# CONTENTS

# Decision Tree:

## 1 Decision Tree Using iris Data Set:

I have Performed the same decision tree using various library

### 1.1 Decision Tree Using Library party

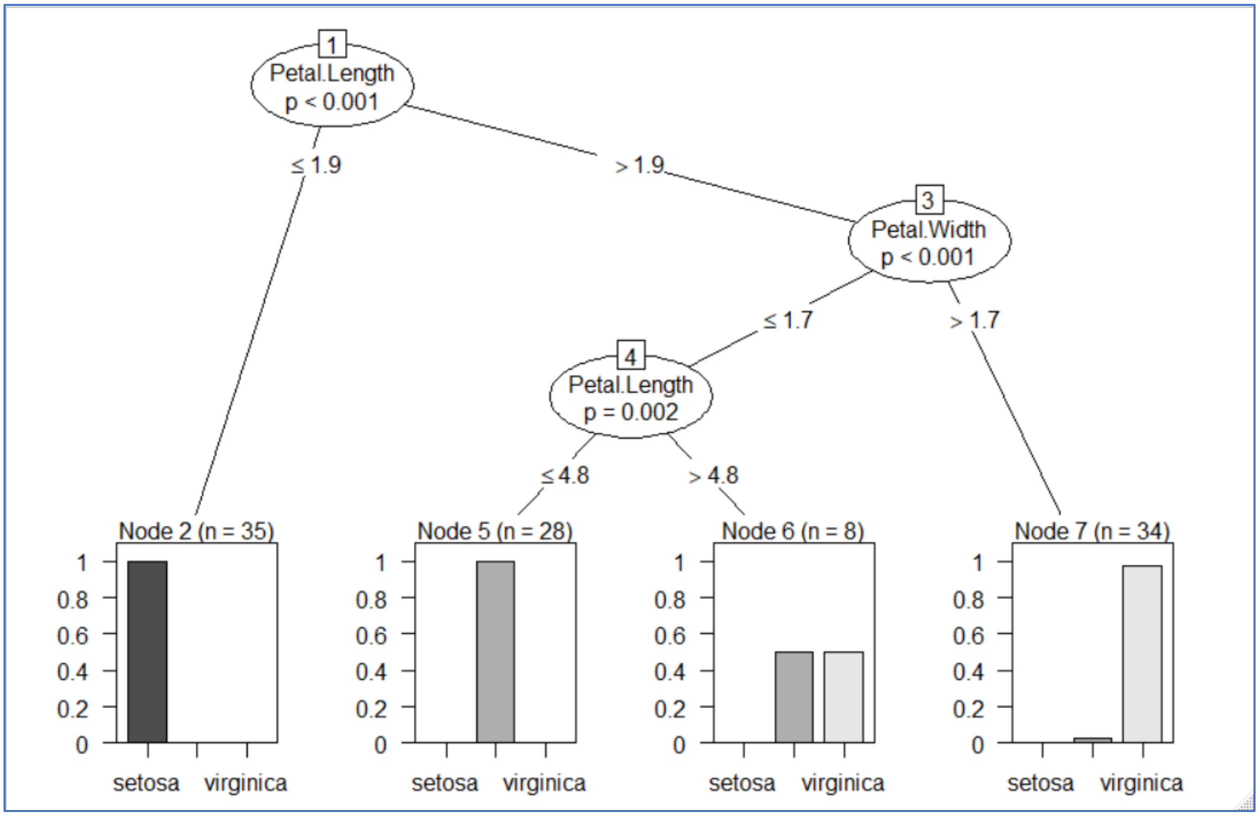
Here I used the function “ctree” from the library “party”.

Performed using all the columns except the species as my independent variable and species as my dependent variable.

Here I come up with confusion matrix given below

```
             Actual
Predicted    setosa versicolor virginica
  setosa       15        0        0
  versicolor    0       17        1
  virginica     0        0       12
```

Here for this model I come up with 97.7% efficiency



## 1.2 DECISION TREE USING LIBRARY TREE

Here I used the function “tree” from the package “tree”.

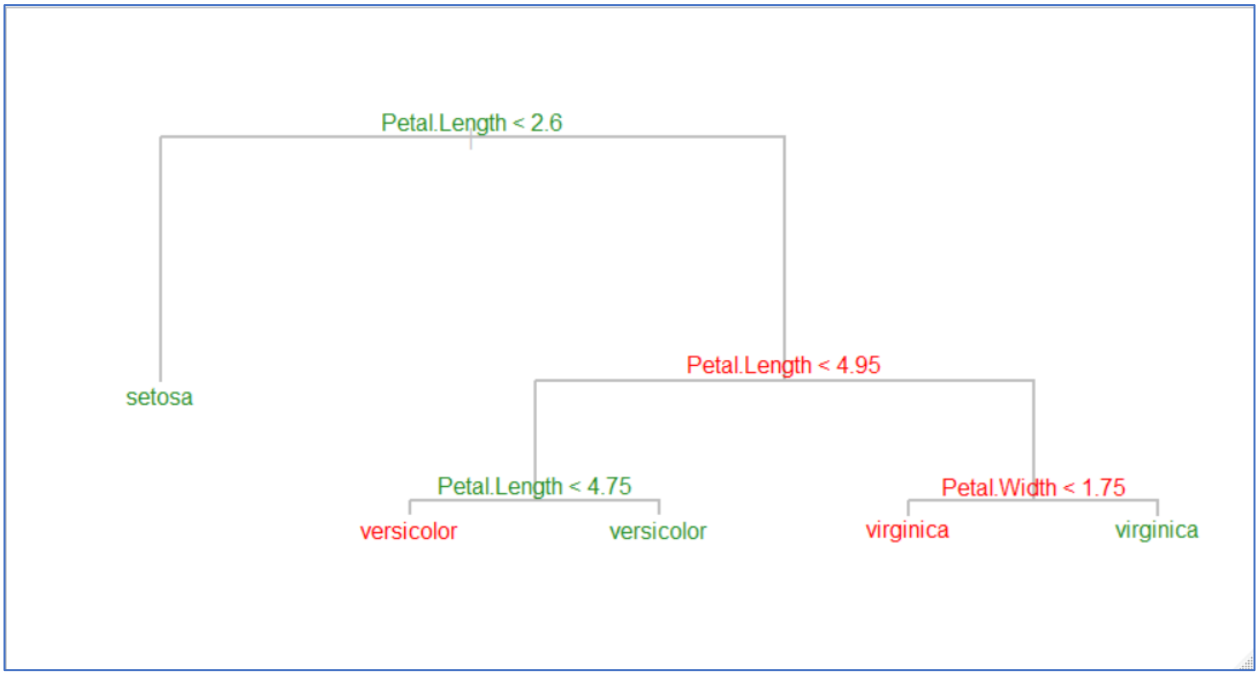
Performed using all the columns except the species as my independent variable and species as my dependent variable.

Here I come up with confusion matrix given below

```
              Actual
Predicted   setosa versicolor virginica
 setosa       15        0        0
 versicolor    0       17        4
 virginica     0        0        9
```

Here for this model my accuracy is 91%

### 1.2.1 Decision Tree pLot



### 1.2.2 Conclusion

The graph is comparatively better than the previous model, to view for a lemma, but it is unable to explain the probability within the graph.

Also from this model we can see that the efficiency of the model decreases from the previous model, so we are going to consider another better model for our calculation with another different package.

## 1.3 Decision Tree Using Library rpart

Here I used the function “rpart” from the package “rpart”.

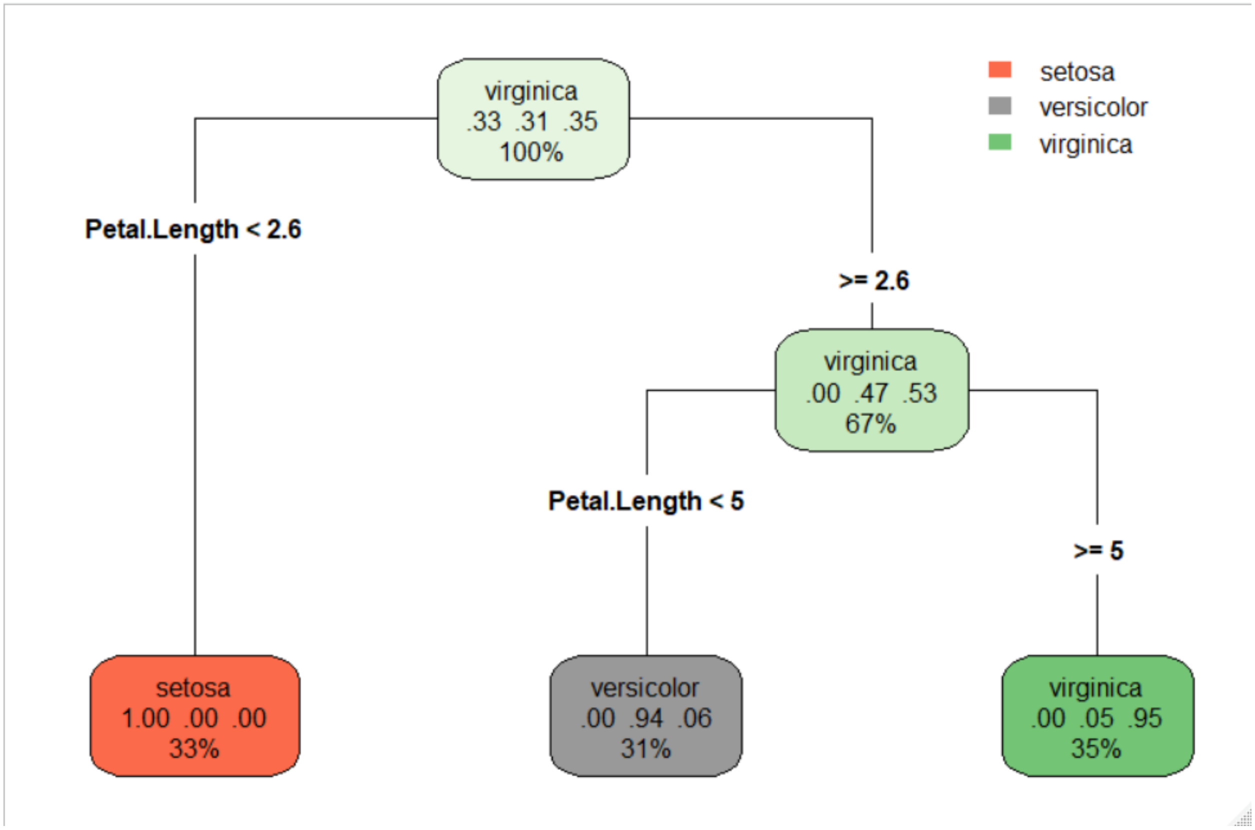
Performed using all the columns except the species as my independent variable and species as my dependent variable.

Here I come up with confusion matrix given below

```
                  Actual
Predicted    setosa versicolor virginica
 setosa        15        0        0
 versicolor    0        17        4
 virginica     0        0        9
```

Here I come up with same 91% of efficiency.

### 1.3.1 Decision Tree pLot



The Representation in this decision tree plot is quite impressive, the plot is done using the library “rpart.plot” .

## 1.4 Decision Tree Using Library C50

Here I used the function “C5.0” from the package “C50”.

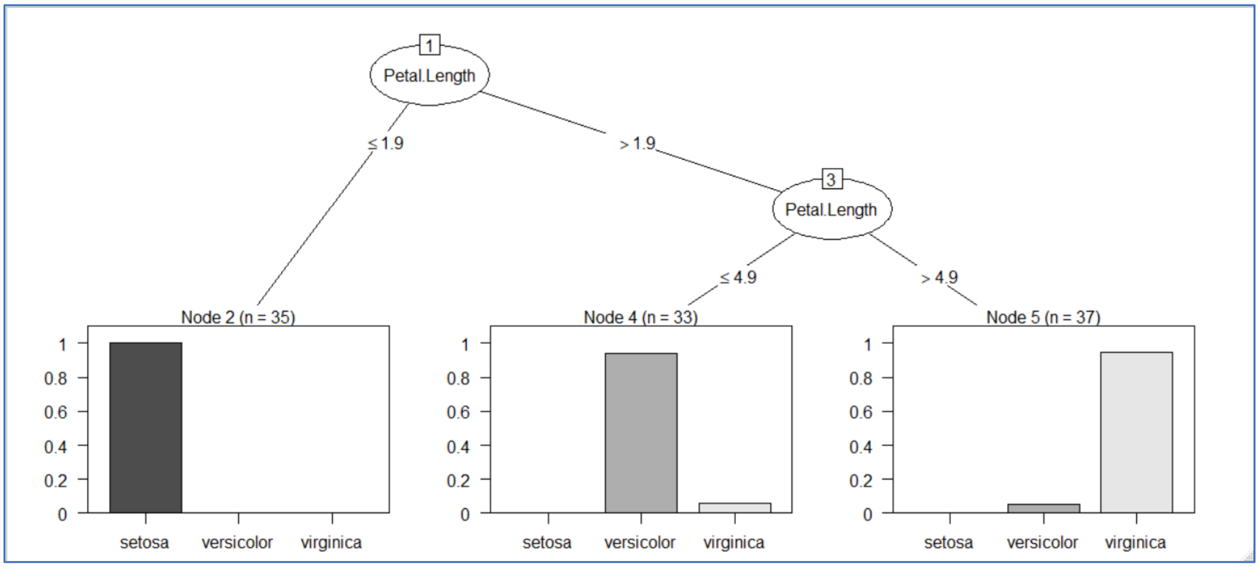
Performed using all the columns except the species as my independent variable and species as my dependent variable.

Here I come up with confusion matrix given below

```
                Actual
Predicted   setosa versicolor virginica
 setosa       15        0        0
 versicolor     0       17        4
 virginica     0        0        9
```

Here my accuracy is 91% similar result with the previous 2 models.

### 1.4.1 Decission Tree pLot



Here we can see the plot is similar to the model fitted in using the “ctree” function from the library “party”. Only differ with branches, here we can find only one branch and 3 leaf nodes as in the previous 2 plots.

Now let’s check the model wing the Ensemble technique called boosting.

Applying Boosting Trial for 10 we can see that the efficiency is increased to 97.7%

## 1.5 Conclusion

Here Using my “ctree” function from “party” library and using “C5.0” function from the “C50” library with boosting, I will get the best result. Other functions are good for plotting purpose.

I considered my boosted model with C5.0 function as my final model.

# 2 Decision Tree Using Fraud_check Data Set

Let's have a look on the data set

```
data.frame':       600 obs. of  6 variables:
 $ Undergrad      : Factor w/ 2 levels "NO","YES": 1 2 1 2 1 1 1 2 1 2 ...
 $ Marital.Status : Factor w/ 3 levels "Divorced","Married",..: 3 1 2 3 2 1 1 3 3 1 ...
 $ Taxable.Income : int  68833 33700 36925 50190 81002 33329 83357 62774 83519 98152 ...
 $ City.Population: int  50047 134075 160205 193264 27533 116382 80890 131253 102481 155482 ...
 $ Work.Experience: int  10 18 30 15 28 0 8 3 12 4 ...
 $ Urban         : Factor w/ 2 levels "NO","YES": 2 2 2 2 1 1 2 2 2 2 ...
```

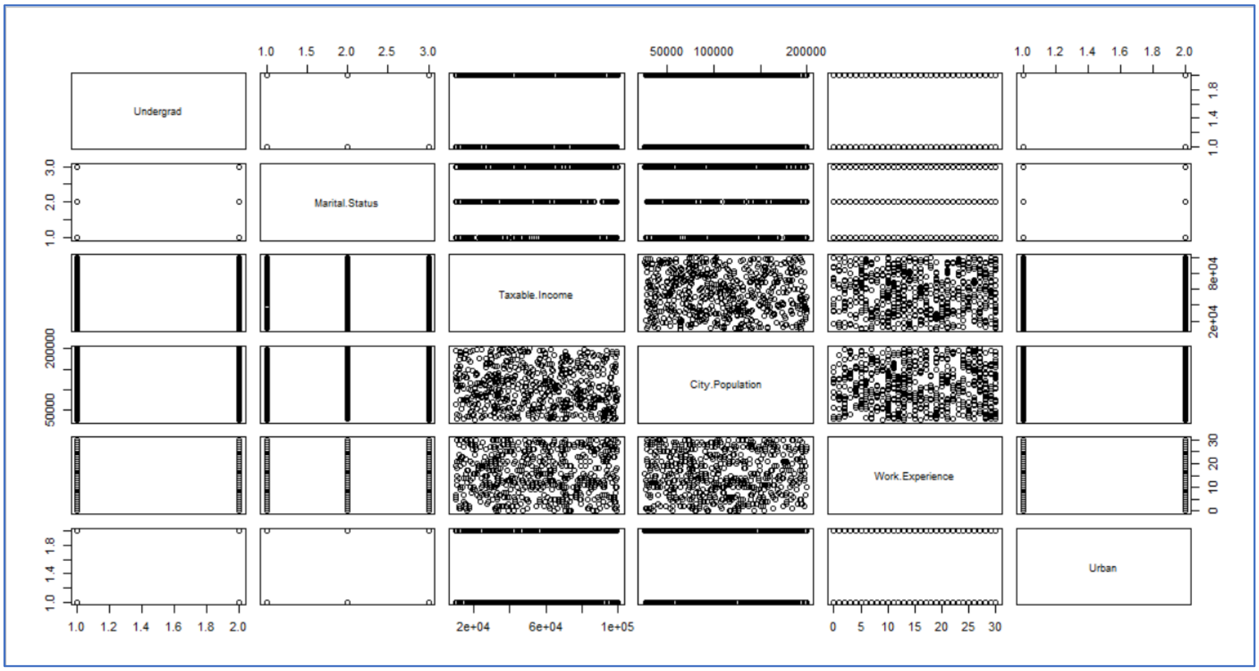
We have 3 factor variables Marital status, Urban & Undergrand. All other variables are numeric. Our focus is on Taxable Income.

Where the persons with Taxable income grater then 30000 are considered as the Good else Risky.

Now we create another variable with type, which is factor and contains desired result i.e. Good or Risky.

All of our model should be made with considering the target variable as type not the Taxable income.

Lets Have a look on the pairs plot, to see whether our taxable income has any correlation between our other variables or not.

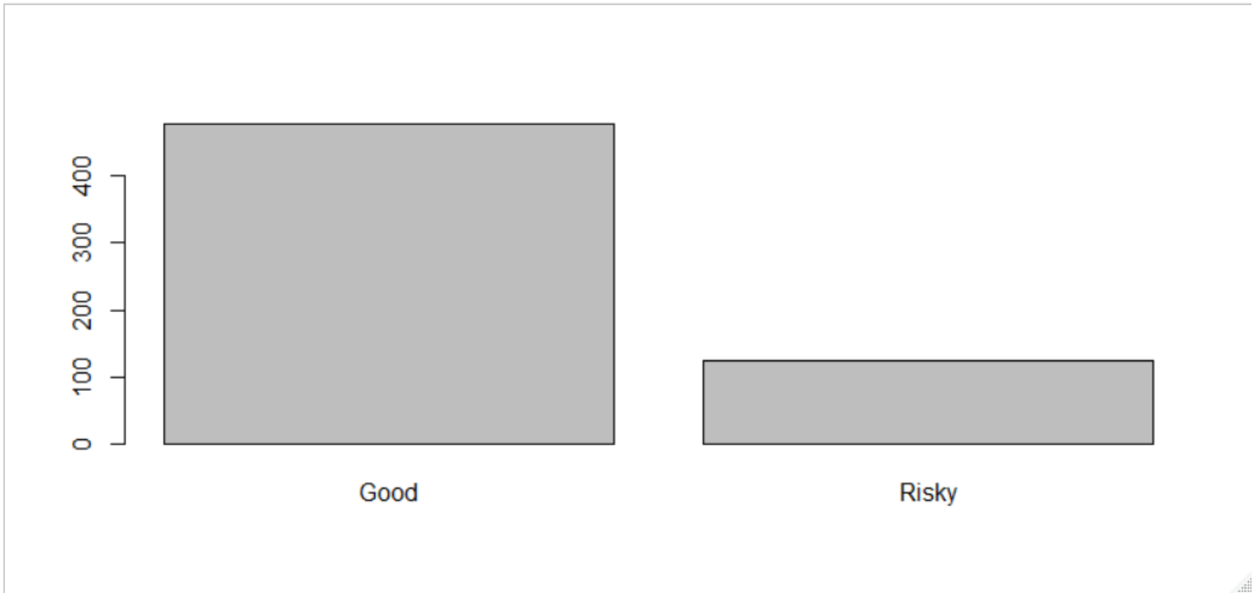


**From the pairs plot we can see that none of the variable is correlated with our variable taxable income, also we can see uniform distributed scatter plots between all the numeric variable, So from the starting itself I may consider that the model I am going to build with the data for classifying the Good or Risky category will be unreliable in my case.**

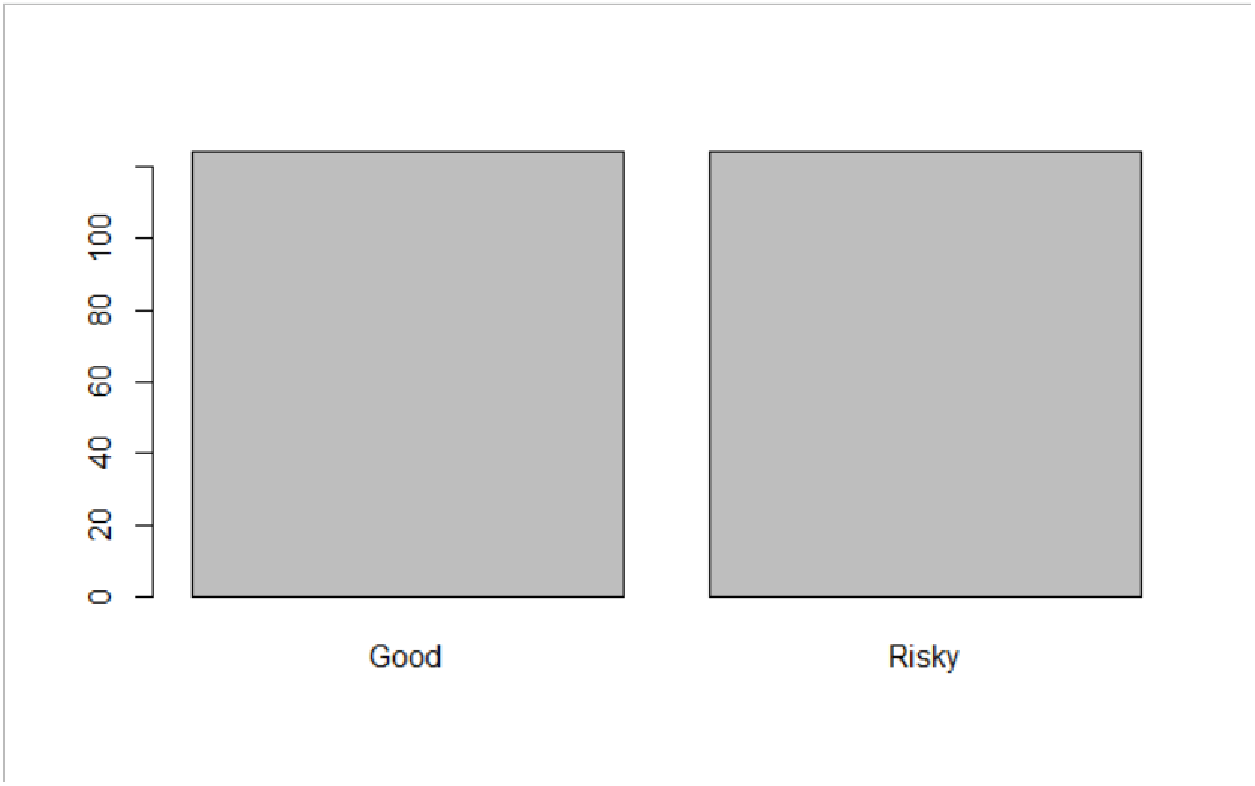
Which ever model we are going to build will be biased upon the ratio of our categorical variable.

## 2.1 TREATMENT WITH IMBALANCE DATA

Lets see whether our categorical Target variable is balanced or not.



Here Our categorical variable is imbalanced in my case, So I prefer to make the ratio equal i.e. 1.So, I am going to choose random sampling technique, for choosing 124 records from the category of Good in my data set, and consider all my 124 records with Risky category.



Here my data is balanced with equal ratio.

Now I am going to perform my modeling in  my data set.

## 2.2 Model 1 Using the ctree function in party library With the whole Data

Here I am going to consider my whole data and see whether the classification model reliable for whole data or not. And here my confusion Metrix I am with:

| | Actual | |
|---|---|---|
| Predicted | Good | Risky |
| Good | 476 | 124 |
| Risky | 0 | 0 |

Here my model contains all independent Columns beside the taxable income, as we encode as our categorical variable.

From our confusion matrix, I can clearly see that my model is predicting each and every variable as in Good category. Because Its not capable of finding any relation between the variables.

Here I am getting my accuracy as 0.7933 i.e. the ratio between the two categories in our column "type".

So, as a statistician I will make the data balanced and perform my model with balanced data to get my accuracy properly.

## 2.3 Model 2 Using the ctree function in party library With the Balanced Data

Here I am considering my sampled record of category "Good" and the all other records with category "Risky"

Here I come up with the confusion Metrix as given below:

| | Actual | |
|---|---|---|
| Predicted | Good | Risky |
| Good | 124 | 124 |
| Risky | 0 | 0 |

From the above confusion matrix, I can comment on my data that the model is not performing with the function "ctree", all of its classification is to "Good" category not "Risky".

Here my accuracy is 0.5, again as I told, it will be the ratio between "Good" and "Risky".

So I prefer to go for my feature selection before building my model.

## 2.4 Model 3 Using the ctree function in party library With the Train Data

Here in my model I am considering the train data for model building and classify the test data based on the model.

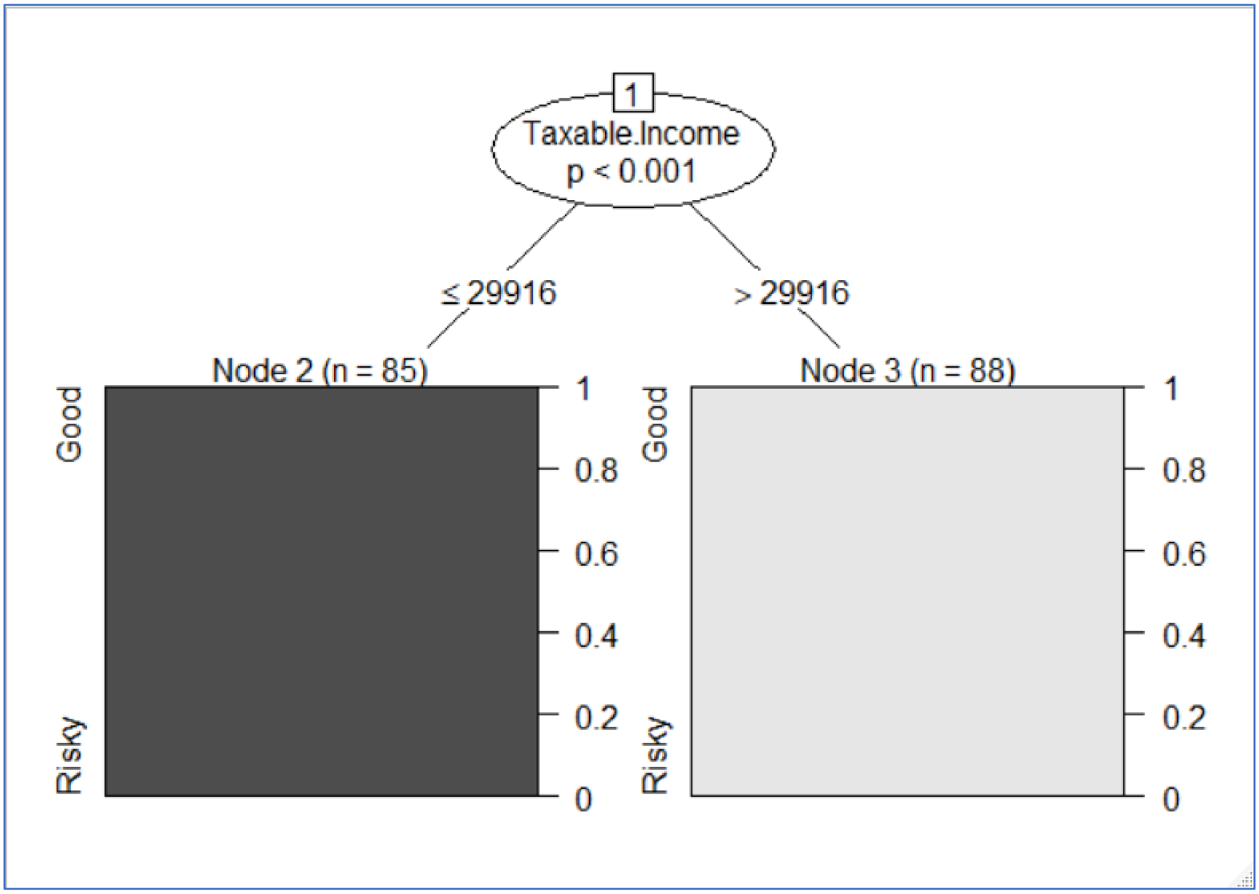
Here again I am considering my variable taxable income. To see that whether my model building formula are wrong or I am lacking behind something which is quite necessary for my model.

Confusion matrix:

|  | Predicted |  |
|---|---|---|
| Actual | Good | Risky |
| Good | 36 | 0 |
| Risky | 1 | 38 |

Here I can see that, our accuracy is very good i.e. 0.98

Lets have a look on the decision tree plot.



As we can see my whole tree dependent upon the root node itself i.e. the taxable income, So I can say that all my classification model is going to be unreliable.

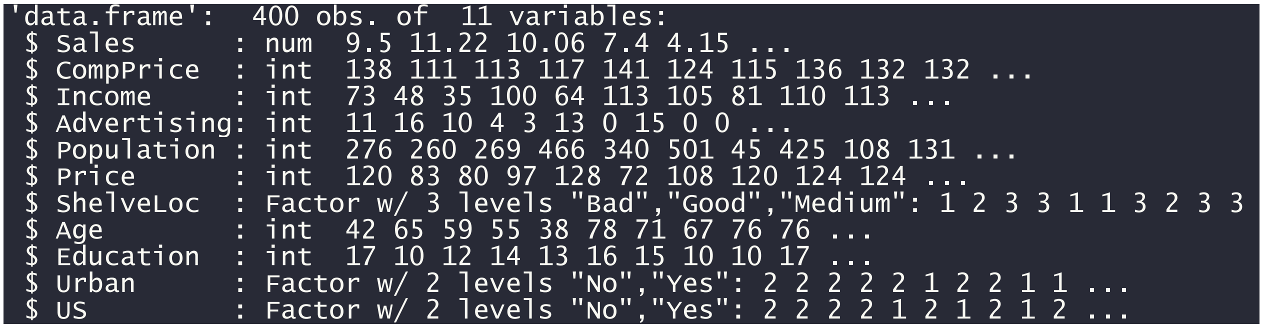
If the reviewer finds any reliable model than kindly let me know, whether any function is there to outperform my model with ctree.

## 2.5 Conclusion

*Due to lake of relevant information in my data set I can conclude that unless and until a relevant variable is not introduced with this data, it won't perform well. All the classification will be biased to one side.*

# 3 Decision Tree Using Company_Data, data Set:

Lets look at the structure of the data set:



```
'data.frame':	400 obs. of  11 variables:
 $ Sales      : num  9.5 11.22 10.06 7.4 4.15 ...
 $ CompPrice  : int  138 111 113 117 141 124 115 136 132 132 ...
 $ Income     : int  73 48 35 100 64 113 105 81 110 113 ...
 $ Advertising: int  11 16 10 4 3 13 0 15 0 0 ...
 $ Population : int  276 260 269 466 340 501 45 425 108 131 ...
 $ Price      : int  120 83 80 97 128 72 108 120 124 124 ...
 $ ShelveLoc  : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3
 $ Age        : int  42 65 59 55 38 78 71 67 76 76 ...
 $ Education  : int  17 10 12 14 13 16 15 10 10 17 ...
 $ Urban      : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...
 $ US         : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...
```

Here we can see excluding “Urban”, “US”, and “ShelveLoc” all are numeric.

Out of these data we need to see which records can be considered as the High Sales.

To Classify it I may consider the Sales to 3 categories as Low, Medium and High. Here I am going to give equal weightage to all categories based on cutoff values.

## 3.1 Data Classification Using variable Sales

So I have my 33.3% records with Low Sales, 33.3% of records with medium Sales and other 33.3% of sales as high sales here. I may consider top 33.3% of my Sales values as the High and rest which are 66.6% classify them as “Medium” and “Low”. As all my focus for “High” Sales, so I am considering 66.6% rest of the data as “Low”.

After sorting the data based on sales I come up with the  cutoff point as 8.67, so I can blindly say that the sales above than 8.5 can be considered as my High Sales values.

Based upon the cutoff point I create another variable with name “SalesC” (sales Category), with levels “High” and “Low”.

In my categorical variable for sale is in ratio 1:3 as High:Low i.e. kind of imbalanced, I may consider it balancing if I find any difficulty in my classification model. I removed my column Sales with the “SalesC” i.e. removing the numeric variable with the desired categorical variable

## 3.2 Model 1 considering all the Variables Using function “ctree” in library “party”

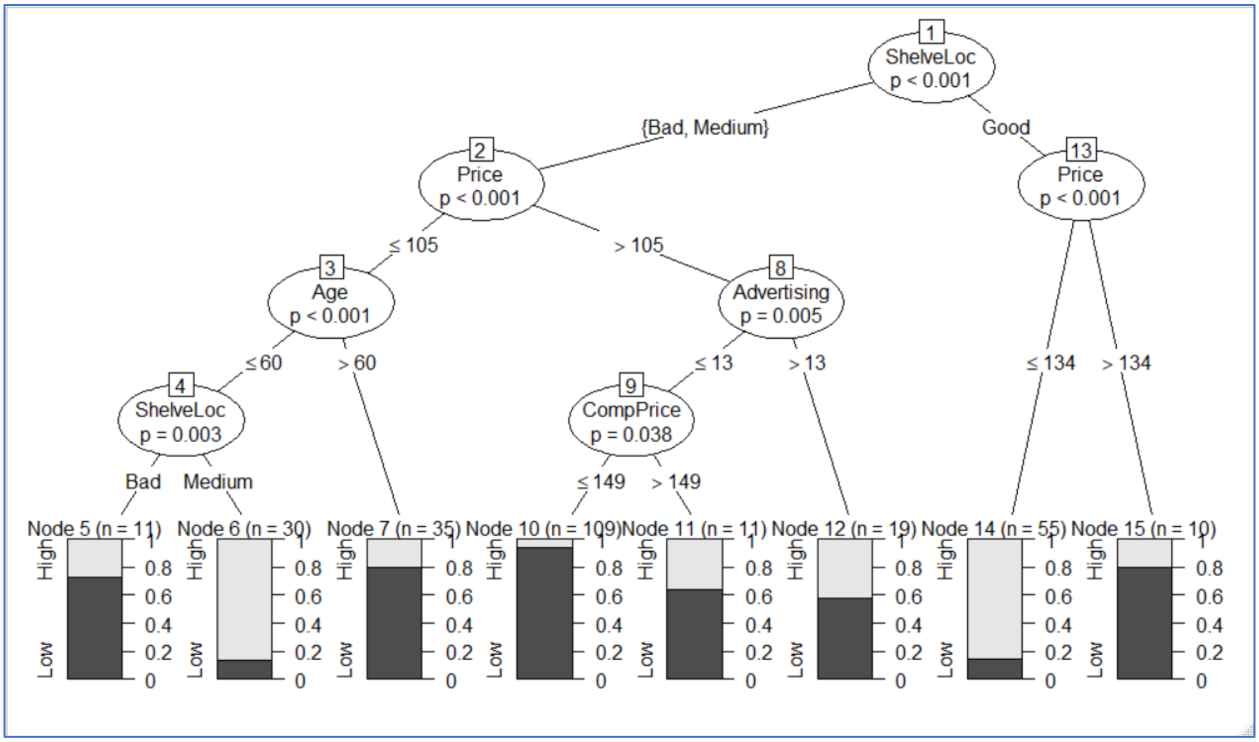
Here I considered my Train data for model fitting and rest Test data for my classification purpose.

I am using the “ctree” function for my model building purpose. Here I come up with the confusion metrix:

|  | Predicted |  |
|---|---|---|
| Actual | High | Low |
| High | 18 | 21 |
| Low | 8 | 73 |

Here our model is giving us 75.8% of accuracy, which is not quite effective for our business scenario. Lets have a look on the decision tree in my model using “ctree”

### 3.2.1 Decision Tree Plot:



To outperform my previous model I am going to build my model using the function “C5.0”.

## 3.3 Model 2 considering all the Variables Using function “C5.0” in library “C50”

In my second model I come up with the confusion Metrix as given below,

| | Predicted | |
|---|---|---|
| Actual | High | Low |
| High | 25 | 14 |
| Low | 10 | 71 |

Here in this model my accuracy is increased to 80% quite significant improvement in my model. For more accuracy I may go for the boosting method.

Using the boosting i.e. including the parameter trials in my model I get my accuracy as 85%
The decision tree structure is not visible properly in 1436x590 i.e. the maximum canvas size in my system, so I am unable to view it.

## 3.4 Conclusion

Here I conclude my modeling with my final boosted model using C5.0 function.