there is no significant change in error term.
- This graph is for **Error vs mtry** value mtry 6 indicates best model.

```
1728 samples
   6 predictor
   4 classes: 'acc', 'good', 'unacc', 'vgood'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 1556, 1556, 1555, 1554, 1556, 1554, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.7313161  0.1663247
   8    0.9340749  0.8564113
  15    0.9269345  0.8386582

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 8.
```

Figure 12 Random Forest Accuracy Result

5.2 K Fold Cross Validation in Random Forest:

```
          Reference
Prediction acc good unacc vgood
     acc    95    0     1     0
     good    0   21     0     0
     unacc   0    0   297     0
     vgood   0    0     0    18

Overall Statistics

               Accuracy : 0.9977
                 95% CI : (0.9872, 0.9999)
    No Information Rate : 0.6898
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9951
```

Figure 13 Random Forest With k fold Cross Validation

Right side image is the model after performing 10-fold validation on entire data and result provides best models with accuracy and kappa values[3].

- Left side image is confusion matrix result on test dataset.
- After 10 fold cross validation the accuray is **99.77%.**

6. Feature Importance:

	Overall
seatsmore	100.00
safetylow	89.11
pricevhigh	44.32
maintenancevhigh	44.32
seats4	39.94
safetymed	27.10
pricelow	26.64
maintenancemed	0.00
storagesmall	0.00
pricemed	0.00
doors3	0.00
storagemed	0.00
doors4	0.00
doors5more	0.00
maintenancelow	0.00

Figure 14 Decision Tree Feature Importance

	Overall
safetylow	100.000
seatsmore	60.140
seats4	53.225
safetymed	28.411
storagesmall	28.392
pricelow	27.621
pricemed	24.403
maintenancelow	24.160
maintenancemed	18.921
maintenancevhigh	17.902
pricevhigh	9.130
storagemed	2.812
doors4	1.526
doors5more	1.018
doors3	0.000

Figure 15 Feature Importance for Random Forest

- From above features, seats have the highest importance in decision tree on the other side in random forest its safety which has the highest importance. So, random forest can capture underlying reason for the car purchase.
- Another observation is random forest weight for the feature is more diverse it is considering up to 1.00% of overall score which also make random forest more efficient than decision tree.

Conclusion:

- We used Decision tree and random forest for classifiiaion problem and came up with certain solutions.
- We split the data into two parts Test and train.
- By using decision tree algorithm, we attained certain accuracy using ID3(Entropy Measurement), also we performed some analysis based on Gini Index and attained certain accuracy.
- Further, we moved and performed random forest algorithm. Random Forest is an ensemble method which creates many decision trees under the hood and then takes an average value if it is quantitative or it takes votes if it is qualitative.
- By performing random forest, we achieved the feature importance which are safety and seats.

References:

[1]Dataaspirant. (2017). *Decision Tree Classifier implementation in R*. [online] Available at: https://dataaspirant.com/2017/02/03/decision-tree-classifier-implementation-in-r/ [Accessed 5 Jun. 2019].

[2]learn data science. (2018). *Visualizing a decision tree using R packages in Exploratory*. [online] Available at: https://blog.exploratory.io/visualizing-a-decision-tree-using-r-packages-in-explortory-b26d4cb5e71f [Accessed 5 Jun. 2019].

[3]Brownlee, J. (2018). *A Gentle Introduction to k-fold Cross-Validation*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/k-fold-cross-validation/ [Accessed 5 Jun. 2019].

[4]Brownlee, J. (2014). *Tuning Machine Learning Models Using the Caret R Package*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/tuning-machine-learning-models-using-the-caret-r-package/ [Accessed 5 Jun. 2019].

[5]Brownlee, J. (2014). *How to Estimate Model Accuracy in R Using the Caret Package*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/how-to-estimate-model-accuracy-in-r-using-the-caret-package/ [Accessed 5 Jun. 2019].