Assignment 4: Theory of Decision Trees

Shashwat Patel
Metallurgical and Materials Engineering
Indian Institute of Technology,Madras
mm19b053@smail.iitm.ac.in

***Abstract*—Decision Trees are a type of supervised machine learning algorithm where the data is continuously divided into different categories on the basis of certain parameters. It is like a tree structure that works on certain conditions. Tree algorithms are always preferred due to stability and reliability. In this paper, we explain the theory behind the decision trees and try to understand what parameters are most important while classifying a car based on different parameters.**

***Index Terms*—Decision Trees, Entropy, Information Gain, Gini Index**

I. Introduction

Cars have now become one of the most important part of our daily lives. The buyer gets so many different choices in cars as there are many different manufacturers. This choice depends mainly on the price, safety, how luxurious and spacious the car is [1]. These parameters vary based on type, model, and manufacturer of the car. The Car Evaluation Dataset was derived from a simple hierarchical decision model.

Decision trees are one of the most important tool in supervised machine learning algorithm for decision making. A decision tree is a tree like structure consisting of different branches and leaves that points to all the various factors concerning a particular situation. Depending on the situation and desired outcome there are various types of decision trees that one can use. Decision trees are very easy to understand and are very intuitive.

Decision tree algorithms find applications in many different fields like in engineering, education, law, business, healthcare, and finance. It can be used to statistically compare data. It can be used in text classification or extraction. Historical data can be used in decision trees that can lead to some changes in the operation of a service. These can be used in genre classification of books or movies.

Buying a car is a very important responsibility. It is important to understand the true financial responsibility that comes with owning a car. The ”car_evaluation.csv” dataset is used to identify how parameters like price, price of maintenance, number of doors, capacity, size of luggage boot and safety affect the choice of a buyer while buying a car.

This paper majorly deals with the theory of decision trees and the mathematics behind it. We try to understand what parameters are the most while choosing a car to buy using the decision trees algorithm. The decision tree model is trained on the dataset ”car_evaluation.csv”.

II. Decision Trees

Decision trees is a supervised machine learning algorithm that involves segmenting the predictor space into different number of simpler regions. There are some rules based on which the predictor space is divided, these rules can be summarized as a tree, so these type of approaches are commonly as decision trees [2]. This algorithm is very simple and easy to interpret. These can be used for regression and classification problems. There is a family of decision tree learning algorithms like ID3, CART, ASSISTANT etc.

A decision tree starts with a root node and ends with a decision made by leaves. Some terminologies related to decision trees are [3]:

- Root Nodes – The node present at the beginning of a decision tree. From this node the space starts dividing according to various features.
- Decision Nodes – The nodes after splitting the root nodes.
- Leaf Nodes- The nodes where further splitting is not possible. Also known as terminal node.
- Sub-tree- Sub-section of the decision tree.

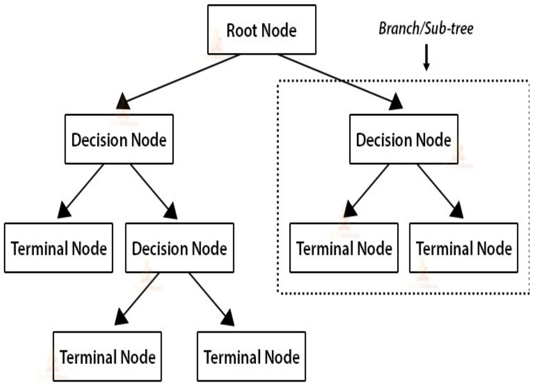


Fig. 1. Decision Tree structure

This algorithm can be considered as a bunch of if-else statements. It checks if the condition is true and if it is then it goes to the next node attached to that decision. Each node in the tree represents a test of a feature and a branch descending from that node indicates one of the possible values for that feature. So, classification starts at a root node of the tree, tests a feature at this node, then moves down the tree branch corresponding to the value of the feature. This process is then repeated. The main aim of a decision tree is to identify the features which contain the most information regarding the target feature and then split the dataset along the values of those features such that the target feature values at the resulting nodes are as pure as possible. A feature that best separates the uncertainty from information about the target feature is said to be the most informative feature. These questions, what should be the root node?, what should be the decision node?, and when to stop the splitting? are decided by different metrics like entropy, gini index and information gain.

Entropy: It is the measure of uncertainty or disorder in the dataset. The Entropy is defined by the formula [4]:

$$H(T) = I_E(p_1, p_2, p_3, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i \tag{1}$$

Here, $p_1, p_2, \ldots, p_J$ are the probability of randomly selecting an example in class $1, 2, \ldots, J$. J is the total number of classes for a particular feature. It is used to determine how a decision tree chooses to split data.

When all the observations of a dataset belongs to the same class, then entropy is zero. Such datasets have no disorderliness. That particular dataset is not good for learning. When the observations in a dataset are equally distributed in different classes, then entropy is 1. That dataset is good for learning.

*Gini Index:*It is calculated by subtracting the sum of squared probabilities of each class from one. It is easy to interpret and favors large partitions. [4]

$$Gini\ Index = 1 - \sum_{i=1}^{J} p_i^2 \tag{2}$$

A feature with a lower Gini index is chosen for a split. An feature with least gini index is preferred as root node while making a decision tree. The CART algorithm uses gini index to construct the decision tree.

Information Gain: [4]It is used for determining the best feature that gives maximum information about a class. It is based on the concept of entropy and aims to reduce the entropy, beginning from the root node to the leaf node. Information gain computes the difference between entropy before and after split and specifies the uncertainty in class elements.

Decision Tree Advantages [5]:

- They are easy to understand and interpret, perfect for visual representation. They very closely mimic the human decision making process.
- They can easily handle categorical features, there is no need for encoding or dummy variables.
- Feature selection happens automatically, unimportant features will not influence the result.

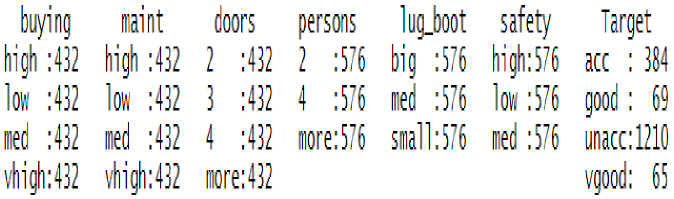
Disadvantages [5]:

- A small change in the data can cause a large change in the structure of the decision tree causing instability.
- Decision trees are prone to overfitting. In order to fit the data (even noisy data), it keeps generating new nodes and ultimately the tree becomes too complex to interpret. But there are methods to prevent overfitting such as pruning, which reduces the size of decision trees by removing sections of the tree that are non-critical and redundant.

III. The Problem

The ”adult_evaluation.csv” dataset is used to identify what parameters are important for a person while buying a car. The features include buying cost, maintenance cost, no. of doors, no. of persons, size of luggage boot and level of safety. The dataset consists of 1728 observations and 7 columns.

Not much data pre-processing was required. There were no missing values in the dataset. All the columns except the ”Target” variable column had equal distribution of observations.



```
  buying      maint      doors     persons    lug_boot    safety      Target
 high :432   high :432   2   :432   2   :576   big  :576   high:576   acc  : 384
 low  :432   low  :432   3   :432   4   :576   med  :576   low :576   good :  69
 med  :432   med  :432   4   :432   more:576   small:576   med :576   unacc:1210
 vhigh:432   vhigh:432   more:432                                     vgood:  65
```

Fig. 2. Dataset Summary

The distribution of ”Target” variable is as follows:

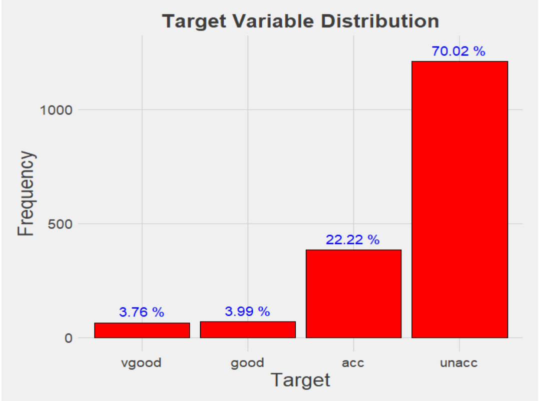


Fig. 3. Target Variable Distribution

Figure 3 is the distribution of the target variable in the dataset. It shows that around 70% of the target variable is ”unacc”, suggesting that this dataset is imbalanced.

*Data Visualization:*certain insights help in determining parameters that are most important while purchasing a car.

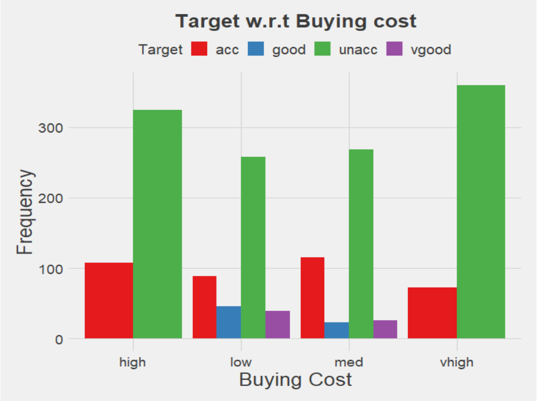


Fig. 4. Target w.r.t Buying Cost

Figure 4, shows the Target with respect to buying cost of the car. The plot suggests that most people prefer those cars whose buying cost is not that much high. They prefer cost within low to medium range. In general as well, people prefer those cars whose buying cost is low to medium as well.

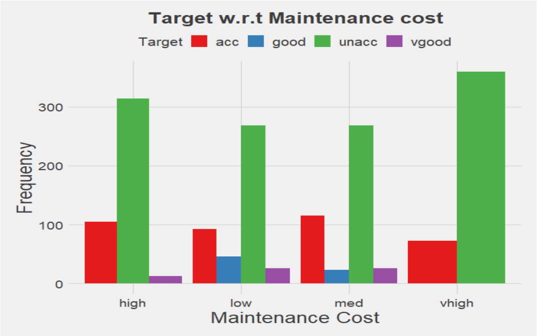


Fig. 5. Target w.r.t maintenance Cost

Figure 5, is the distribution of the target with respect to maintenance cost of the car. Here also, people prefer those cars mainly whose maintenance cost is in low to medium range. People tend to prefer cars with low maintenance cost in general as well.

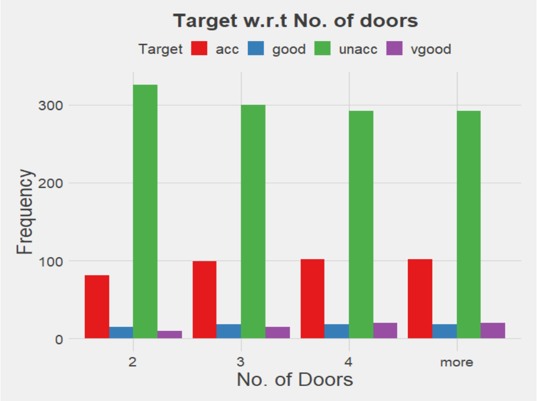


Fig. 6. Target w.r.t number of doors

Figure 6 shows the distribution of target with respect to number of doors in the car. This plot shows that people are mostly indifferent to number of doors in the car, but majority of people prefer those cars which have 3 or more doors. The number of doors doesn't seem to be an important factor while buying a car.

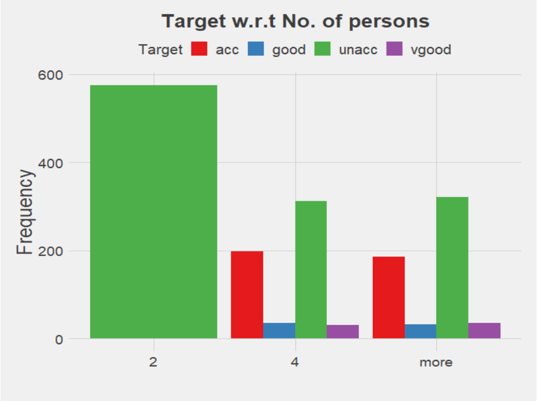


Fig. 7. Target w.r.t seating capacity

Figure 7 shows the distribution of target with respect to capacity of the car. The plot shows that people prefer those cars in which 4 or more persons can sit. According to the data, no one prefers cars with seating capacity of 2.

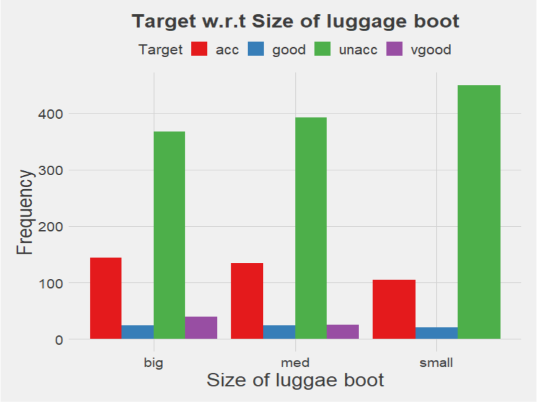


Fig. 8. Target w.r.t size of luggage boot

Figure 8 shows the distribution of target with respect to the size of luggage boot. People prefer cars with medium to larger luggage boot size but if the size is small it doesn't matter that much to people.

Figure 9 shows the distribution of target with respect to level of safety provided by the car manufacturers. The plot shows that the people prefer those cars which provide higher level of safety. In general as well, people tend to prefer those cars with better equipments of safety. The level of safety variable is very likely to be an important factor while purchasing a car.

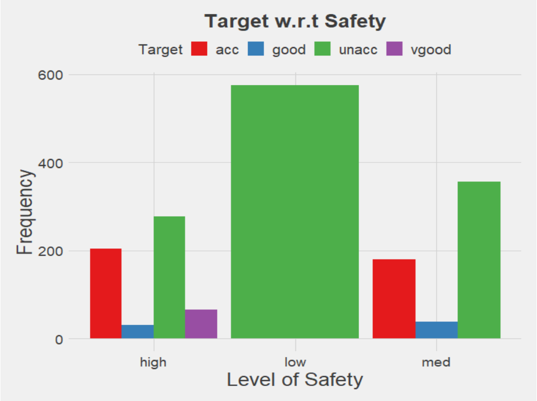
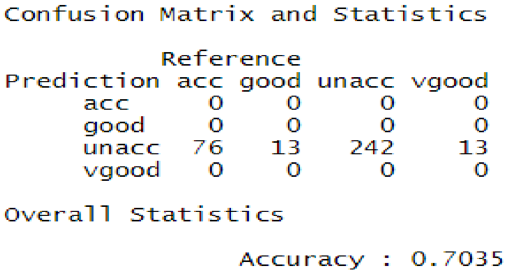


Fig. 9. Target w.r.t level of safety

*Decision Tree Model:*many different models have been built to check the different factors and to study what factors are most important while buying a car. The dataset was split into two test data set and train data set. 1384 observations in train data set and 344 observations in test dataset.

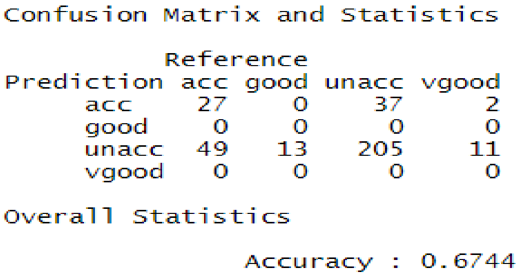
1) Model 1: It consists of the different costs only. These costs include the buying as well maintenance cost. The Model has an accuracy of 70% but it has predicted all the 344 observations as ”unacc”. This model doesn’t work.



```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc     0    0     0     0
     good    0    0     0     0
     unacc  76   13   242    13
     vgood   0    0     0     0

Overall Statistics

               Accuracy : 0.7035
```

Fig. 10. Confusion Matrix for Model-1

2) Model 2: 2nd model consists of seating capacity along with the costs. The model has an accuracy of 67%. The accuracy decreased from model-1 but in reality our model works better as it predicts other targets as well. This model also suggested that seating capacity is more important factor than the costs.

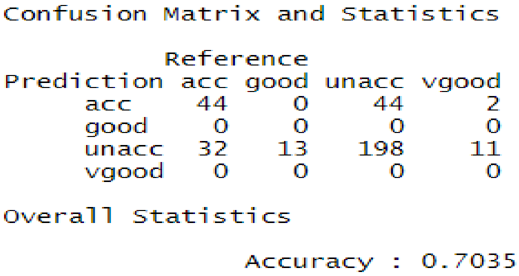


```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc    27    0    37     2
     good    0    0     0     0
     unacc  49   13   205    11
     vgood   0    0     0     0

Overall Statistics

               Accuracy : 0.6744
```

Fig. 11. Confusion Matrix for Model-2

3) Model 3: Along with all the other factors used in model 2, the model 3 also uses the number of doors factor. This model had an accuracy of 67% as well. There was no change detected in the decision tree suggesting that the number of door factor is not important while choosing a car.

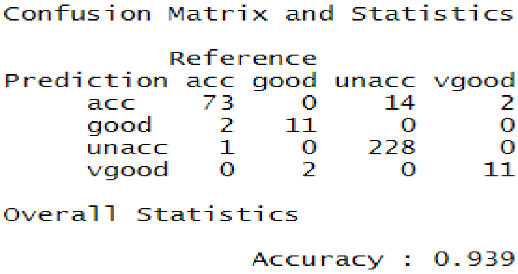
4) Model 4: This model has size of luggage boot added as well. This model is an important factor while buying a car. Our model had an accuracy of 70%.



```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc    44    0    44     2
     good    0    0     0     0
     unacc  32   13   198    11
     vgood   0    0     0     0

Overall Statistics

               Accuracy : 0.7035
```

Fig. 12. Confusion Matrix for Model-4

5) Model 5: This model has level of safety factor along with all the other features. This model had an accuracy of 93% suggesting that safety is an very important factor while choosing a car.

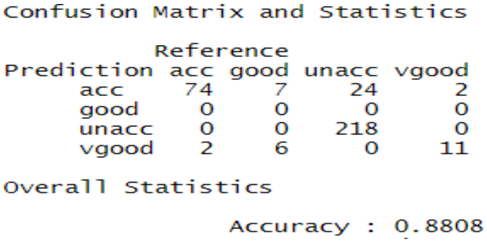


```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc    73    0    14     2
     good    2   11     0     0
     unacc   1    0   228     0
     vgood   0    2     0    11

Overall Statistics

               Accuracy : 0.939
```

Fig. 13. Confusion Matrix for Model-5

From all these models we learn that level of safety, seating capacity of the car are very important factors, cost and size of luggage boot are important and number of doors is not an important factor while choosing a car.

6) Model 6: The model 6 consists of buying cost, maintenance cost, safety and no of persons only. This model had an accuracy of 88%. This much accuracy is good as well. From visualizing the tree it is learnt that the root node is number of persons factor.



```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc    74    7    24     2
     good    0    0     0     0
     unacc   0    0   218     0
     vgood   2    6     0    11

Overall Statistics

               Accuracy : 0.8808
```

Fig. 14. Confusion Matrix for Model-6

All these models indicate that number of persons and safety are the most important factor to keep in mind while buying the car, the costs should be thought after that. Number of doors in the car is not important while purchasing a car.

IV. Conclusion

Decision trees helped in finding those parameters which were important to keep in mind while purchasing a car. Seating capacity of the car, level of safety in the car are the most important factor, buying cost, maintenance costs and size of luggage boots are pretty important and number of doors doesn't matter that much while purchasing a car. The model(Model 5) without number of door factor gave an accuracy of 93% on our validation dataset.

Further improvements are very likely possible. The relationship between the features was not studied that much, the relationship can help in finding better insights. Instead of only using the decision tree which is highly prone to overfitting, usage of random forest, support vector machine and other sophisticated models like neural networks can help as well.

References

[1] Awwalu, Jamilu & Ghazvini, Anahita & Abu Bakar, Azuraliza. (2014). Performance Comparison of Data Mining Algorithms: A Case Study on Car Evaluation Dataset. International Journal of Computer Trends and Technology. 13. 78-82. 10.14445/22312803/IJCTT-V13P117.

[2] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013.

[3] https://techvidvan.com/tutorials/decision-tree-in-r/

[4] https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

[5] http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of.html

Code Documentation

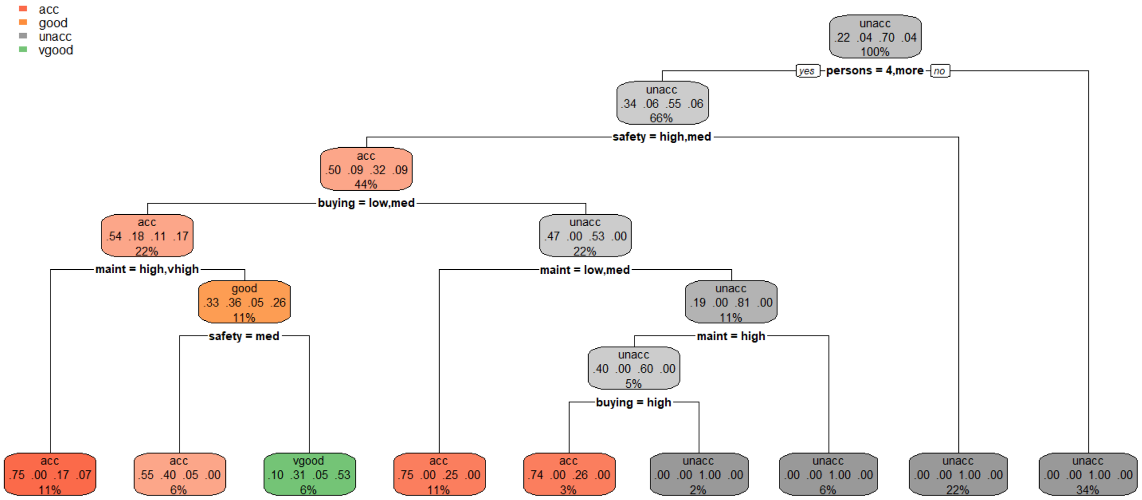


Figure 1: Tree Visualization

Import Libraries

```
library(tidyverse)
library(stringr)
library(rpart)
library(rpart.plot)
library(ggthemes)
library(caret)
```

Reading and giving column names to dataset

```
data<-read_csv("car_evaluation.csv",col_names=FALSE)

colnames(data)<-c("buying","maint","doors","persons","lug_boot","safety","Target")
```

Converting all variables to factors

```
data$doors<-str_replace_all(data$doors,pattern = "5more",replacement = "more")

data<-data%>% mutate_if(is.character,as.factor)
```

Basic Summary of data

```
str(data)
summary(data)
```

Check NAs in the data

```
na_check<-function(dataset){
  sapply(dataset,function(x) sum(is.na(x)))
}

na_check(data)
```

Visualizations

Setting the theme

```
my_theme<-theme_fivethirtyeight()+theme(plot.title = element_text(hjust = 0.5,size=20),
                                        axis.title = element_text(size=20),
                                        axis.text = element_text(size=14),
                                        plot.subtitle = element_text(hjust=0.5),
                                        legend.position = "top",
                                        legend.title= element_text(size=15),
                                        legend.text = element_text(size=15))
```

fig 3

```
target_var<-as.data.frame(table(data$Target))

colnames(target_var)<-c("Target","Frequency")

target_var%>%mutate(Target=reorder(Target,Frequency))%>%
  ggplot()+geom_col(aes(Target,Frequency),fill="red",col="black")+
  labs(title = "Target Variable Distribution")+
  my_theme+
  geom_text(aes(x=Target,y=Frequency+50,
                label=paste(round((Frequency*100)/sum(Frequency),2),"%")),
            size=5,col="blue")
```

fig 4

```
data%>%ggplot()+geom_bar(aes(buying,fill=Target),position = "dodge")+
  labs(x="Buying Cost",y="Frequency",title = "Target w.r.t Buying cost")+
  my_theme+
  scale_fill_brewer(palette = "Set1")
```

fig 5

```
data%>%ggplot()+geom_bar(aes(maint,fill=Target),position = "dodge")+
  labs(x="Maintenance Cost",y="Frequency",title = "Target w.r.t Maintenance cost")+
  my_theme+
  scale_fill_brewer(palette = "Set1")
```

fig 6

```
data%>%ggplot()+geom_bar(aes(doors,fill=Target),position = "dodge")+
  labs(x="No. of Doors",y="Frequency",title = "Target w.r.t No. of doors")+
  my_theme+
  scale_fill_brewer(palette = "Set1")
```

fig 7

```
data%>%ggplot()+geom_bar(aes(persons,fill=Target),position = "dodge")+
  labs(x="",y="Frequency",title = "Target w.r.t No. of persons")+
  my_theme+
  scale_fill_brewer(palette = "Set1")
```

fig 8

```
data%>%ggplot()+geom_bar(aes(lug_boot,fill=Target),position = "dodge")+
  labs(x="Size of luggae boot",y="Frequency",title = "Target w.r.t Size of luggage boot")+
  my_theme+
  scale_fill_brewer(palette = "Set1")
```

fig 9

```
data%>%ggplot()+geom_bar(aes(safety,fill=Target),position = "dodge")+
  labs(x="Level of Safety",y="Frequency",title = "Target w.r.t Safety")+
  my_theme+
  scale_fill_brewer(palette = "Set1")
```

Models

Splitting the dataset

```
set.seed(123)
trainIndex <- createDataPartition(data$Target,p=0.8,list=FALSE)

train<-data[trainIndex,]
test<-data[-(trainIndex),]
```

All models

Model 1

```
model_1<-rpart(Target~buying+maint,data=train,method='class')

predict_model1<-predict(model_1,test,type='class')

test$preicted_target_model1<-predict_model1

cm_model1<-confusionMatrix(predict_model1,test$Target)
```

Model 2

```
model_2<-rpart(Target~buying+maint+persons,data=train,method='class')

predict_model2<-predict(model_2,test,type='class')

rpart.plot(model_2)
```

```
cm_model2<-confusionMatrix(predict_model2,test$Target)
```

Model 3

```
model_3<-rpart(Target~buying+maint+persons+doors,data=train,method='class')

predict_model3<-predict(model_3,test,type='class')

rpart.plot(model_3)
```

Model 4

```
cm_model3<-confusionMatrix(predict_model3,test$Target)



model_4<-rpart(Target~buying+maint+persons+doors+lug_boot,data=train,method='class')

predict_model4<-predict(model_4,test,type='class')

rpart.plot(model_4)
```

```
cm_model4<-confusionMatrix(predict_model4,test$Target)
```

Model 5

```
model_5<-rpart(Target~buying+maint+persons+lug_boot+safety,data=train,method='class')

predict_model5<-predict(model_5,test,type='class')

rpart.plot(model_5)
```

```
cm_model5<-confusionMatrix(predict_model5,test$Target)
```

Model 6

```
model_6<-rpart(Target~safety+persons+buying+maint,data=train,method='class')

predict_model6<-predict(model_6,test,type='class')

cm_model6<-confusionMatrix(predict_model6,test$Target)

rpart.plot(model_6)
```