The dataset “Car.data " consists of 1728 rows with 7 features, that includes a “classification” column that describes four classes; unacc, acc, good and vgood that counts 1210, 384, 69 and 65 in number as shown in the Fig1. A unique numerical value is assigned to each class; dividing data between 4 classes. As all columns contain categorical data, label encoder is used. To balance the imbalance dataset, SMOTE (synthetic minority oversampling technique) is selected which balances the class distribution (handles the skewness) by having 1210 data in each as shown in Fig 2. To scale the input data, a StandardScaler is used.

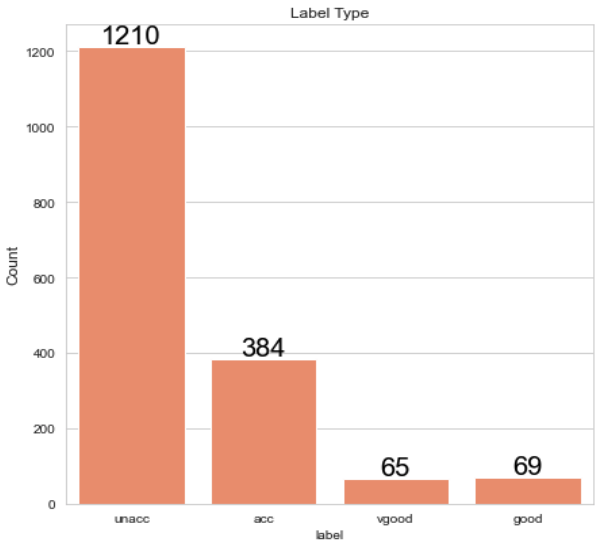

Fig 1. Imbalanced Data

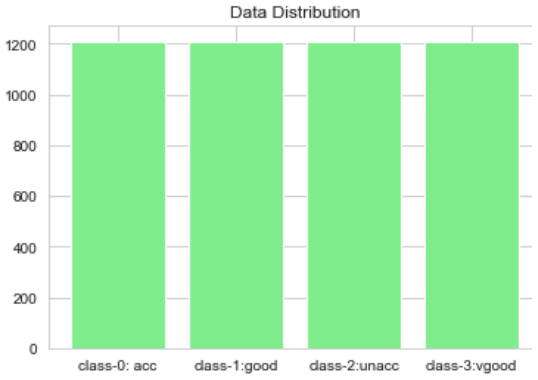

Fig 2. Balanced Data

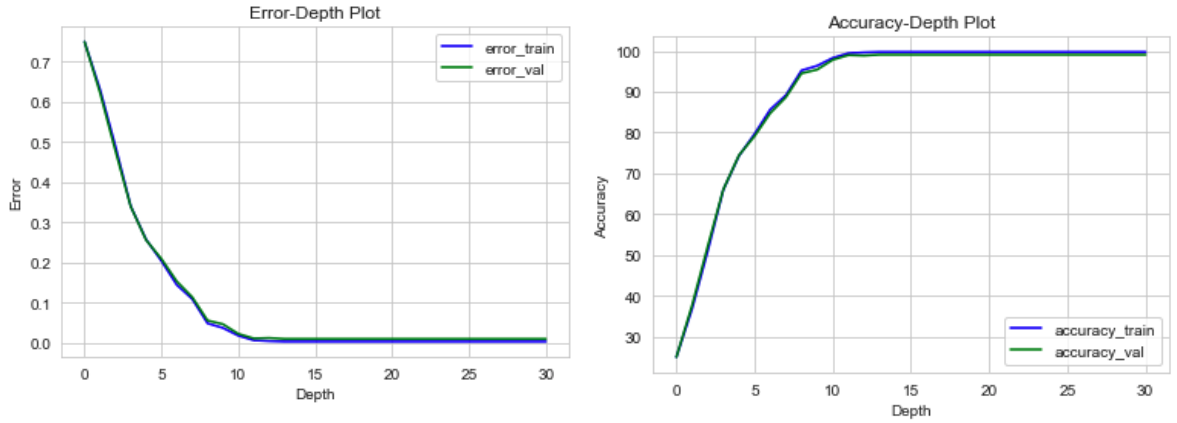

Fig 3. Error andAccuracy plot with Depth (Without Pruning)

The dataset is split into 3 chunks; train (0.7), validation (0.2) and test set (0.1). Stratified sampling and shuffling is done in the dataset. The impurity of a parent and child class is calculated (both Gini and Information gain). Since we have categorical data, Gini is used to split nodes. To split, impurity is calculated for every child node and gini of each split is calculated accordingly, Then, the split with lower gini value is selected from overall impurity values. It can be seen in Fig 3 that when depth is more, error is less and accuracy is more. Bith, Pre-pruning and Post-pruning is used as the stopping criteria to build the tree. While, the former one is achieved by setting the maximum depth of the tree and then, evaluating the results at different depth levels and finding out in which depth there is higher accuracy, the latter one is achieved by observing the Cost Complexity Pruning (CCP) value.

Fig 4. Shows the Alpha vs Accuracy plot (includes both train and test. It is noticeable that with a slight increase in alpha value, training accuracy and testing accuracy starts decreasing. When the alpha value is between 0.001 and 0.02, it seems it gives the optimal range. Moreover, when alpha value is 0.001, accuracy is 0.98 and when alpha value is 0.0025, it gives accuracy of 0.97. When alpha value is not applied in the classifier, for instance, it gives maximum depth as 8, which has 97 nodes with 49 leaves; giving accuracy of around 0.94. In the same scenario, when alpha value (0.02) is applied, it gives maximum depth as 8, which has 19 nodes and 10 leaves; giving starts accuracy at around 0.78. So, pruning is able to decrease the depth of the tree minimising the number of nodes by 80.41%.

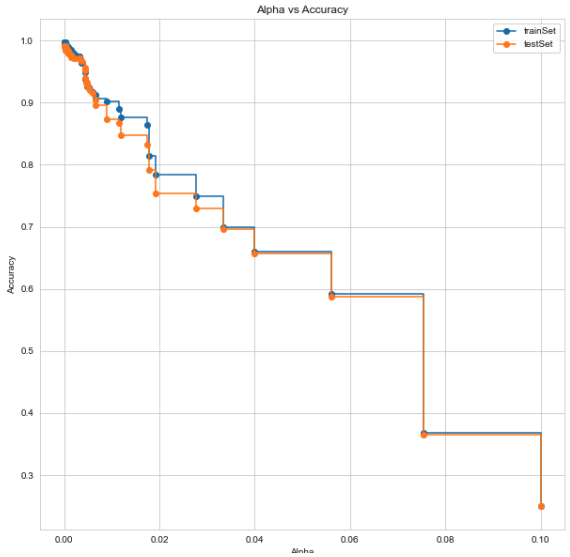

Fig 4. Alpha vs Accuracy plot

We can also verify this by plotting the graph between the number of nodes and depth with alpha value. In my case, when alpha increases, the accuracy decreases as the number of nodes and depth sharly decreases. During post pruning, different values of alpha that lie within the optimal interval are selected. Even, by giving the accurate alpha value to the classifier, it returns the calculated depth along with the number of nodes accordingly. Pruning strategy also overcomes the overfitting problem as it limits the size of the tree by assigning the maximum depth of the tree and other parameters as required. Also, cross validation is used to determine the accuracy of the model based on different ranges of depths (selected by observing accuracy, alpha values). Confusion Matrix and Classification report is generated to see the result.

Result:

Train set:

When depth= 10 it gives mean accuracy of 0.98 with error rate to be around 0.02.
When depth= 11 it gives mean accuracy of 0.99 with error rate to be around 0.08;
When depth= 12 it gives mean accuracy of 0.99 with error rate to be around 0.06;
When depth= 13 it gives mean accuracy of 0.99 with error rate to be around 0.07.

Validation Set:

When depth= 10 it gives mean accuracy of 0.97 with error rate to be around 0.02.
When depth= 11 it gives mean accuracy of 0.98 with error rate to be around 0.01.
When depth= 12 it gives mean accuracy of 0.98 with error rate to be around 0.01.
When depth= 13 it gives mean accuracy of 0.99 with error rate to be around 0.01.

Test Set:

When depth= 10 it gives mean accuracy of 0.98 with error rate to be around 0.01.
When depth= 11 it gives mean accuracy of 0.98 with error rate to be around 0.01.
When depth= 12 it gives mean accuracy of 0.98 with error rate to be around 0.01.
When depth= 13 it gives mean accuracy of 0.99 with error rate to be around 0.08.

When ccp_alpha = 0.03, max_depth predicted = 6, nodes = 15, leaf nodes as 8, accuracy = 0.74
When ccp_alpha = 0.0025, max_depth predicted = 11, nodes= 83, leaf nodes = 42, accuracy= 0.97.
When ccp_alpha = 0.001, max_depth predicted = 12, nodes= 105, leaf nodes = 53, accuracy= 0.98.
When ccp_alpha = 0.006, max_depth predicted = 10, nodes= 45, leaf nodes = 23, accuracy= 0.91.
When ccp_alpha = 0.007, max_depth predicted = 10, nodes= 37, leaf nodes = 19, accuracy= 0.90.

When depth = 8 and ccpa_alpha= 0, it gives 97 nodes , 49 leaves giving accuracy of around (0.94).
When depth = 8, ccpa_alpha value= 0.02, it gives 19 nodes, 10 leaves giving accuracy of around (0.78).

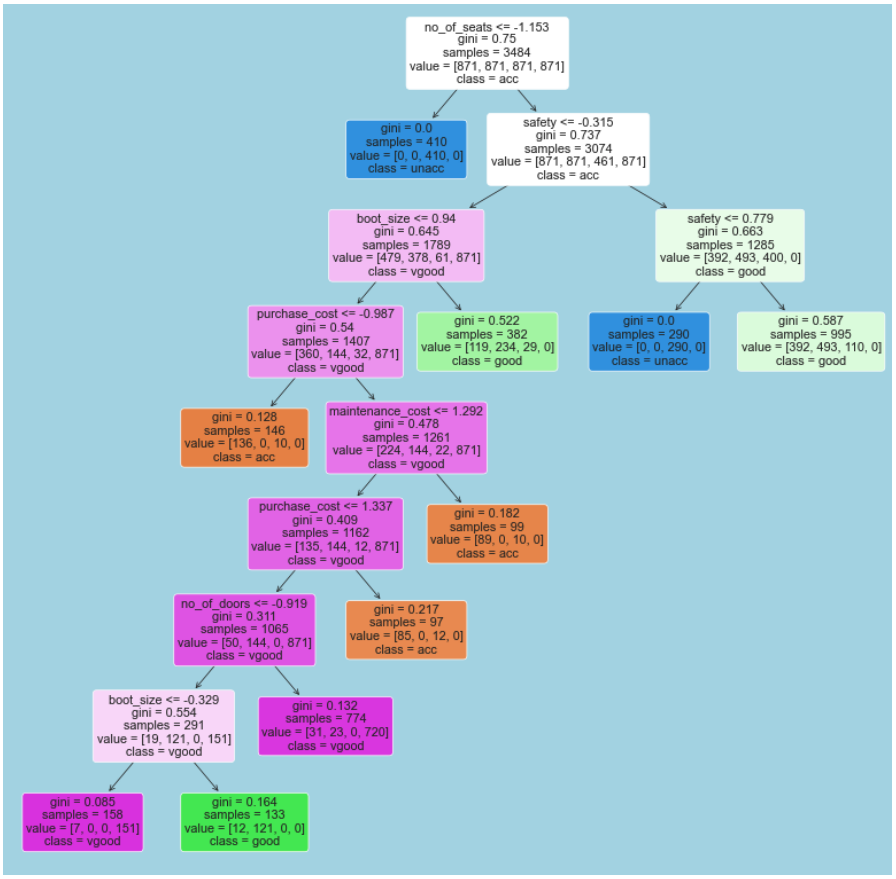

Fig 5. Decision Tree with depth = 8