Fuel Consumption Classification Problem

using

Decision Trees

Table of Contents

1. Aim: 2
2. Data: 2
3. Data Cleaning: 3
4. Decision Tree Classifiers 4
5. Summary 10
6. R script for Decision Tree Classifier 0

1. Aim:

The aim behind this project is to build a simple decision tree classifier on a fuel consumption dataset. Ultimately, to be able to predict the fuel consumption of a car based on certain previously trained (classified) features.

2. Data:

The data used shows the fuel consumption of vehicles.

Data source: https://open.canada.ca/data/dataset/98f1a129-f628-4ce4-b24d-6f16bf24dd64

3. Data Cleaning:

Before Cleaning

MODEL YEAR	MAKE	MODEL # = high output engir	VEHICLE C	ENGINE SI (L)	CYLINDER	TRANSMIS	FUEL TYPE	FUEL CONSUMPTION* CITY (L/10	HWY (L/1(	COMB (L/	COMB (mi	CO2 EMISS (g/km)
2014	ACURA	ILX	COMPACT	2	4	AS5	Z	9.9	6.7	8.5	33	196
2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	7.7	9.6	29	221
2014	ACURA	ILX HYBRII	COMPACT	1.5	4	AV7	Z	6	5.8	5.9	48	136
2014	ACURA	MDX 4WD	SUV - SMA	3.5	6	AS6	Z	12.7	9.1	11.1	25	255
2014	ACURA	RDX AWD	SUV - SMA	3.5	6	AS6	Z	12.1	8.7	10.6	27	244

After Cleaning

	YR	BRAND	MODEL	CLASS	GEARS	TRANSMISSION	AuthomaticOrManual	ENG	CYLINDERS	FUEL	consume
0	2014	ACURA	ILX	COMPACT	5	AS	A	2.0	4	Z	0
1	2014	ACURA	ILX	COMPACT	6	M	M	2.4	4	Z	1
2	2014	ACURA	ILXHYBRID	COMPACT	7	AV	A	1.5	4	Z	0
3	2014	ACURA	MDX4WD	SUVSMALL	6	AS	A	3.5	6	Z	1
4	2014	ACURA	RDXAWD	SUVSMALL	6	AS	A	3.5	6	Z	1

Features removed –

1. **TRANS** – This feature had valuable information that, thought could best serve our goals by getting segregated. Features *TRANSMISSION*, *GEARS*, and *AutomaticOrManual* were created from this feature.
2. **CITY_L**
3. **HWY_L**
4. **CITY_MI**
5. **HWY_MI**
6. **CO2**

} Fuel Consumption metrics were removed

Features added –

1. **GEARS**
2. **TRANSMISSION**
3. **AutomaticOrManual**
4. **Consume (for classification purpose)**

} All these come from the *Trans* feature of original dataset

4. Decision Tree Classifiers

In-order to achieve the best possible accuracy from the simplistic decision tree classifier built here, I decided to try out multiple epochs or instances of the same classifier. My aim here was to make each instance as different as possible from the previous one.

The modifications I made at each epoch were –

1. Choose different feature sets to drill down on the best possible combination of features
2. Choose different values for the afore-mentioned parameters/controls

Here are the steps I went through in-order to build varied classifiers–

Step 1. Check for datatypes:

YR	BRAND	MODEL	CLASS	GEARS	TRANSMISSION	AuthomaticOrMa	ENG	CYLINDERS	FUEL	consume
int	Factor	Factor	Factor	Factor	Factor	Factor	num	int	Factor	int

In actual "GEARS" is integer and "consume" is factor. So, changed the datatypes.

Step2. Identify important variables for model building:

Based on the description, I exclude "YR", "BRAND" and "MODEL" feature sets due to their limited contribution to our model.

Step3. Build initial decision tree and identify variables useful in classification

CLASS	GEARS	TRANSMISSION	AuthomaticOrMa	ENG	CYLINDERS	FUEL	consume
Factor	int	Factor	Factor	num	int	Factor	Factor

I used the **rpart** package in R to build decision tree classifiers.

Based on the decision tree (fintree1 in appendix), the most important independent variables from decision tree are (Fig: 1): **ENG, CLASS, FUEL, CYLINDERS, GEARS**

Step 4. Identify best control parameters

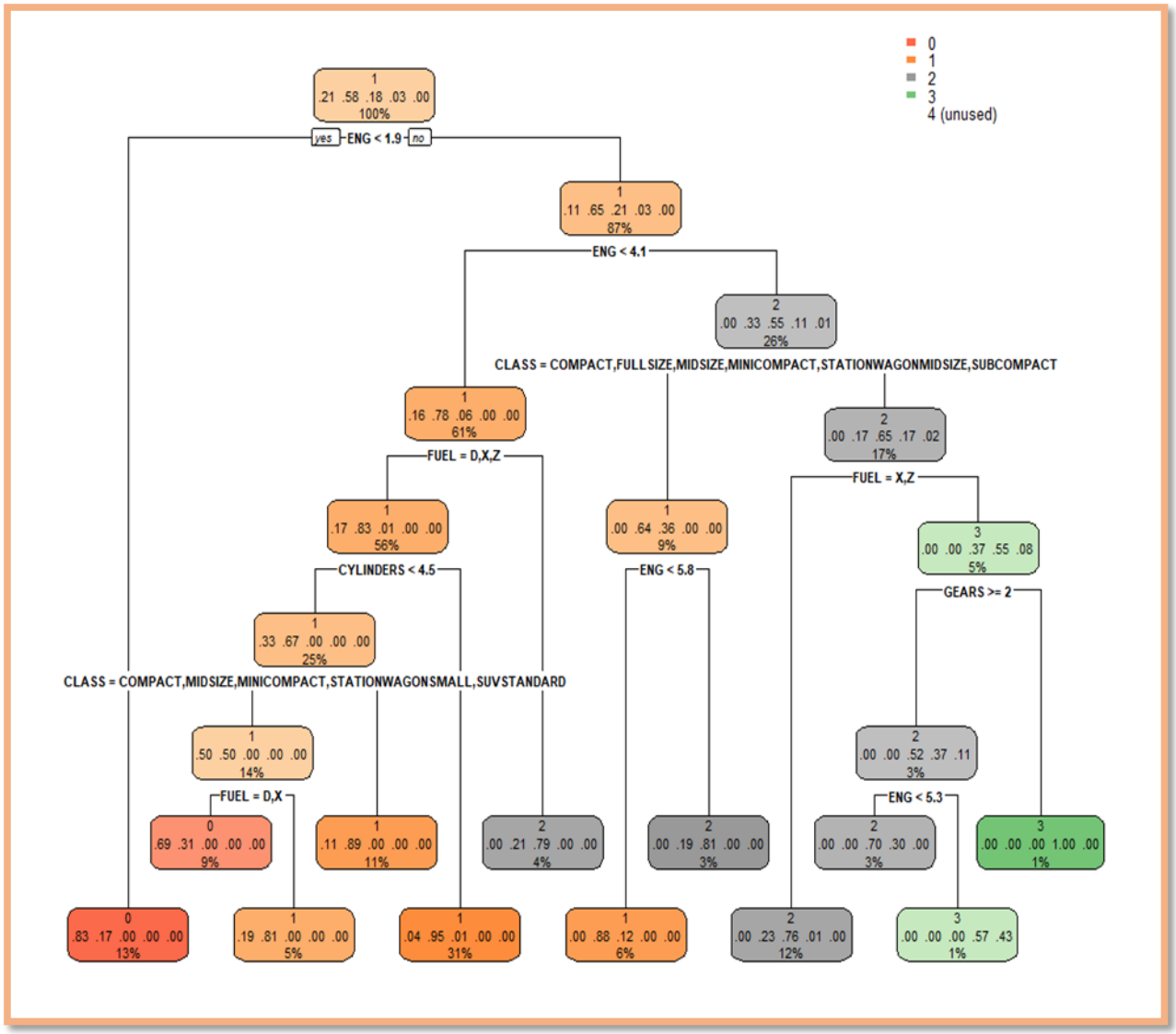
Control Parameter 1: MINSPLIT & MINBUCKET

Min Split: The minimum number of values in a node that must exist before a split is attempted. In other words, if the node has two members and the minimum split is set to 5, the node will become terminal, that is, no split will be attempted.

Min Bucket: The minimum number of entities allowed in any leaf of the tree. The default number is one-third of the value specified for Min Split.

Reference:
https://infocenter.informationbuilders.com/wf80/index.jsp?topic=%2Fpubdocs%2FRStat16%2Fsource%2Ftopic47.htm

Fig: 1 Identifying Important Variables



Created several decision trees while taking important variables in the above step and observed the effect of minsplit and minbucket on accuracy of the model.

Observations

Min Split	Min Bucket	Accuracy
1	Default: 0.33 (~`1)	86.47%
10	Default: 3.33 (~4)	86.47%
30	Default: 10	85.34%
40	25	85.71%
50	**Default: 16.66(~17)**	**85.71%**
80	Default: 26.66(~27)	84.59%
100	Default: 33.33(~34)	85.59%
200	Default: 66.66(~67)	77.82%

Based on above table, I observed that max accuracy is 86.47% comes at minsplit of 1, which is not ideal as it will create too many subtrees and add complexity in creating decision rules. Accuracy decreases till minsplit =30.

Going further, the next best accuracy is 85.71% comes at minsplit of 50. Post minsplit of 50, accuracy starts decreasing again. High values of minsplit are not desirable as well as it wont create good enough rules for classification.

Therefore, I fixed minplit at 50. Minbucket is default (minsplit/3 = 16.66 in this case)

Control Parameter 2: Complexity:

Complexity: Complexity is used to establish a control level that determines whether a split contributes to a better model fit. Any split that increases the model fit by a factor greater than the defined complexity factor is attempted. Default is 0.01.

Reference:
https://infocenter.informationbuilders.com/wf80/index.jsp?topic=%2Fpubdocs%2FRStat16%2Fsource%2Ftopic47.htm

Created several decision trees while taking important variables from step 3 and observed the effect of complexity (not clubbed with minsplit or minbucket):

Observations

Complexity	Accuracy
0.001	87.59%
0.005	87.59%
0.01	**86.09%**
0.02	85.71%
0.05	79.32%
0.1	72.56%

As low complexity will not contribute to the better model fit, discarded complexity values 0.001 and 0.005. Moreover, the model will create more complex rules, indicating overfitting.

Complexity values of 0.05 and 0.01 are poor fit as well as they are reducing the accuracy of the model.

Complexity value of 0.01 and 0.02 seems good fit. Therefore, choose 0.01 for further evaluation.

Step 5. Final Model

Data Points: 1083

Target Variable: Consume

Independent Variables: ENG, CLASS, FUEL, CYLINDERS, GEARS

Rpart Control parameters: minsplit=50 , cp (complexity) = 0.01

Test vs. Train = 25% vs. 75%

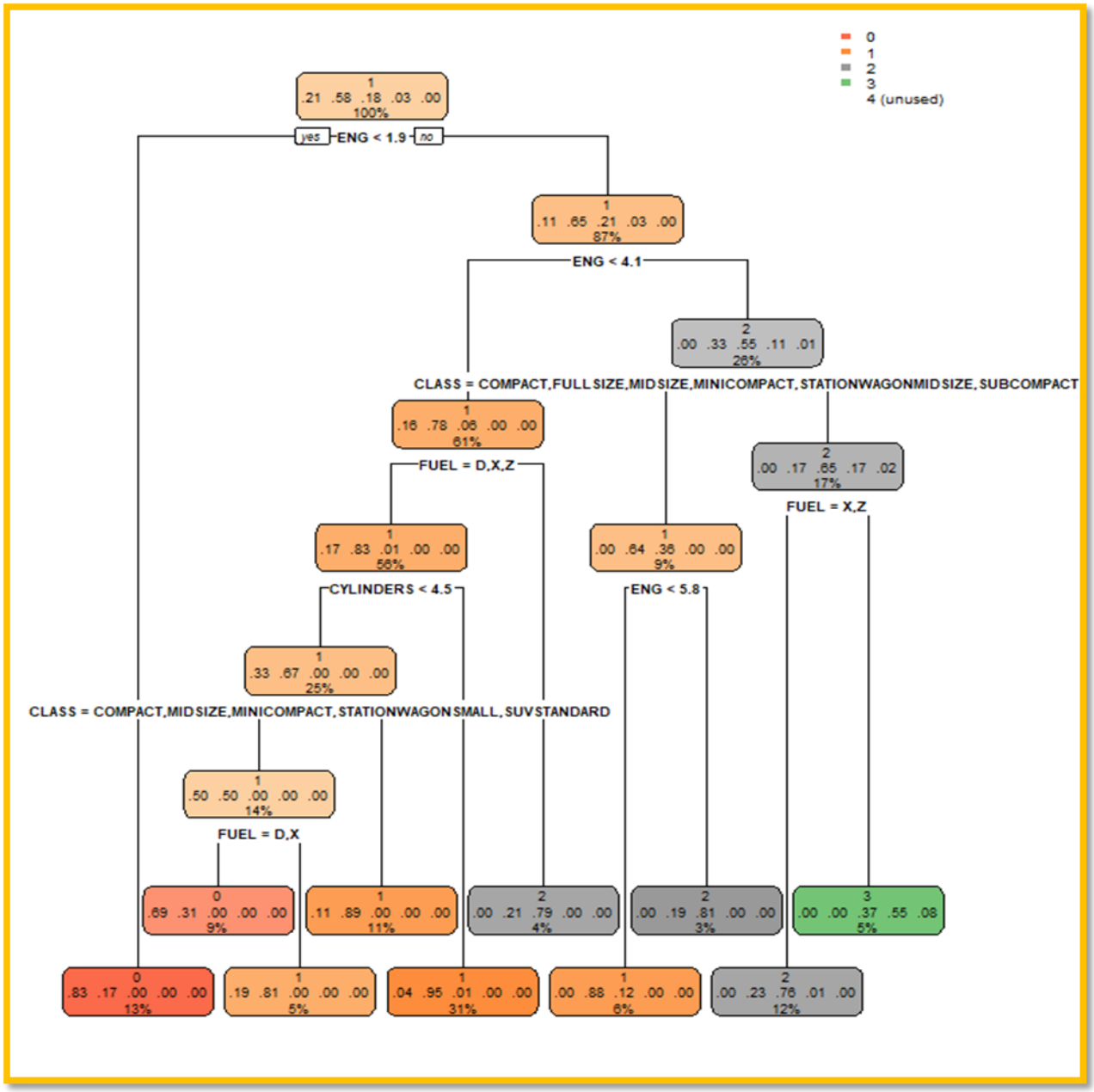
***Result** (fintree15):*

```
Confusion Matrix:

          Reference
Prediction   0   1   2   3   4
         0  58  12   0   0   0
         1   8 131   3   0   0
         2   0  12  31   0   0
         3   0   0   2   8   1
         4   0   0   0   0   0

Accuracy : 0.8571
                 95% CI : (0.8092, 0.8969)


P-Value [Acc > NIR] : < 2.2e-16

                     Class: 0 Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            0.8788   0.8452   0.8611  1.00000 0.000000
Specificity            0.9400   0.9009   0.9478  0.98837 1.000000
```

This model correctly predicts ‘Class 0’ 88%, ‘Class 1’ 85%, ‘Class 2’ 86%, ‘Class 3’ 100%. Unfortunately, it does not predict ‘Class 4’ correctly.

However, based this decision tree model should only be used for Classifying ‘Class 0’,’Class 1’ and ‘Class 2’ because from the data I see that I don’t have many data points for ‘Class 3’ (2.7%) and ‘Class4’ (0.3%).

Fig:2 Final Decision Tree


0
1
2
3
4 (unused)
1
.21 .58 .18 .03 .00
100%
yes
ENG < 1.9
no
1
.11 .65 .21 .03 .00
87%
ENG < 4.1
2
.00 .33 .55 .11 .01
26%
CLASS = COMPACT,FULL SIZE,MIDSIZE,MINICOMPACT,STATIONWAGONMIDSIZE,SUBCOMPACT
1
.16 .78 .06 .00 .00
61%
2
.00 .17 .65 .17 .02
17%
FUEL = X,Z
FUEL = D,X,Z
1
.17 .83 .01 .00 .00
56%
1
.00 .64 .36 .00 .00
9%
CYLINDERS < 4.5
ENG < 5.8
1
.33 .67 .00 .00 .00
25%
CLASS = COMPACT,MIDSIZE,MINICOMPACT,STATIONWAGONSMALL,SUVSTANDARD
1
.50 .50 .00 .00 .00
14%
FUEL = D,X
0
.69 .31 .00 .00 .00
9%
1
.11 .89 .00 .00 .00
11%
2
.00 .21 .79 .00 .00
4%
2
.00 .19 .81 .00 .00
3%
3
.00 .00 .37 .55 .08
5%
0
.83 .17 .00 .00 .00
13%
1
.19 .81 .00 .00 .00
5%
1
.04 .95 .01 .00 .00
31%
1
.00 .88 .12 .00 .00
6%
2
.00 .23 .76 .01 .00
12%

Step 6. Creating Decision Rules: With these rules, it can be correctly classified for 'Class 0', 'Class 1', 'Class 2' 85.6% time.

Decision rules created:

Class	Rules
0	Eng < 1.9
0	1. 1.9 <= Eng < 4.1; 2. Class= Compact, Mid Size, Mini Compact, Station Wagon Small, SUV Standard; 3. Fuel = D, X; 4. Cylinders < 4.5
1	1. 1.9 <= Eng < 4.1; 2. Class= Compact, Mid Size, Mini Compact, Station Wagon Small, SUV Standard; 3. Fuel = Z; 4. Cylinders < 4.5
1	1. 1.9 <= Eng < 4.1; 2. Class <> Compact, Mid Size, Mini Compact, Station Wagon Small, SUV Standard; 3. Cylinders < 4.5 4. Fuel= D, X, Z
1	1. 1.9 <= Eng < 4.1; 2. Cylinders => 4.5 3. Fuel = D, X, Z;
1	1. 4.1 <= Eng < 5. 8; 2. Class = Compact, Full Size, Mid Size, Mini Compact, Station Wagon Mid Size, Sub Compact
2	1. 1.9 <= Eng < 4.1; 2. Fuel <> D, X, Z;
2	1. Eng > 5. 8; 2. Class = Compact, Full Size, Mid Size, Mini Compact, Station Wagon Mid Size, Sub Compact
2	1. Eng => 4.1; 2. Class <> Compact, Full Size, Mid Size, Mini Compact, Station Wagon Mid Size, Sub Compact 3. Fuel Size = X, Z
3 (not recommended)	1. Eng => 4.1; 2. Class <> Compact, Full Size, Mid Size, Mini Compact, Station Wagon Mid Size, Sub Compact 3. Fuel Size <> X, Z

5. Summary

In summary, I achieved a prediction accuracy of 85.6%. Of the multiple classifiers built, I found that the one with features:

ENG, CLASS, FUEL, CYLINDERS, GEARS

and control parameters:

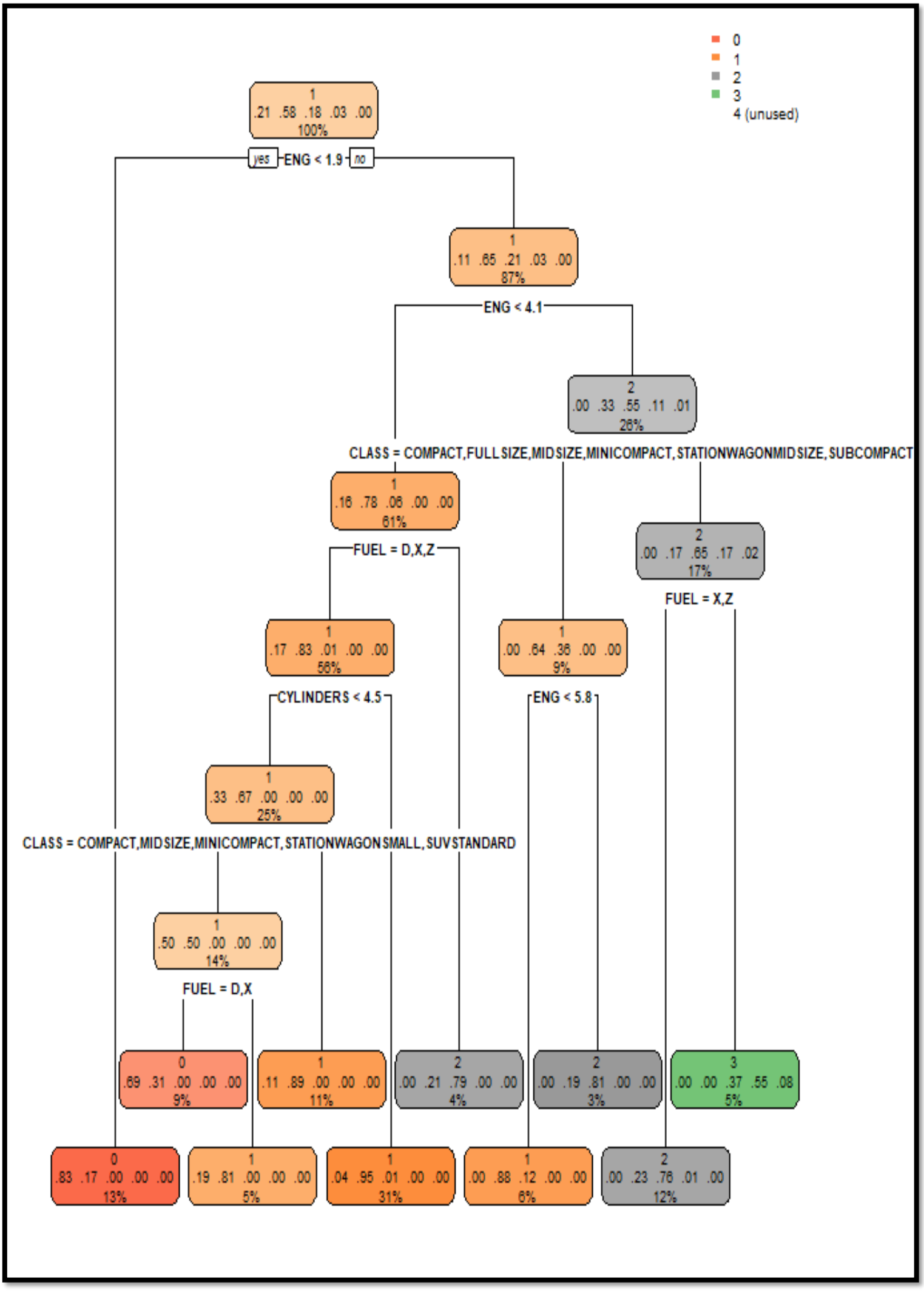
MinSplit=50 , Complexity (cp) = 0.01

yielded the best accuracy.

6. R script for Decision Tree Classifier

```
fueltree15= rpart(consume~ENG+CLASS+FUEL+CYLINDERS+GEARS,data=train,method="class",c
ontrol=rpart.control(cp=0.001,minsplit = 50))
predFuel=predict(fueltree15 ,newdata=test[,c(4:6,8:11)],  type="class")
fuelCM=table(test$consume,predFuel)
fuelCM
confusionMatrix(predFuel,test$consume)
rpart.plot(fueltree15)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1   2   3   4
         0  58  12   0   0   0
         1   8 131   3   0   0
         2   0  12  31   0   0
         3   0   0   2   8   1
         4   0   0   0   0   0

Overall Statistics

               Accuracy : 0.8571
                 95% CI : (0.8092, 0.8969)
    No Information Rate : 0.5827
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7621
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            0.8788   0.8452   0.8611  1.00000 0.000000
Specificity            0.9400   0.9009   0.9478  0.98837 1.000000
Pos Pred Value         0.8286   0.9225   0.7209  0.72727      NaN
Neg Pred Value         0.9592   0.8065   0.9776  1.00000 0.996241
Prevalence             0.2481   0.5827   0.1353  0.03008 0.003759
Detection Rate         0.2180   0.4925   0.1165  0.03008 0.000000
Detection Prevalence   0.2632   0.5338   0.1617  0.04135 0.000000
Balanced Accuracy      0.9094   0.8730   0.9045  0.99419 0.500000
```


0
1
2
3
4 (unused)
1
.21 .58 .18 .03 .00
100%
yes
ENG < 1.9
no
1
.11 .65 .21 .03 .00
87%
ENG < 4.1
2
.00 .33 .55 .11 .01
26%
CLASS = COMPACT,FULL SIZE,MID SIZE,MINICOMPACT,STATION WAGON MID SIZE,SUBCOMPACT
1
.16 .78 .06 .00 .00
61%
FUEL = D,X,Z
2
.00 .17 .65 .17 .02
17%
FUEL = X,Z
1
.17 .83 .01 .00 .00
56%
1
.00 .64 .36 .00 .00
9%
CYLINDERS < 4.5
ENG < 5.8
1
.33 .67 .00 .00 .00
25%
CLASS = COMPACT,MID SIZE,MINICOMPACT,STATION WAGON SMALL,SUV STANDARD
1
.50 .50 .00 .00 .00
14%
FUEL = D,X
0
.69 .31 .00 .00 .00
9%
1
.11 .89 .00 .00 .00
11%
2
.00 .21 .79 .00 .00
4%
2
.00 .19 .81 .00 .00
3%
3
.00 .00 .37 .55 .08
5%
0
.83 .17 .00 .00 .00
13%
1
.19 .81 .00 .00 .00
5%
1
.04 .95 .01 .00 .00
31%
1
.00 .88 .12 .00 .00
6%
2
.00 .23 .76 .01 .00
12%