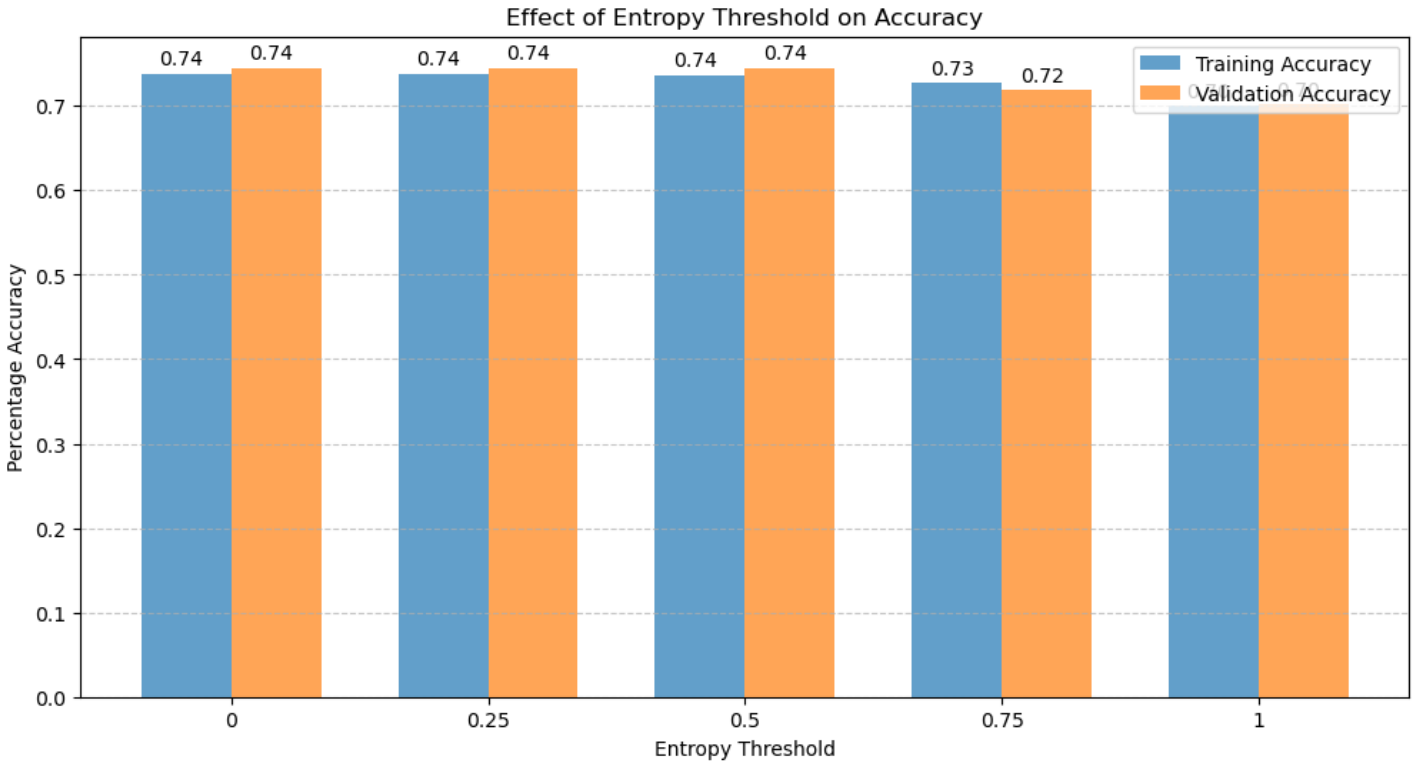
MLFA Assignment-4

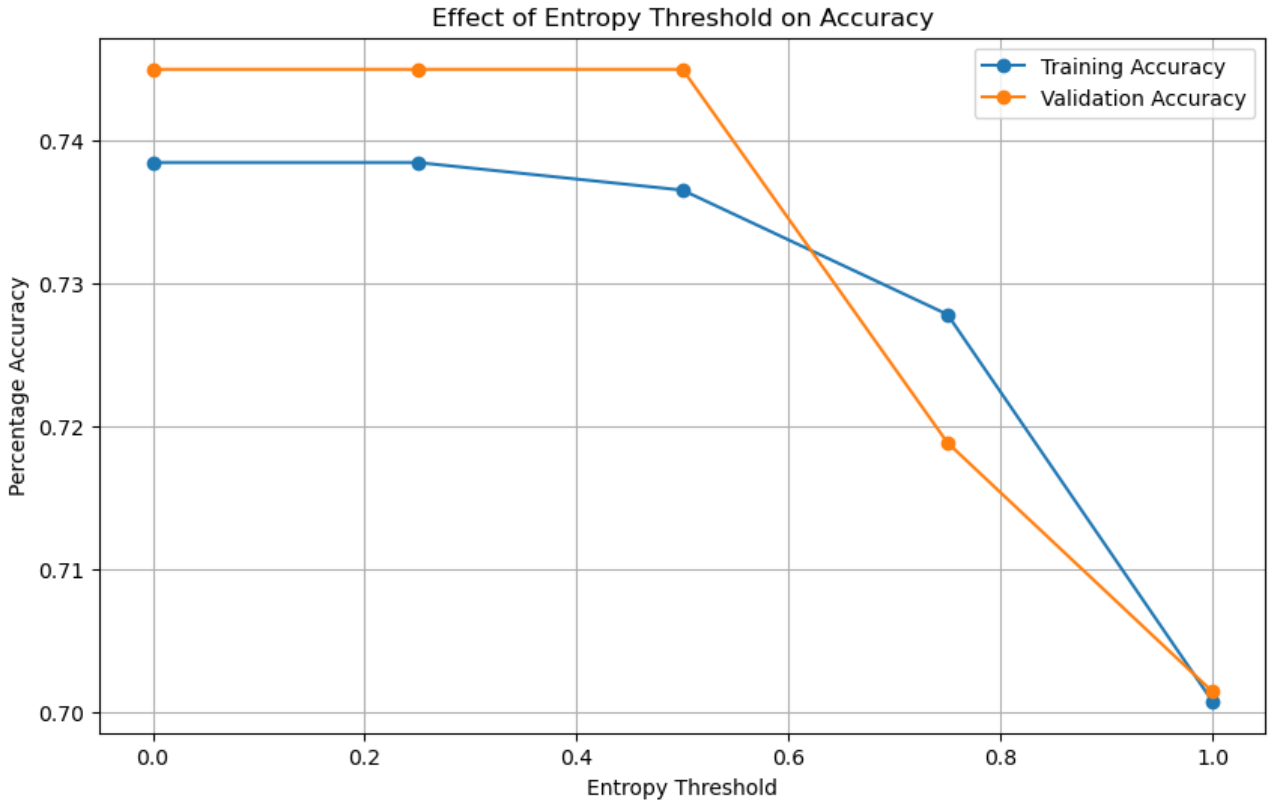
Name: Chichari Anusha **Roll:** 21CH10020

Experiment 1:

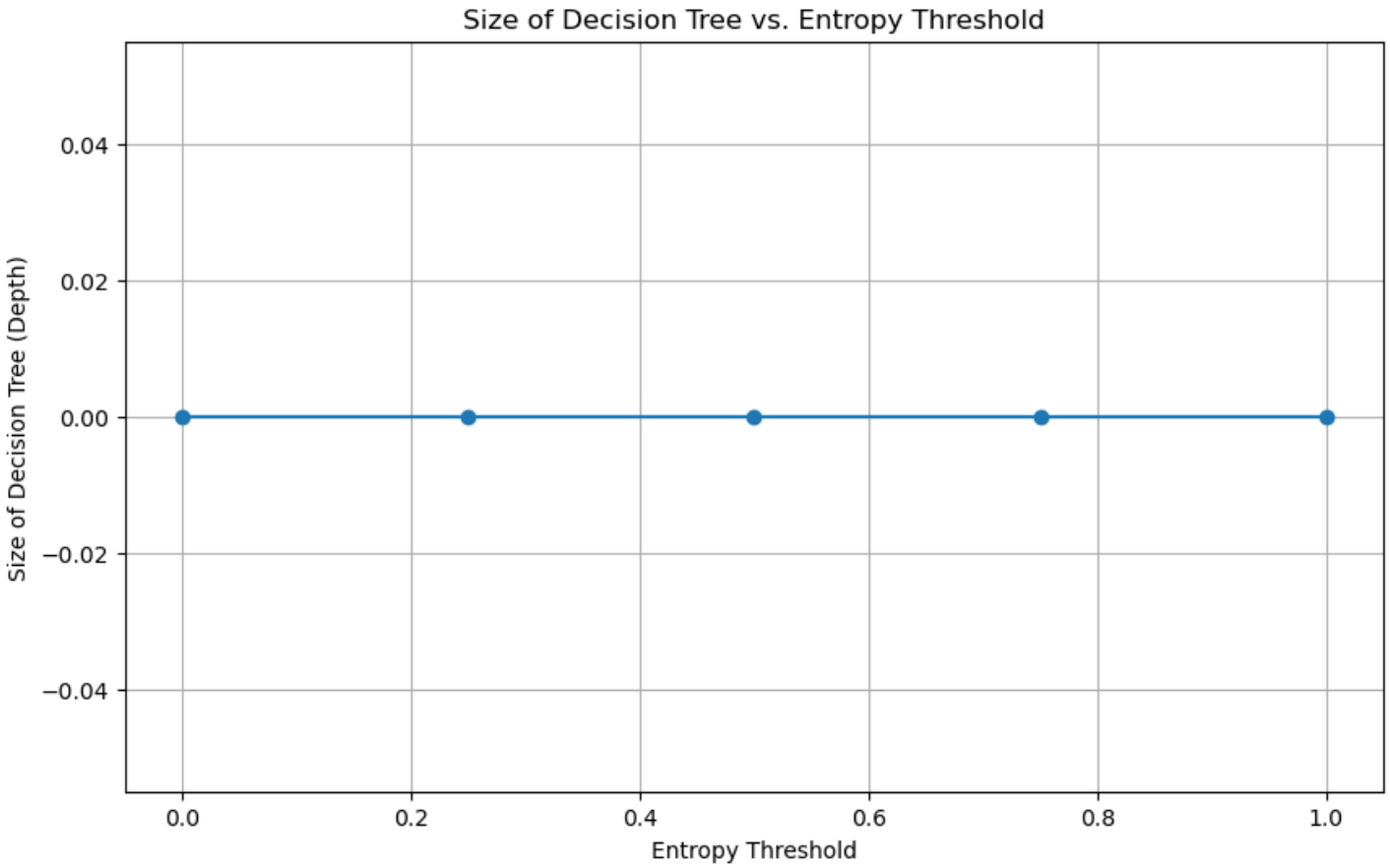
a) Plotting Percentage Accuracy vs. Threshold as a bar chart



Plotting Percentage Accuracy vs. Threshold as line chart



b) Size of Decision Tree vs. Entropy Threshold



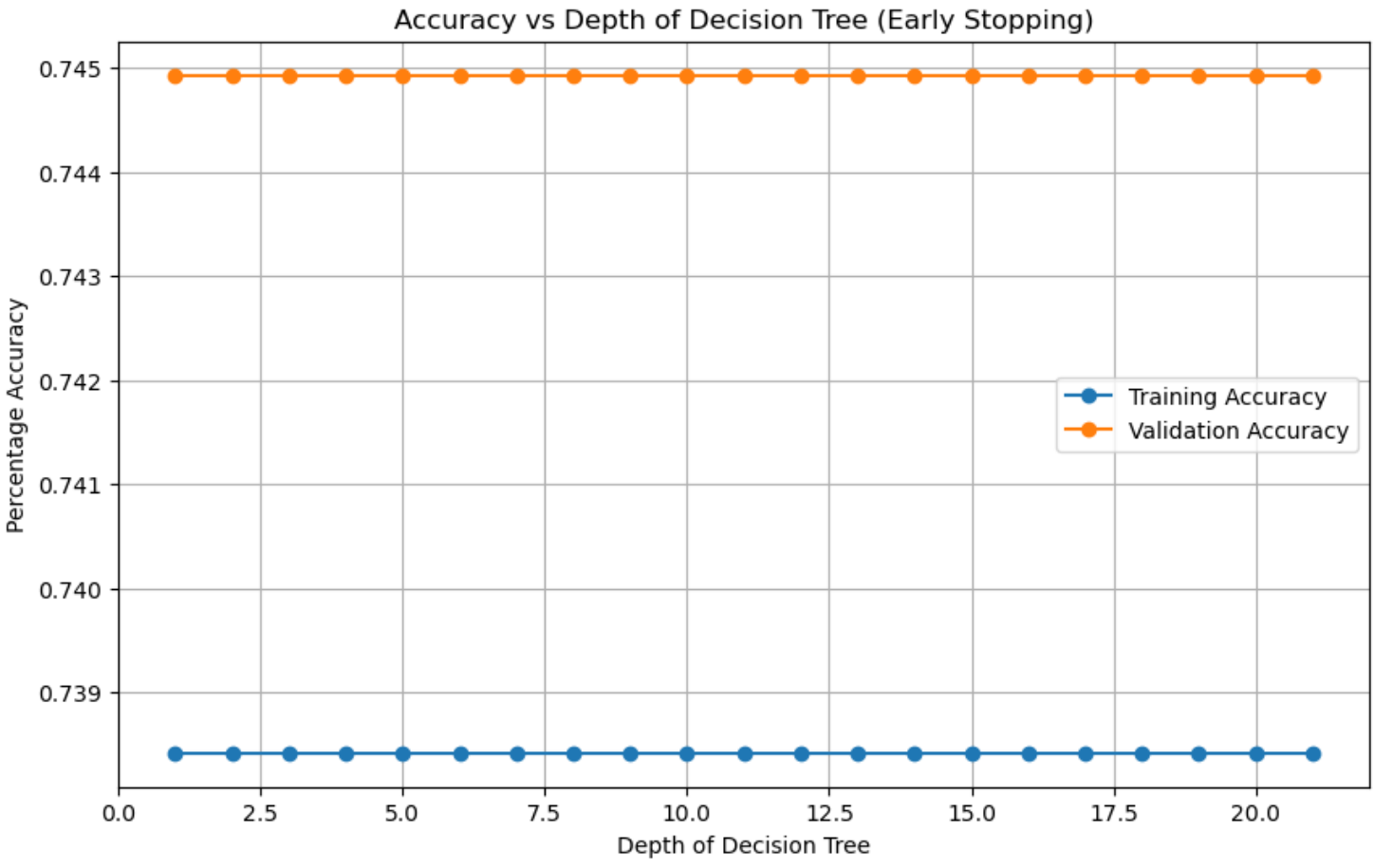
c) Optimal Entropy Threshold based on Validation Accuracy is 0.

Experiment 2:

Exp 2(a): With the optimal Entropy threshold found in the earlier experiment:
Overall Training Accuracy with Early Stopping: 0.74 (74 percent)
Overall Testing Accuracy with Early Stopping: 0.74 (74 percent)

Exp 2(b):
In Experiment 2(b), we are plotting the percentage accuracy on the training and validation datasets after every branch formation. This means that at each step, we build the decision tree by adding one more level (branch) to the tree, and then we calculate the accuracy on both the training and validation datasets.

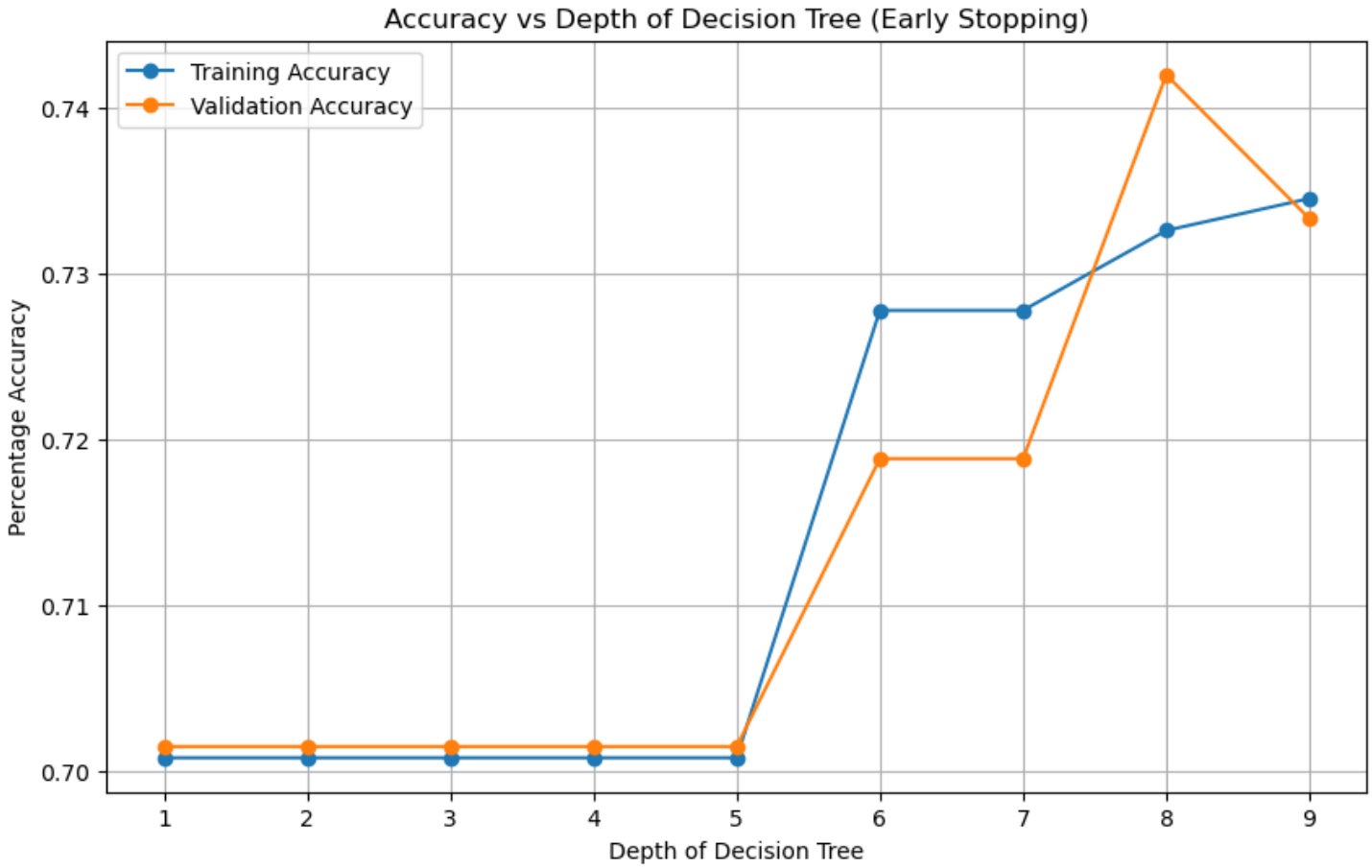
Plotting Accuracy vs Depth after every branch formation:



Exp 2(c):

In Experiment 2(c), we are stopping the tree construction when the validation accuracy starts decreasing. This means that we are effectively pruning the tree when it becomes too complex and starts overfitting.

Plotting Accuracy vs Depth:



- Total number of nodes at which correspond to the case when validation percentage starts to decrease: **511**

Validation Accuracy in Experiment 2(c): 0.73 (73 percent)
Training Accuracy in Experiment 2(c): 0.73 (73 percent)

In Experiment 2(c), the validation accuracy is 73 percent, and the training accuracy is also 73 percent. Therefore, in the new decision tree from Experiment 2(c), both the training and testing

accuracies are slightly lower than those calculated in Experiment 2(a). There is no improvement in accuracy, and the model performs slightly worse in terms of accuracy compared to the model from Experiment 2(a).

Experiment 3:

Rules for Experiment 1 (Optimal Threshold):

IF AND (Persons == 1) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND (Price Maintenance == 3) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND (Price Buying == 1) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND (Lug_boot == 1) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND (Doors == 3) AND (Doors == 2) AND (Doors == 1) AND (Persons == 2) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND (Doors == 3) AND (Doors == 2) AND (Doors == 1) AND NOT (Persons == 2) AND (Lug_boot == 2) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND (Doors == 3) AND (Doors == 2) AND (Doors == 1) AND NOT (Persons == 2) AND NOT (Lug_boot == 2) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND (Doors == 3) AND (Doors == 2) AND NOT (Doors == 1) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND (Doors == 3) AND NOT (Doors == 2) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND NOT (Doors == 3) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND (Safety == 2) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND NOT (Safety == 2) AND (Doors == 2) AND (Lug_boot == 1) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND NOT (Safety == 2) AND (Doors == 2) AND NOT (Lug_boot == 1) AND (Price Buying == 2) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND NOT (Safety == 2) AND (Doors == 2) AND NOT (Lug_boot == 1) AND NOT (Price Buying == 2) AND (Doors == 1) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND NOT (Safety == 2) AND (Doors == 2) AND NOT (Lug_boot == 1) AND NOT (Price Buying == 2) AND NOT (Doors == 1) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND NOT (Safety == 2) AND NOT (Doors == 2) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND NOT (Lug_boot == 2) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND NOT (Price Buying == 3) => THEN (Class unacc)

Rules for Experiment 2 (Early Stopping):

IF AND (Persons == 1) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND (Price Maintenance == 3) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND (Price Buying == 1) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND (Lug_boot == 1) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND (Doors == 3) AND (Doors == 2) AND (Doors == 1) AND (Persons == 2) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND (Doors == 3) AND (Doors == 2) AND (Doors == 1) AND NOT (Persons == 2) AND (Lug_boot == 2) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND (Doors == 3) AND (Doors == 2) AND (Doors == 1) AND NOT (Persons == 2) AND NOT (Lug_boot == 2) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND (Doors == 3) AND (Doors == 2) AND NOT (Doors == 1) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND (Doors == 3) AND NOT (Doors == 2) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND (Safety == 1) AND NOT (Lug_boot == 1) AND NOT (Doors == 3) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND (Safety == 2) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND NOT (Safety == 2) AND (Doors == 2) AND (Lug_boot == 1) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND NOT (Safety == 2) AND (Doors == 2) AND NOT (Lug_boot == 1) AND (Price Buying == 2) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND NOT (Safety == 2) AND (Doors == 2) AND NOT (Lug_boot == 1) AND NOT (Price Buying == 2) AND (Doors == 1) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND NOT (Safety == 2) AND (Doors == 2) AND NOT (Lug_boot == 1) AND NOT (Price Buying == 2) AND NOT (Doors == 1) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND (Lug_boot == 2) AND NOT (Safety == 2) AND NOT (Doors == 2) => THEN (Class acc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND (Price Buying == 3) AND NOT (Safety == 1) AND NOT (Lug_boot == 2) => THEN (Class unacc)

IF AND NOT (Persons == 1) AND NOT (Price Maintenance == 3) AND NOT (Price Buying == 1) AND NOT (Price Buying == 3) => THEN (Class unacc)