

Part 1:

MakeDTModel: For question one we were tasked to create a method that would create a decision tree model taking three parameters. The criterion, X_train, and Y_train were the parameters. I first used "gini" as the criterion to measure impurity, then "entropy" was used for the information gained. I then looked at the CART classifier and C5.0 classifier code in our notes to replicate using the decision tree classifier. This is where the max leaf nodes and random state could be altered. I added max_depth to the makeDTModel where it could be edited and used. When I added this I noticed that it made the data easier to read since all the nodes were not on one side.

displayTree: This method tasked us with displaying the text and graphical representation of the tree classifier. This method took a classifier and variable as parameters. I condensed the code from the slides and made it into a method that could be called with either the cart classifier or c5 classifier. This uses the sklearn.tree.plot_tree package to plot it. The max_depth attribute seems to organize the graph better.

displayCM: This method wanted us to display the text and the heatmap of the confusion matrix. This required many imports from sklearn python packages. This took both the training data and testing data along with the cart classifier.

compareModels: Here I used the notes from class to make the model for side-by-side comparisons. Before this, we had to get classification accuracy on the testing and training sets. These were put into the score variables. The python package using plt was used to plot the side by side comparison.

*I added max_depth to the makeDTModel where it could be edited and used. When I added this I noticed that it made the data easier to read since all the nodes were not on one side.

*When testing maxleafnodes = 1 python

```
ValueError: max_leaf_nodes 1 must be either None or larger than 1  
a
```