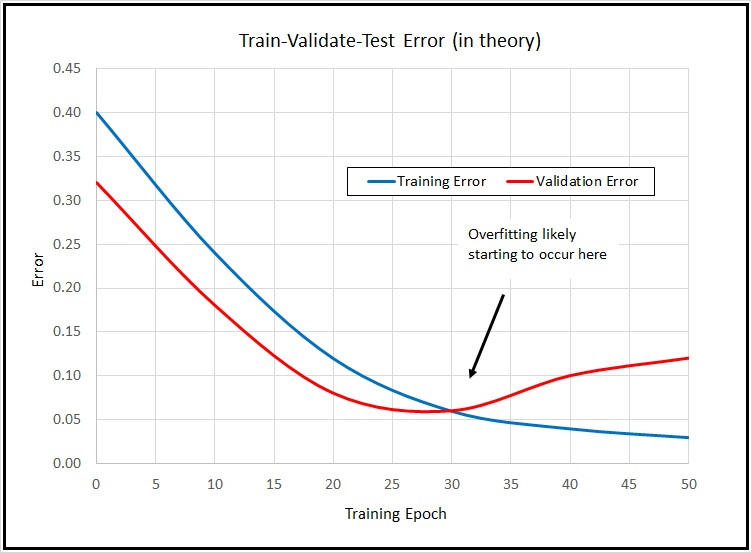
Part II Analysis

1. Discuss the results (e.g. how is accuracy impacted by training set size and how it compares to just predicting the dominant class) in a few sentences. For each training set size, is the decision tree overfitting?

* The accuracy on the test set decreases since the model overfits as the training dataset size increases, because we are not validating in this case.
* Since, we are dealing with binary classification and their is a case of imbalanced classes in the dataset, the baseline accuracy is high (~75 %) as compared to the ID3 accuracy.
* Yes, the model fits for each of the training size, this can be seen in the plot as the test accuracy decreases with increase in the train dataset size.
* The number of nodes in the graph increases with increase in the train dataset size.

1. Discuss the results (e.g. how is accuracy impacted by training set size and how

it compares to just predicting the dominant class) in a few sentences. For each

training set size, is the decision tree overfitting?

* The accuracy on the test set fluctuates around the baseline accuracy since we are validating alongside training, and trying to prevent overfitting.
* Since, we are now cross validating our model, when the training dataset size goes above a certain threshold (> 70 %), we receive accuracy better than baseline
* Through pruning we are preventing overfitting by decreasing the size of our tree.

Part III Theoretical Question

1. No, it is not necessary that ID3 will include all the attributes. In the given example, we can classify with just attribute A. So the decision tree will not have attribute B as it will just check if A=1, Class=K and A=0, Class=L

|  |  |  |
| --- | --- | --- |
| A | B | Class |
| 1 | 1 | K |
| 1 | 0 | K |
| 1 | 1 | K |
| 0 | 1 | L |
| 0 | 0 | L |

1. We cannot drop the cross validation set and use the test set for puning, because as this guarantees to be overestimating how good our model is. Using the test set, our model will optimize it according to it and show better results which are not real.
2. We can deal with missing values in the following ways:-
   1. Use the most common value of that attribute in data
   2. Use the most common value of that attribute in data, with the same class label (at train time)
   3. Assign fractional count instead of majority vote on value

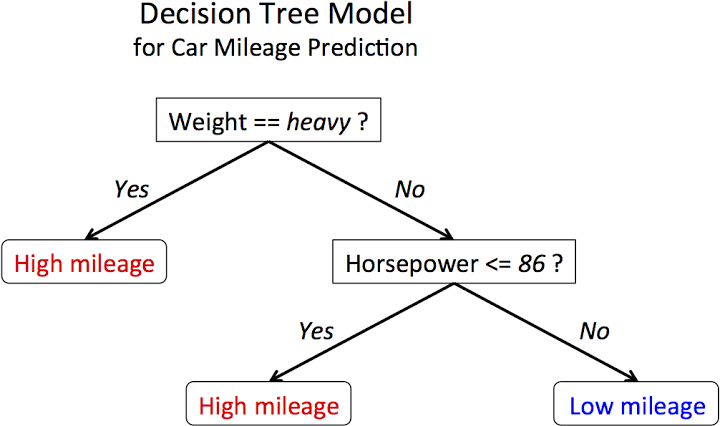
For the testing process, we can store the attributes values using one of the above techniques and fill in the missing values for the test set.

1. For generating a ranking model from decision tree, we can traverse the complete decision tree instead of the single path as done in ID3. In the preorder traversal, we can assign the edges as n or 0 (where n is a function of current depth, and decrease with increase in depth) based on if our sample contains that property/attribute, upto the leaf nodes.

We can then perform another tree traversal and add the weights from root to leaf along each path and assign the sum to the leaf nodes. Based on this, we can sum up the weights for leaf nodes which belong to the same class and normalize them to get the required probabilities of each class.

1. We can use threshold or ranges to get boolean tests for the range variables. We can split the continuous ranges based on the idea that the split change the proportion of labels in the children and each child should have a higher proportion of one of the labels.

Idea :- Go over split points with different labels, and find the one with highest information gain.

The split based on horsepower is an example of continuous range decision tree.