**Program Description**:

First, all the missing values are replaced with NaN. All the attributes are then labelled from A to P. Each column is sorted alphabetically/numerically in ascending order in order to find the median. The rows which have NaN are ignored when computing the median. The computed median replaces the NaNs in the training and test data for their respective columns. The program has cells which print out the number of missing values in each column for the training set along with the index which contains the median and the actual median itself.

The pos function calculates the number of positive values for a given categorical attribute. The pos2 function calculates the number of positive values below or equal to a certain value. The pos3 function calculates the number of positive values greater than a certain value. The pos4 function calculates the number of positive values for a parent dataset.

The entropy function calculates the entropy. The iv function calculates the intrinsic value of a split. The gini function calculates the gini index. The info\_gain function calculates the information gain.

The cat\_split, cat\_split4, cat\_split5, cat\_split6, cat\_split7, cat\_split9, and cat\_split13 are all used to help to calculate the weighted entropy of their respective categorical attributes. These functions will be called in the tree function to create the decision tree for the C4.5 algorithm.

The cat2\_split, cat2\_split4, cat2\_split5, cat2\_split6, ca2t\_split7, cat2\_split9, and cat2\_split13 are all used to help to calculate the weighted gini indices of their respective categorical attributes. These functions will be called in the tree2 function to create the decision tree for the CART algorithm.

The con\_split function helps to find the best split value out of a set of candidate split values. The con\_split2 function formally splits a dataset of continuous values according to the chosen split value and helps to compute their weighted entropy. These functions will be called in the tree function to create the decision tree for the C4.5 algorithm.

The con2\_split function helps to find the best split value out of a set of candidate split values. The con2\_split2 function formally splits a dataset of continuous values according to the chosen split value and helps compute their weighted gini indices. These functions will be called in the tree2 function to help create the decision tree for the CART algorithm.

The nsplit1 function puts into a list, all the values less than or equal to a given value. The nsplit2 function puts into a list, all the values greater than a given value.

The Node class defines the nodes that will make up the tree. It includes the indices that are part of its dataset, the attributes that have not been used yet, its respective children, the attribute in which its children node have been split by, the decision (+ or -) at that node in case it is a leaf node, and finally a variable that points back its parent node.

**Note**: Each node has been given a maximum of 14 children because this is the maximum number of children a node will have as Attribute 6 (the attribute with the most categories) of our dataset has a total of 14 categories.

The tree function is the function which builds the decision tree for the C4.5 algorithm. Instead of a single IMPORTANCE function, the tree function calls upon an assortment of the above mentioned functions which together serve as the IMPORTANCE method for this decision tree.

The tree2 function is the function which builds the decision tree for the CART algorithm. Instead of a single IMPORTANCE function, the tree2 function calls upon an assortment of the above mentioned functions which together serve as the IMPORTANCE method for this decision tree.

The testing function traverses the decision tree of the best model to predict the outcome of the given row of the test set. The validate function traverses the decision tree of a model to predict the outcome of the given row of the validation set.

Some of the nodes in a decision tree have been incorrectly given a decision/outcome value even though they are not leaf nodes. This occurs due to one of the node’s being empty, and according to how the tree/tree2 function runs, this node would be given a decision value. The cleaner method rectifies this problem and removes the decision value of the incorrectly identified leaf nodes.

**Results**:

**Models Produced by C4.5 Algorithm**

Model 1

- -

- -

+ +

- -

- -

- +

+ +

- -

+ +

- -

+ -

+ +

+ +

- +

- +

+ +

- -

- -

+ +

- -

- -

+ +

+ +

+ -

- -

- -

+ +

- -

+ +

+ +

+ +

- -

+ -

- -

- -

+ +

- -

- -

+ +

+ +

- +

- +

- -

- +

+ +

+ +

+ +

- -

- -

+ +

- -

+ +

- -

- -

- -

0.823529411764706

Model 2

+ -

+ +

+ -

- +

+ -

- +

+ +

+ +

+ +

- +

+ +

- +

- +

- -

- +

- +

- +

- +

- -

+ +

- -

- -

- -

- -

+ +

- -

+ -

- -

+ +

+ +

+ +

- -

+ +

+ +

+ -

- -

- -

- +

- -

- +

+ +

- -

+ -

+ +

- -

+ -

- -

- +

- -

+ +

- -

- -

- +

- -

+ +

0.6153846153846154

Model 3

+ -

- -

- +

- -

+ +

- +

- -

+ +

+ -

- -

+ +

- -

+ -

+ +

- -

+ +

- +

- -

+ +

- -

- +

+ +

+ +

- -

- -

- +

- -

+ +

+ -

+ -

- -

- -

- -

- +

+ -

+ -

+ +

- -

+ -

- -

+ +

+ -

- -

+ +

+ +

+ +

+ -

- -

- +

+ +

- -

+ +

- -

+ +

- -

0.6666666666666667

Model 4

+ -

- -

- +

- -

+ +

- -

- -

+ +

+ +

+ +

- -

+ +

+ +

- -

- -

- -

- -

+ +

+ +

+ -

- +

- -

+ -

+ +

- -

+ -

+ +

+ -

+ +

+ +

- -

- +

+ +

- -

+ +

- -

- -

+ -

- -

- +

- -

+ +

+ +

- -

- -

+ +

- -

+ +

+ +

+ +

+ +

- -

- -

- +

+ +

0.7999999999999999

Model 5

+ +

- -

+ -

- -

+ -

- -

+ -

+ +

- -

- -

+ +

+ +

- -

+ +

- -

- -

+ +

- -

+ +

- -

- -

+ +

- -

- -

- -

+ +

- -

+ +

- -

- -

+ +

+ +

+ -

- -

- -

+ -

- -

+ +

+ +

+ -

+ +

- +

- -

- -

+ -

- -

- -

- -

+ -

- -

+ +

- -

- -

+ +

- -

0.7906976744186047

Model 6

- -

- -

- -

+ +

+ +

- +

+ +

+ +

+ +

+ +

+ -

- -

- -

+ +

+ +

+ +

- -

- +

- -

+ +

- -

+ +

+ +

+ +

+ +

- -

- -

- -

- +

- -

+ +

+ +

+ -

+ +

+ +

- -

- -

+ -

- -

+ +

+ +

- -

- -

+ +

+ +

- -

+ +

+ -

- +

- -

+ +

- -

+ +

+ -

+ +

0.8524590163934426

Model 7

+ +

+ +

- -

+ -

+ +

- -

- -

- -

- -

+ +

+ +

- -

+ +

+ +

+ +

+ +

- +

+ +

- -

- -

- -

+ +

- -

+ +

- -

+ -

+ -

+ -

+ -

- -

- -

- -

+ +

- -

- -

+ +

- +

- -

+ +

- -

- -

- -

+ +

+ +

- -

+ -

+ +

- -

- -

+ -

- -

- -

+ -

- +

- -

0.7659574468085107

Model 8

+ +

+ +

- -

+ +

+ +

- -

- -

- -

+ +

+ +

+ -

- -

- -

- -

- +

- -

- -

- -

- +

+ +

- -

- -

+ +

- -

+ +

- -

- +

+ +

- -

+ +

- -

- -

+ +

+ +

- -

+ -

+ -

+ +

+ -

- -

- -

+ +

+ -

+ +

+ -

+ +

+ +

+ +

- -

- +

- -

- -

- +

+ +

- -

0.7843137254901961

Model 9

- +

- -

- -

- -

- -

- -

- -

+ +

- -

+ +

- -

+ +

+ -

- -

- -

- -

- +

- -

+ +

- -

- -

- -

+ +

+ +

- -

+ +

+ +

+ +

+ +

- -

- -

+ +

- -

- -

- -

+ +

- -

- +

+ +

- -

+ +

- -

- -

- -

- -

+ -

- +

- -

- -

+ +

+ +

- -

- -

- +

+ +

0.8292682926829269

Model 10

- -

+ +

+ -

+ +

+ +

+ +

- -

- -

+ +

- +

- -

- -

+ -

+ -

- -

- +

- -

- -

+ +

+ -

- -

+ -

+ +

+ +

- -

- -

+ -

- -

- -

+ +

- -

+ -

- -

- -

- -

- -

+ +

+ +

- +

- -

- +

- -

+ -

+ +

- -

- -

+ +

+ +

- -

+ +

+ +

- -

- -

+ +

- -

0.7391304347826089

Best Model: Model 6

0.8524590163934426

The left column represents the predicted outcome while the right column represents the actual outcome. The model number here also indicates the fold which was used as a test set. For example, Model 2 used fold2 as the test set (examples 56-110) while the rest of the examples were used as a training set to train the model. We can see that the models produced by the C4.5 algorithm produce F1 values that are relatively close to each other, within the range of 0.615 to 0.852. The model with the best outcome here is the sixth model.

**Models Produced by CART Algorithm**

Model 1

+ -

- -

+ +

- -

- -

+ +

+ +

+ -

+ +

- -

- -

+ +

+ +

- +

+ +

+ +

- -

- -

+ +

- -

- -

+ +

+ +

+ -

- -

- -

- +

- -

+ +

+ +

+ +

- -

- -

- -

- -

+ +

- -

- -

+ +

+ +

+ +

+ +

- -

+ +

+ +

+ +

+ +

- -

- -

+ +

- -

+ +

- -

- -

- -

0.9090909090909091

Model 2

+ -

- +

+ -

- +

- -

+ +

+ +

+ +

+ +

+ +

+ +

+ +

+ +

- -

+ +

+ +

+ +

+ +

- -

+ +

- -

+ -

- -

- -

+ +

- -

+ -

- -

+ +

+ +

+ +

+ -

+ +

+ +

+ -

- -

- -

- +

- -

- +

+ +

- -

+ -

+ +

- -

+ -

+ -

+ +

+ -

+ +

- -

- -

- +

- -

+ +

0.7619047619047619

Model 3

- -

- -

+ +

+ -

- +

+ +

- -

+ +

- -

- -

- +

- -

- -

+ +

- -

+ +

- +

- -

+ +

- -

- +

- +

- +

- -

- -

- +

- -

+ +

- -

- -

+ -

+ -

- -

- +

- -

+ -

+ +

- -

+ -

+ -

+ +

+ -

- -

+ +

+ +

+ +

- -

- -

+ +

- +

+ -

- +

- -

+ +

+ -

0.5957446808510638

Model 4

+ -

+ -

- +

- -

- +

+ -

- -

+ +

+ +

+ +

- -

+ +

+ +

- -

+ -

- -

- -

+ +

+ +

+ -

- +

+ -

- -

+ +

- -

- -

- +

- -

+ +

+ +

- -

- +

+ +

- -

- +

- -

- -

+ -

- -

+ +

- -

+ +

+ +

- -

- -

+ +

- -

- +

- +

+ +

- +

- -

- -

+ +

+ +

0.6923076923076923

Model 5

+ +

- -

+ -

- -

+ -

- -

+ -

+ +

- -

- -

+ +

+ +

+ -

- +

- -

- -

+ +

- -

- +

- -

- -

+ +

- -

- -

+ -

+ +

- -

+ +

- -

- -

- +

- +

- -

- -

- -

+ -

- -

+ +

+ +

+ -

+ +

+ +

- -

- -

- -

- -

- -

- -

+ -

- -

- +

- -

- -

+ +

+ -

0.65

Model 6

- -

- -

- -

+ +

- +

+ +

- +

- +

+ +

+ +

+ -

- -

- -

- +

- +

+ +

- -

- +

- -

- +

- -

- +

+ +

+ +

+ +

- -

- -

- -

- +

- -

- +

- +

+ -

- +

+ +

- -

- -

- -

+ -

- +

- +

- -

- -

- +

- +

- -

+ +

- -

- +

- -

- +

- -

+ +

- -

- +

0.5

Model 7

+ +

+ +

- -

+ -

+ +

- -

- -

+ -

- -

+ +

+ +

+ -

- +

+ +

- +

+ +

- +

- +

+ -

- -

- -

+ +

- -

+ +

- -

+ -

- -

+ -

+ -

+ -

- -

- -

+ +

+ -

- -

+ +

- +

+ -

+ +

+ -

- -

- -

- +

+ +

+ -

+ -

+ +

- -

- -

+ -

- -

+ -

+ -

- +

+ -

0.5384615384615384

Model 8

+ +

+ +

- -

- +

- +

+ -

- -

- -

+ +

+ +

- -

+ -

- -

- -

+ +

+ -

+ -

- -

- +

+ +

+ -

- -

+ +

+ -

- +

- -

- +

- +

- -

+ +

- -

- -

- +

+ +

- -

+ -

- -

- +

+ -

- -

- -

- +

- -

+ +

+ -

+ +

- +

+ +

- -

- +

- -

- -

+ +

+ +

- -

0.5833333333333334

Model 9

- +

+ -

- -

- -

- -

- -

+ -

+ +

+ -

- +

+ -

+ +

- -

+ -

- -

- -

- +

- -

- +

+ -

+ -

+ -

+ +

+ +

- -

+ +

+ +

+ +

+ +

+ -

+ -

+ +

+ -

+ -

+ -

+ +

- -

- +

+ +

+ -

+ +

- -

+ -

- -

+ -

+ -

+ +

- -

+ -

+ +

+ +

- -

- -

- +

+ +

0.5714285714285714

Model 10

- -

- +

- -

- +

- +

+ +

+ -

- -

- +

- +

- -

+ -

+ -

- -

- -

+ +

- -

+ -

- +

+ -

- -

- -

- +

+ +

- -

- -

+ -

+ -

+ -

+ +

- -

+ -

+ -

- -

+ -

+ -

- +

- +

- +

- -

+ +

- -

- -

+ +

- -

- -

- +

+ +

+ -

+ +

- +

+ -

- -

- +

- -

0.37209302325581395

Best Model: Model 1

0.9090909090909091

We can see that the models produced by the C4.5 algorithm produce F1 values that are relatively far from each other, within the range of 0.372 to 0.909. The model with the best outcome here is the sixth model.

**Result of Running Test Data on Best Model Produced by C4.5 Algorithm**

- -

+ +

+ +

- +

- -

- -

- -

+ +

- +

- -

- -

+ +

+ +

+ +

- -

- -

+ -

+ +

+ +

- -

- -

- +

- -

+ -

+ +

- -

+ +

- -

- -

- -

- -

+ -

+ +

- -

- -

- -

+ +

+ +

+ +

- -

+ -

+ -

+ +

+ +

- +

+ +

+ +

+ +

- -

+ +

- -

- -

+ +

- -

+ +

- -

- -

+ +

+ +

- -

+ +

- -

+ +

+ +

- -

- -

+ +

+ +

+ +

- -

+ +

- -

+ +

- -

+ +

+ -

- -

+ +

- -

- -

- +

- -

- -

+ +

- -

+ +

+ +

+ -

- -

+ +

- -

- +

+ -

- +

+ +

+ +

- -

+ +

- -

- -

- -

- -

- -

+ +

+ +

+ +

- -

- +

- -

- -

- -

- +

- -

- -

- -

- -

- -

- -

- -

- -

+ +

- -

+ -

- -

- -

+ +

- +

- -

+ +

- -

- +

+ +

- +

+ +

+ -

- -

+ -

+ +

- +

- -

F1 Score:

0.8064516129032259

**Result of Running Test Data on Best Model Produced by CART Algorithm**

- -

+ +

- +

- +

- -

- -

- -

+ +

+ +

- -

- -

+ +

- +

+ +

- -

- -

- -

+ +

+ +

- -

- -

+ +

- -

+ -

- +

- -

+ +

- -

- -

+ -

- -

+ -

- +

- -

- -

- -

+ +

+ +

- +

- -

- -

+ -

- +

+ +

- +

+ +

+ +

- +

- -

+ +

- -

- -

+ +

- -

+ +

- -

- -

+ +

- +

- -

- +

- -

+ +

+ +

- -

- -

+ +

+ +

+ +

- -

+ +

- -

+ +

- -

+ +

- -

- -

+ +

- -

- -

- +

- -

- -

+ +

- -

- +

+ +

+ -

- -

+ +

- -

+ +

+ -

+ +

+ +

+ +

- -

- +

- -

- -

- -

- -

- -

- +

+ +

+ +

- -

- +

- -

- -

- -

+ +

+ -

- -

- -

- -

- -

- -

- -

- -

+ +

- -

+ -

- -

- -

+ +

+ +

- -

+ +

- -

- +

+ +

- +

+ +

+ -

+ -

- -

+ +

+ +

- -

F1 Score:

0.7627118644067797

From running the test data on the best models produced by the C4.5 and CART algorithm, we can see that the C4.5 model produces a better F1 score (0.806) when compared to the CART algorithm (0.763). We can also say from our cross-validation results that the C4.5 algorithm produced higher quality models than those produced by the CART algorithm. As such it is fair to say that the C4.5 algorithm produced better results for this particular dataset.