

EECS738
Madhu Peduri (3006758)
pmspraju@ku.edu

Question 1:

Total number of records: 768

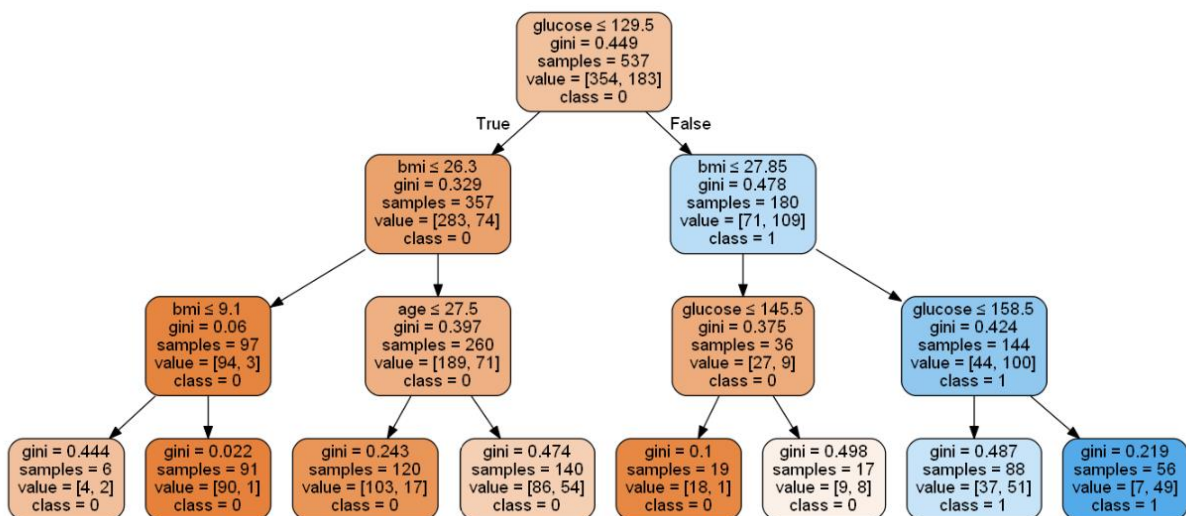
Total number of features: 9

Training set has 537 samples.

Testing set has 231 samples.

Times for Training, Prediction: 0.00174, 0.00122

Accuracy for Training, Test sets: 0.76350, 0.75758



Question 2:

Pre processing steps:

- 1) In 'Insurance fraud' dataset, we have few attribute 'Id' that is continuous and sequential. This is not useful to train our model. We can drop this attribute
 - 2) Another attribute 'Insurance type' is same across the dataset. This is not useful to train our model. We can drop this attribute.
 - 3) After few test run, I found attribute 'Marital status' is not contributing much to our model. We can drop this attribute. Inclusion of this attribute does not have effect on accuracy of the model, but omission of this is reducing the elapsed time.
 - 4) We have another attribute 'Claim amount Received', if we look at the data for this feature, we can observe that if amount > 0 then target value is 0 and if amount = 0, then target = 1. If we use this feature, it will complete the tree with one iteration. So, we can avoid this feature.
 - 5) We have attributes 'Income of policy holder' and 'Total Claimed amount' having high values and zeroes consistently. As zeros are outliers in this case and this makes this attribute skewed. So, we use logarithmic transformation to make the outliers even.
 - 6) We have another attribute 'Claim Amount' which has high values consistently which makes it outlying attribute compare to other attributes. We use minMaxscaler to make it uniform.
 - 7) We have categorical and text attribute 'Injury Type'. Since model deals with numeric value, we use one-hot encoding to change them to numeric attributes.
 - 8) We have features 'NumSoftTissues', 'PerSoftTissue' which do not contribute to the model and we can drop these features.
- In this case, we use max_depth = 3 and number of features = 11. Increase in this parameter might result in overfitting.
 - We use very less percentage of data to train. This might cause our model less exposure to the complexity of the data. This might cause underfitting.

Train – 10% Test – 90% max_depth = 3 random_state = 1

Total number of records: 500

Total number of features: 11

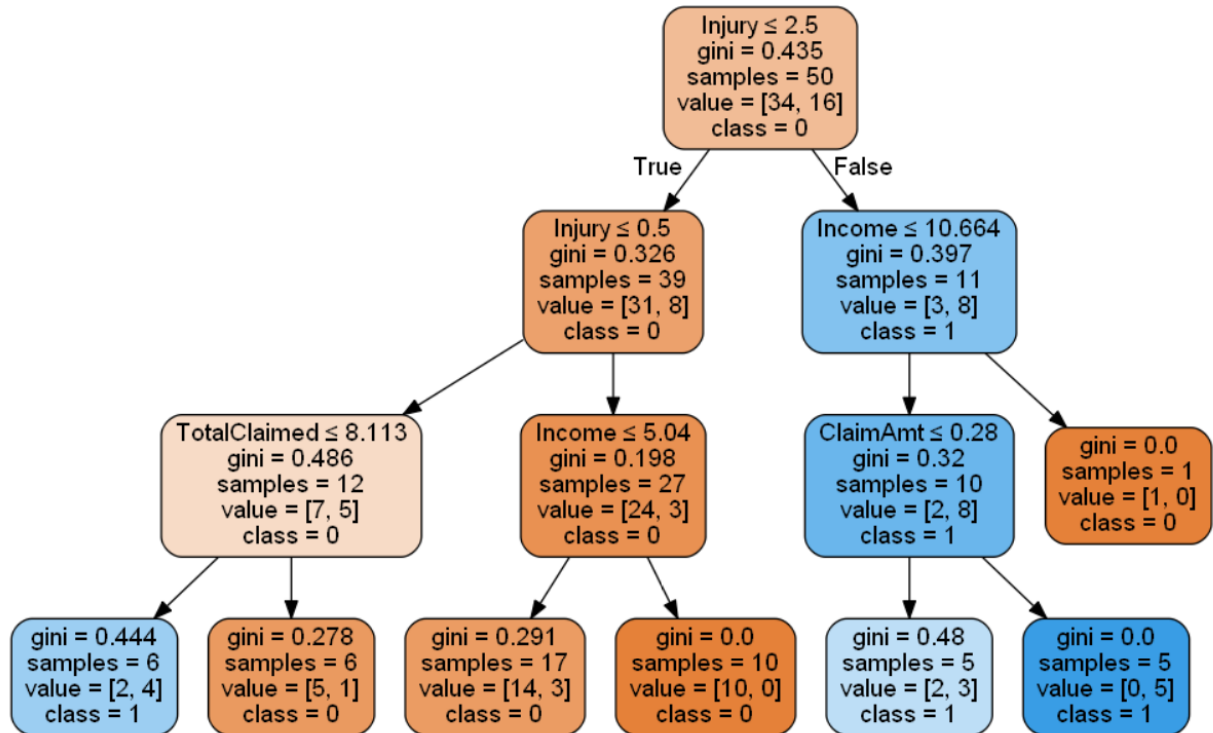
14 total features after one-hot encoding.

Training set has 50 samples.

Testing set has 450 samples.

Times for Training, Prediction: 0.00275, 0.00132

Accuracy for Training, Test sets for Gini measure: 0.84000, 0.64444



Question 3:

- 1) Pre-processing steps will remain same.

Total number of records: 500

Total number of features: 14

14 total features after one-hot encoding.

Training set has 50 samples.

Testing set has 450 samples.

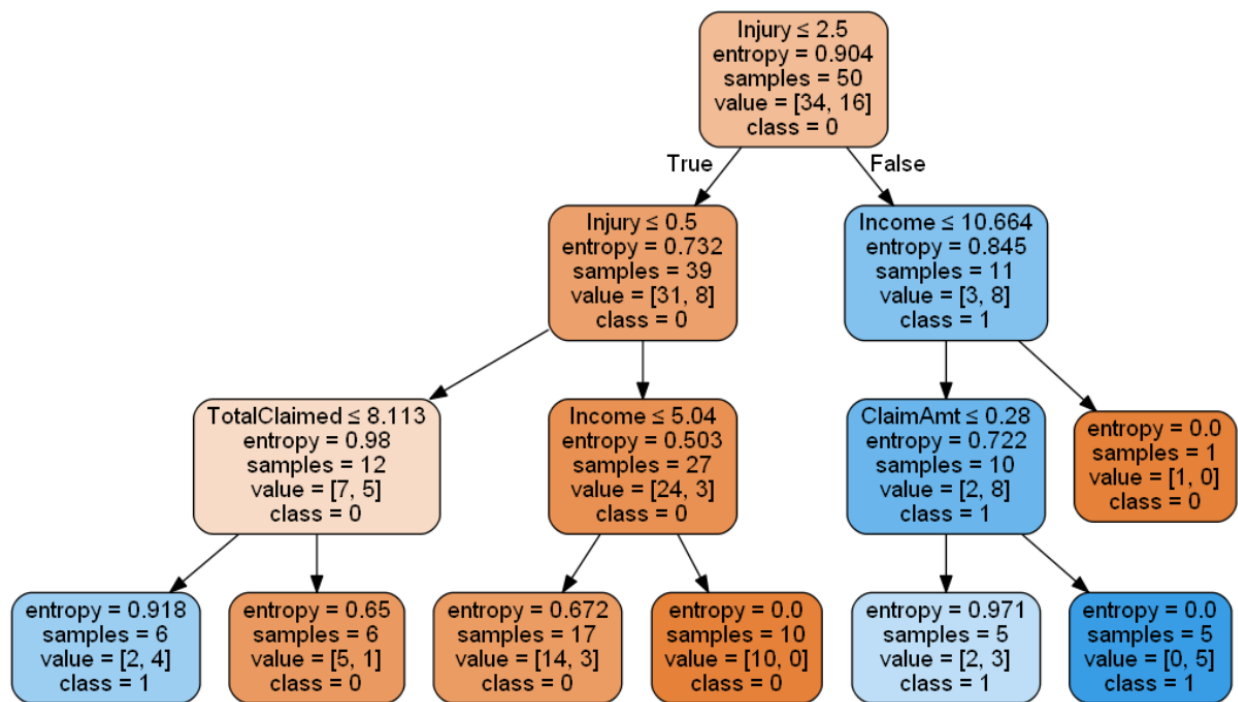
Times for Training, Prediction: 0.00275, 0.00132

Accuracy for Training, Test sets: 0.84000, 0.64444

Times for Training, Prediction: 0.00250, 0.00128

Accuracy for Training, Test sets: 0.84000, 0.64444

- Accuracy did not change even if we change the criterion to 'Entropy'.
- Tree is also did not change. Only there is slight improvement in the elapsed time.
- For the given parameters, I think both Gini and Entropy criterion perform same.



Question 4:

1) Pre-processing steps remain same

Total number of records: 500

Total number of features: 14

14 total features after one-hot encoding.

Training set has 350 samples.

Testing set has 150 samples.

Times for Training, Prediction: 0.00263, 0.00126

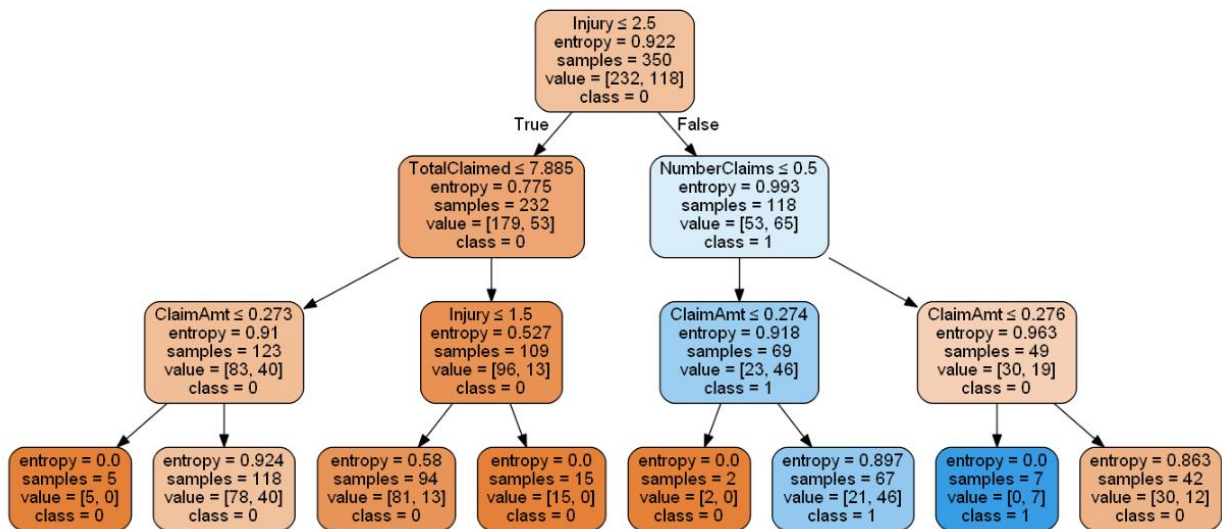
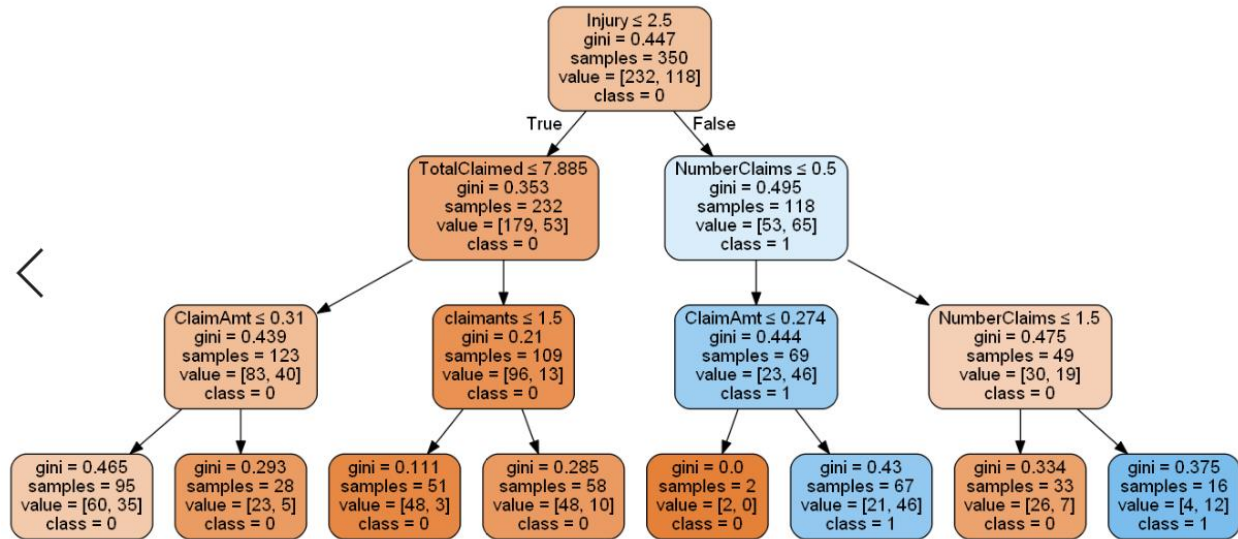
Accuracy for Training, Test sets: 0.75714, 0.69333

Times for Training, Prediction: 0.00228, 0.00127

Accuracy for Training, Test sets: 0.75429, 0.70000

-
- We can see an improvement in the performance. By increasing the training data for training. With less training data, there is a chance that model overfit on the training data and underfit for Test data.
 - The split 70-30, between training and test, is better than the 10-90
 - With these parameters, Gini measure performed well than the Entropy criteria.

- We do not see the difference in the tree structures or in accuracies, but interior nodes differ between Gini and entropy.



Question 5:

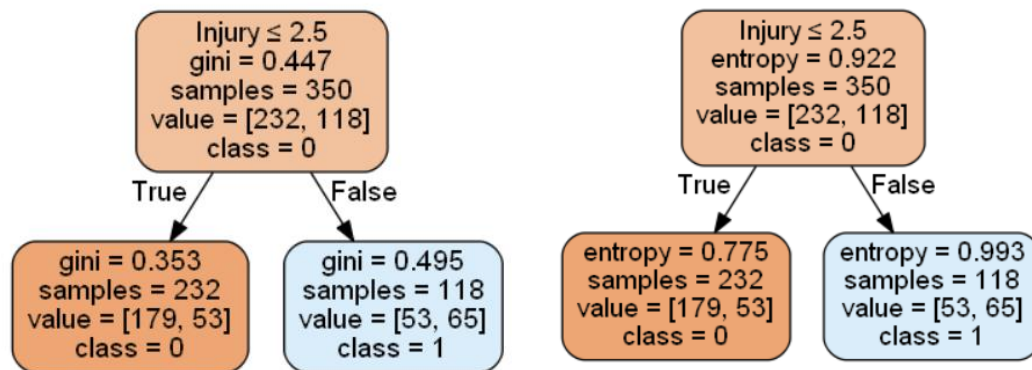
1) Pre-processing steps remain same.

Max_depth	1	2	3	4	5
Acc_Gini_Train	0.69714	0.72857	0.75714	0.78571	0.80571
Acc_Gini_Test	0.68000	0.73333	0.69333	0.67333	0.66000
Acc_Entr_Train	0.69714	0.72857	0.75429	0.76286	0.79143
Acc_Entr_Test	0.68000	0.73333	0.70000	0.66667	0.65333

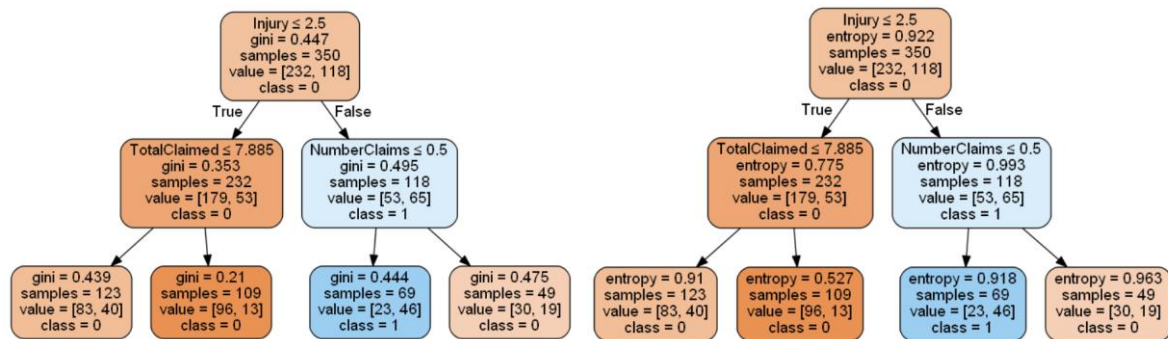
Max_depth	6	7	8	9	10	15	20
Acc_Gini_Train	0.83143	0.85429	0.90000	0.91143	0.92286	0.98286	1.00000
Acc_Gini_Test	0.65333	0.61333	0.58667	0.54667	0.53333	0.56000	0.52667
Acc_Entr_Train	0.81714	0.84286	0.86857	0.90286	0.92857	0.96000	0.99143
Acc_Entr_Test	0.62667	0.64000	0.59333	0.55333	0.58667	0.56667	0.56000

- We can see as max_depth increases, Training accuracy is also increasing
- Similarly, Test accuracy is increasing till max_depth = 2, then it is decreasing For Gini.
- For Entropy method, as max_depth increases, training accuracy increases and testing accuracy decreases.
- I think, if max_depth >3, model is overfitting on training data and underfitting on test data.

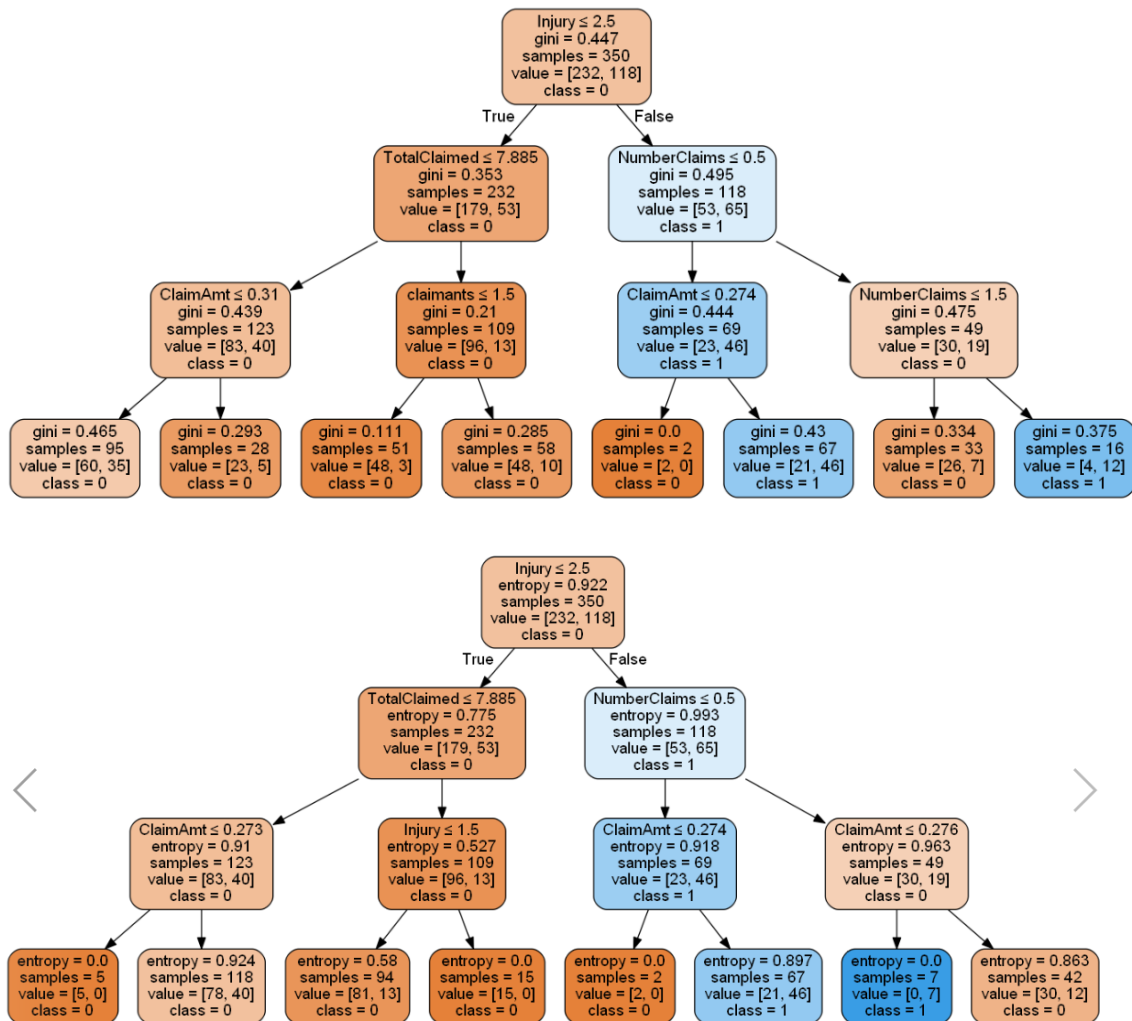
Max_depth = 1



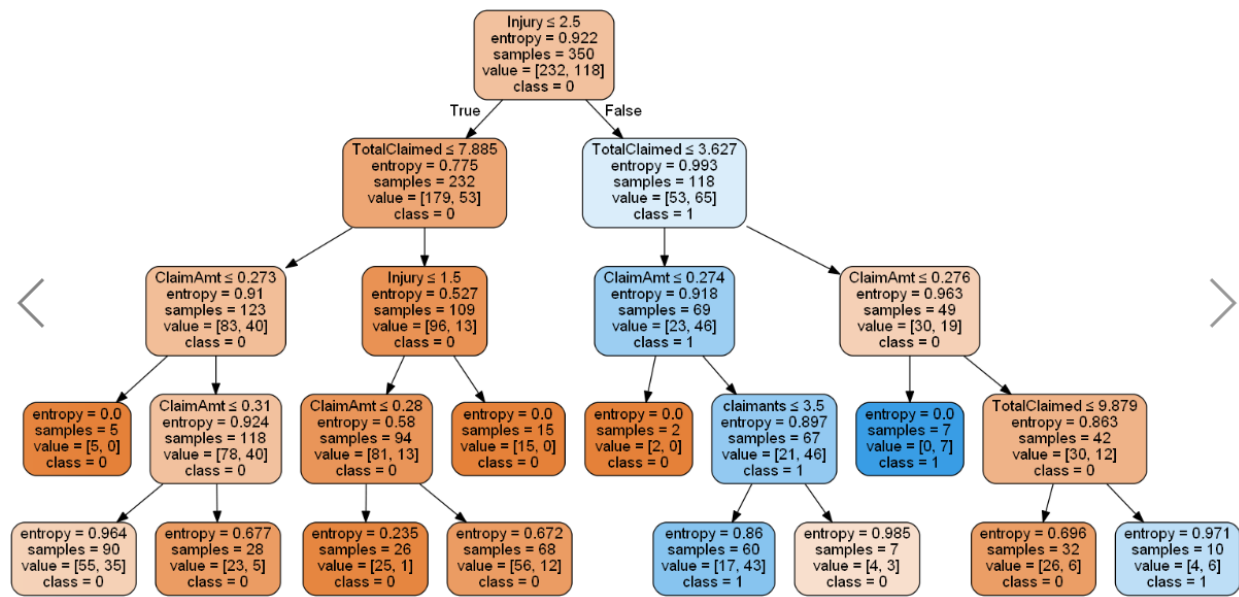
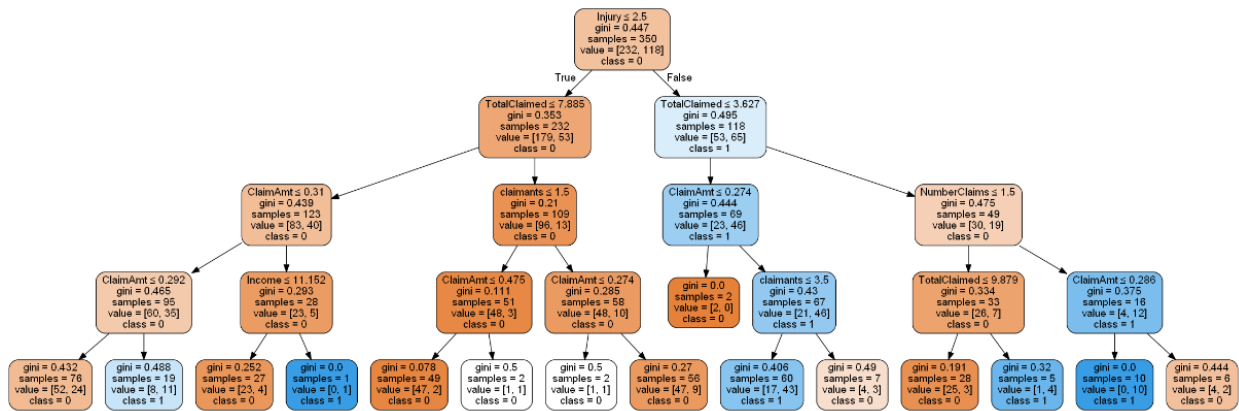
Max_depth = 2



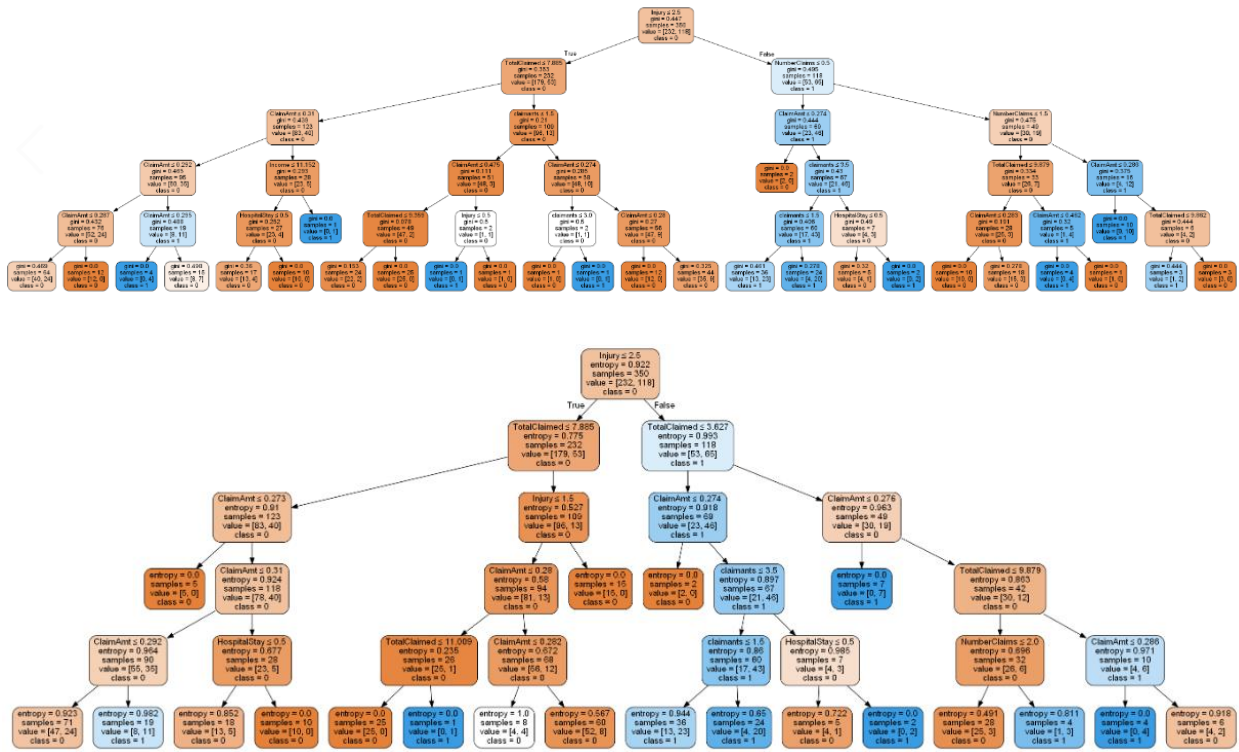
Max_depth = 3



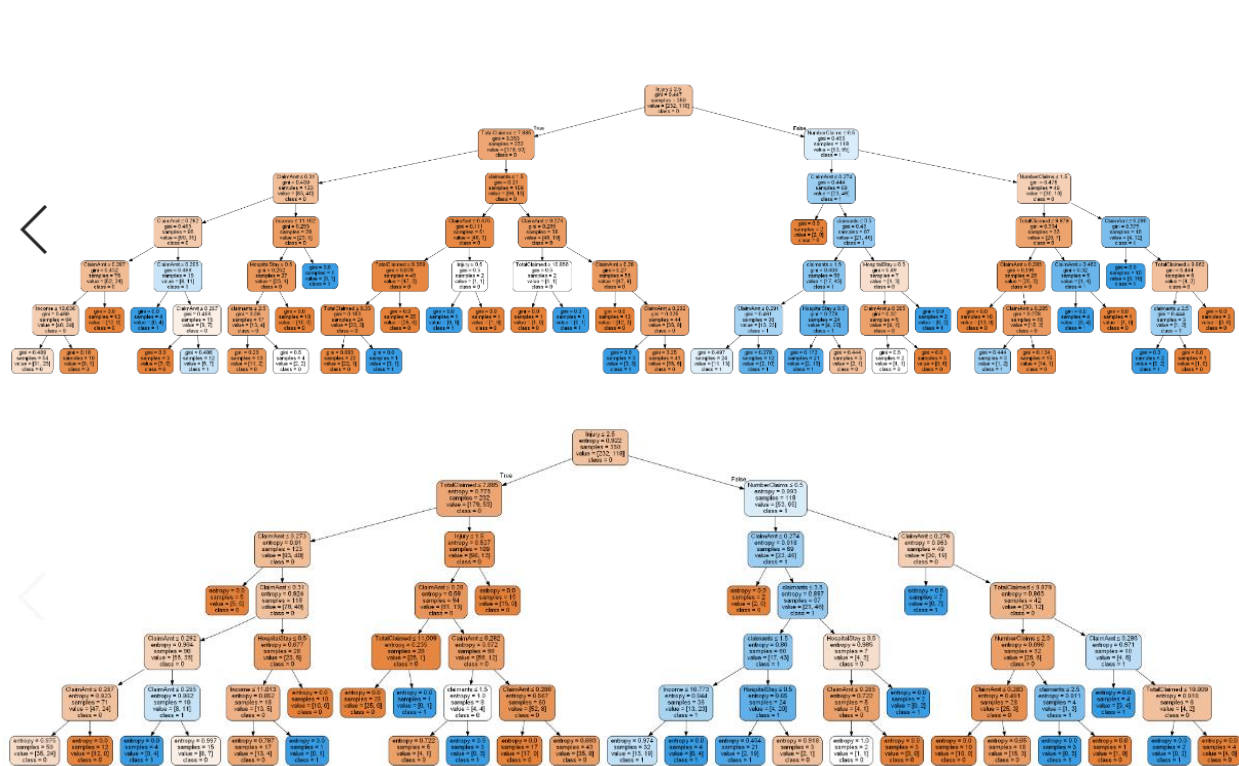
Max_depth = 4



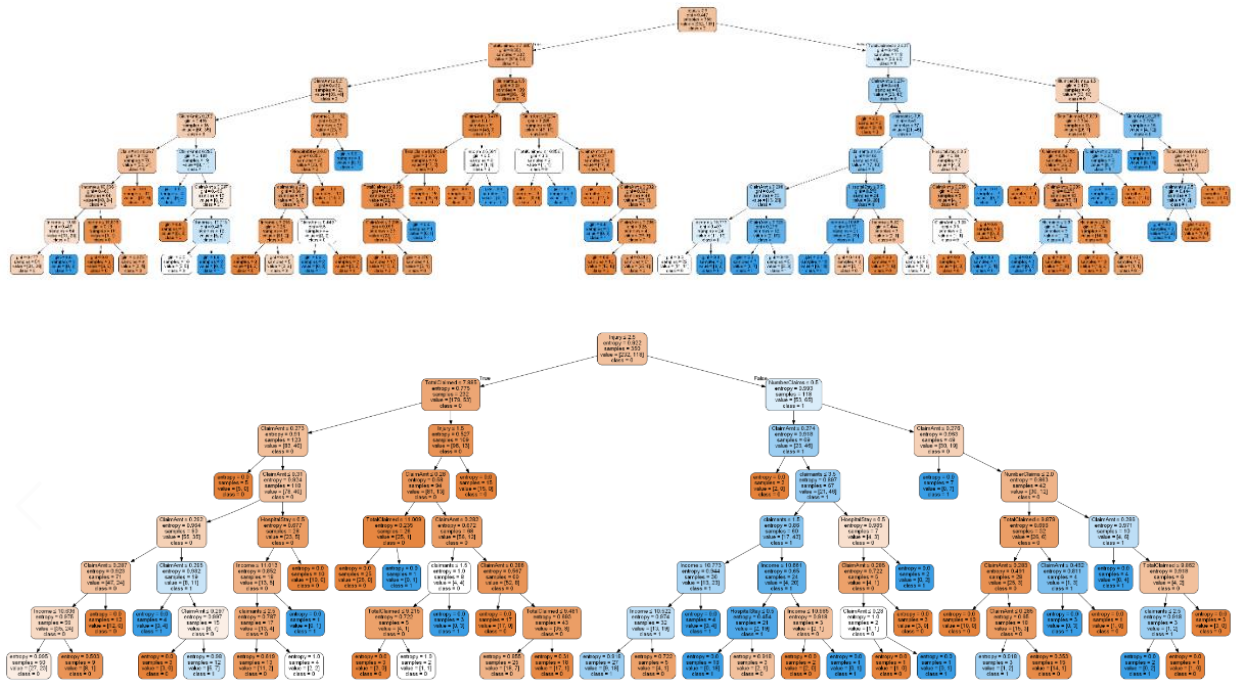
Max_depth = 5



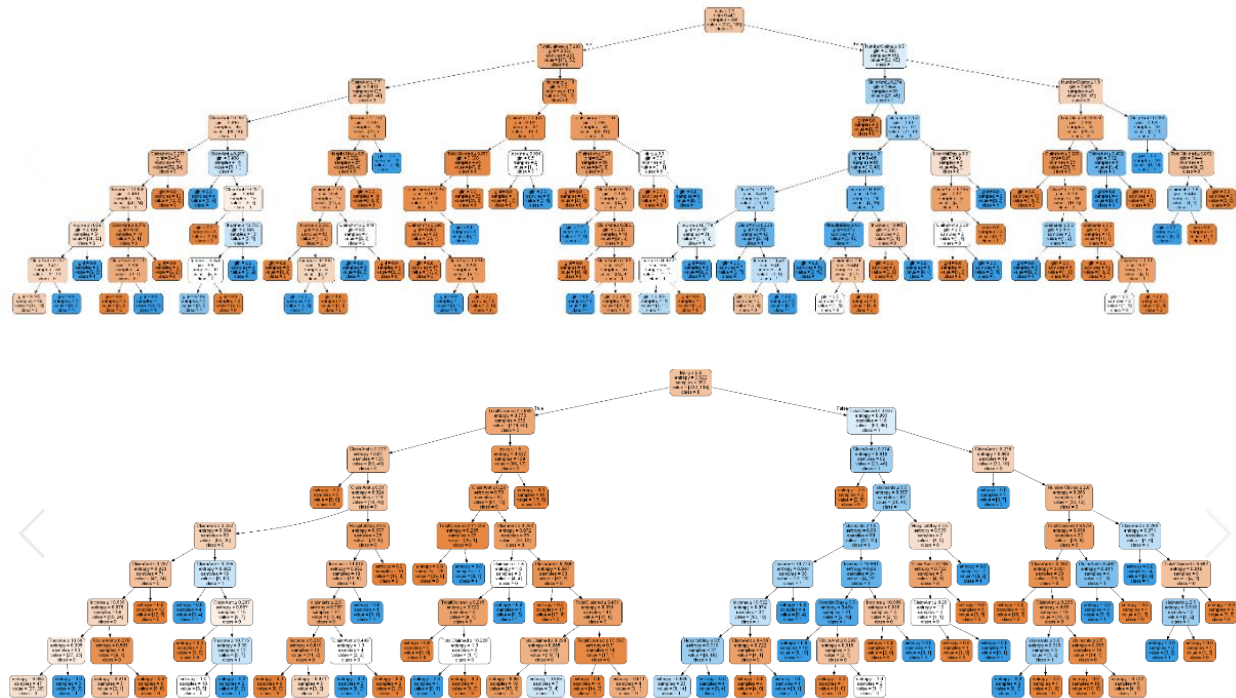
Max_depth = 6



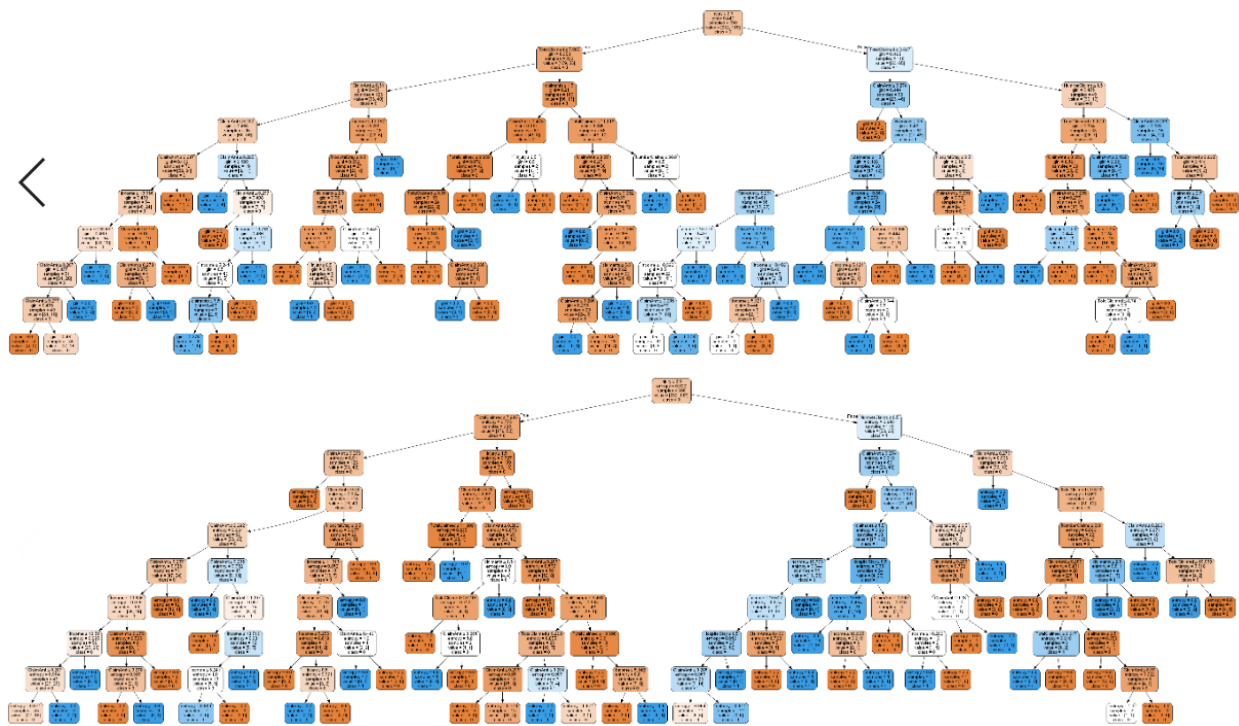
Max_depth = 7



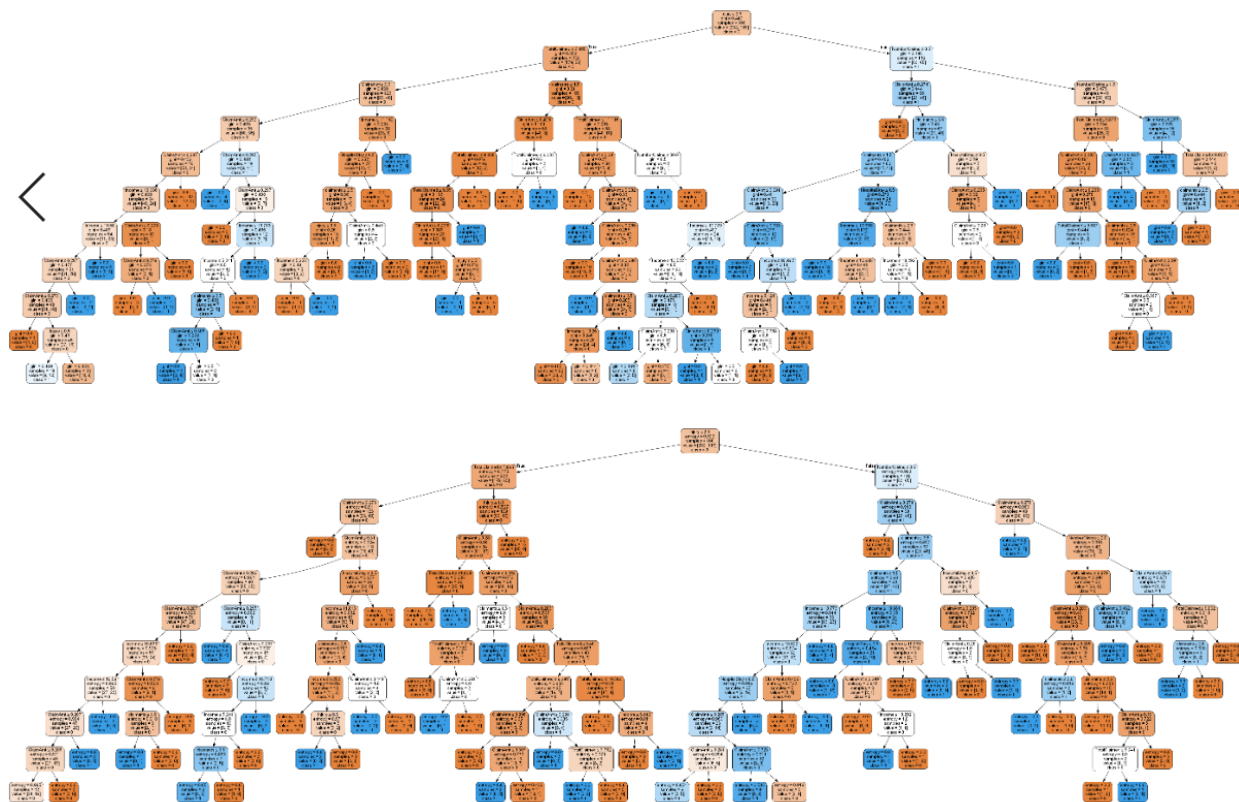
Max_depth=8



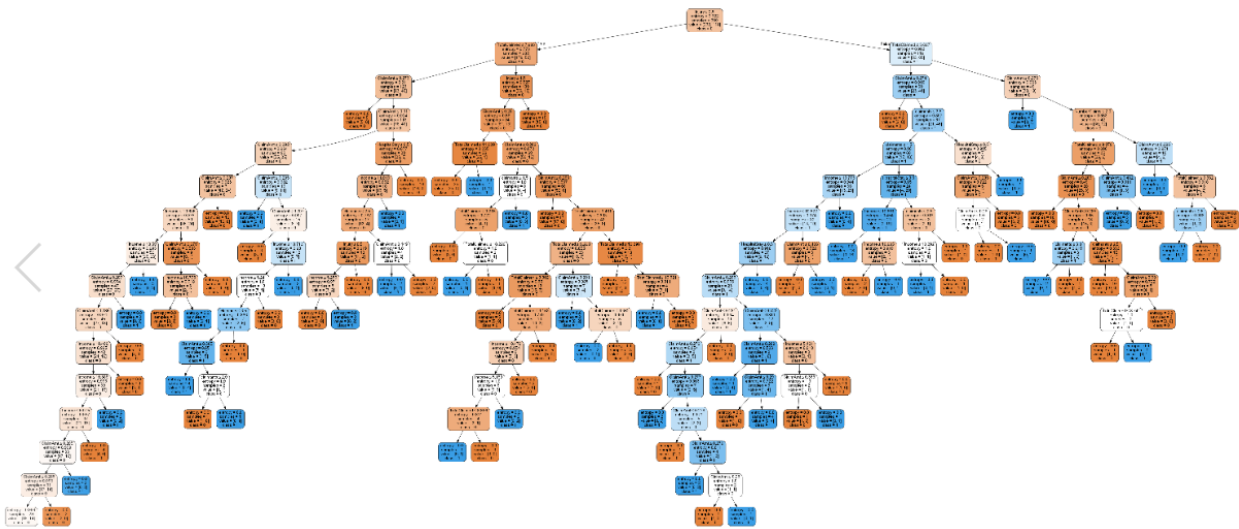
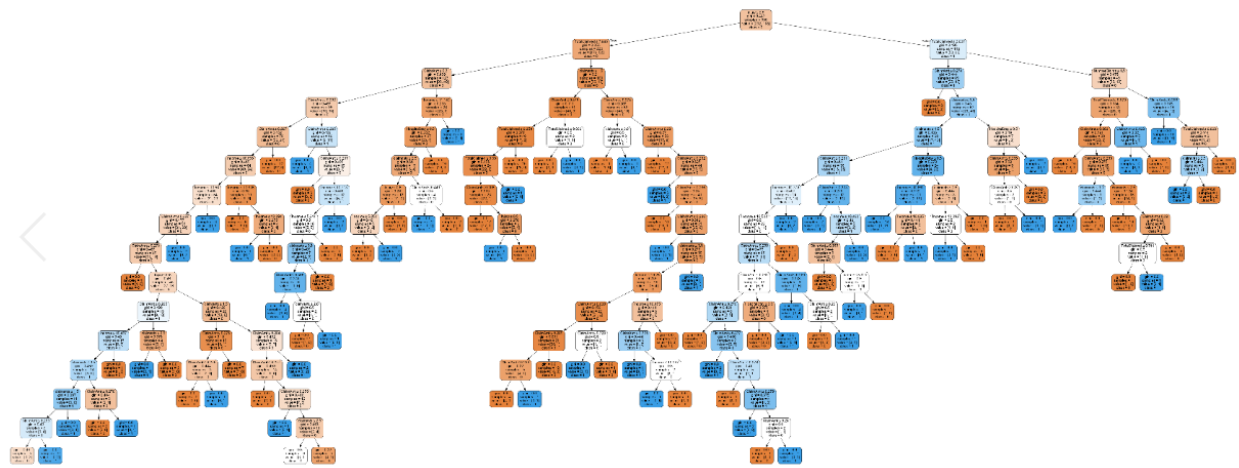
Max_depth = 9



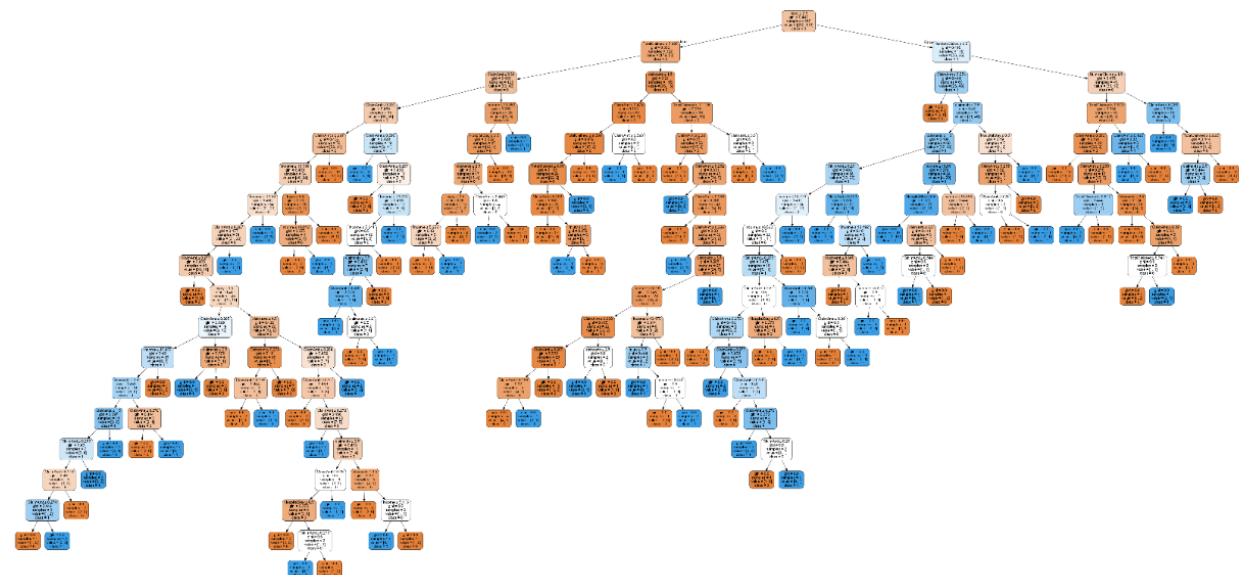
Max_depth = 10

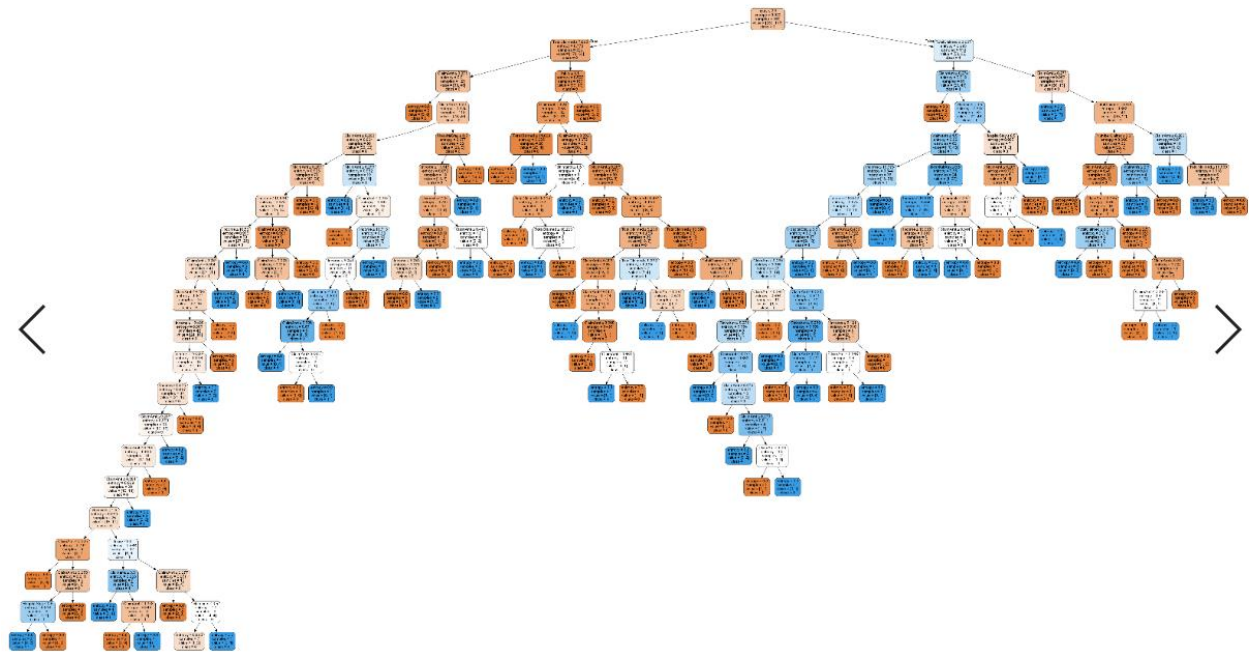


Max_depth=15



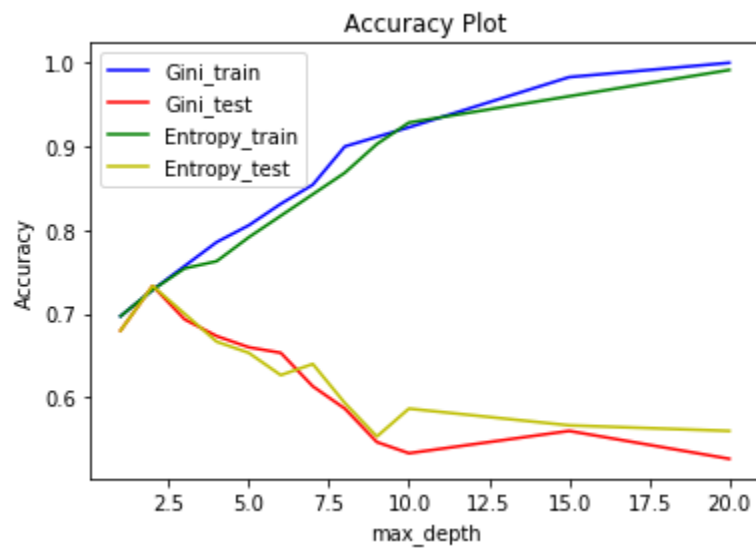
Max_depth=20





Plotting Accuracies against max_depth:

- We can see as depth increases; training accuracies increases suggesting over fit.
- Similarly, for test accuracies, after depth >3, we can see decrease in accuracies.



Question 6:

- 1) If we include all features, 'Claim Amount Received' has most entropy. But as it looked like a continuous variable and causing overfit.
- 2) After omitting above feature, 'Injury Type' has more entropy.

Question 7:

- 1) Dataset consists of some discrepancies. We have few attributes with below drawbacks
 - a. Columns having missing data
 - b. Columns having string data
 - c. Columns with skewed data
 - d. Columns that are continuous
- 2) Decision tree is sensitive to data with above properties. We can tune the overfit behavior by pre-processing above columns.
- 3) Tuning parameters of the Decision tree model are important. Parameters like, Max_depth and random_state will tune the model to suit the data better.
- 4) Splitting of Training and test data plays important role. By using less training data, model can overfit the training data and underfit on test data.
- 5) Accuracy does not represent all aspects of performances. Increase in accuracy, sometimes, points to overfitting of the data.

Question 8:

Yes. We have skewed features like 'Income of Policy holder' , 'Total Claimed amount' and 'Claim amount received. We have outliers (mostly zeros) in these features.

- a. Either we can use a mean or median value of the feature and replace the zero with that or doing some imputation.
- b. We can use clamp transformation using upper and lower threshold values.

Question 9:

Yes. I think.

Question 10:

- 1) I do not see any problems in the mentioned paper.
- 2) I think, having different amount of data will result in to two different trees, with in a given set of model parameters, only if domain knowledge of the dataset changes.
- 3) Maybe we can use the statistical measures that are sensitive to the changes happen in a feature. Like variance in both the datasets for a given feature.

