# Weekly Report

Name: Utkarsh Patel
Roll No: 18EC30048
Summary of lectures given in **Week-4** (Sept 23, Sept 24 and Sept 25)

**Topics Covered**

- **High Variance in Decision Tree Learning**
  - The ID3 algorithm that is used for building decision tree is very prone to overfit the training data.
  - Generally, a branch of decision tree is not terminated by the algorithm until it yields a pure leaf node.
  - This approach may not yield a low generalisation error if one of the following conditions is met:
    - Number of samples corresponding to a particular leaf node is very small.
    - There is noise in the training samples.
    - The size of training set is itself very small.

    In all this cases, the decision tree learned cannot represent the entire distribution very effectively.

- **Measures to reduce generalisation error**
  - Given a hypothesis space $H$, any hypothesis $h \in H$ is said to overfit the training set if $\exists h' \in H$, such that training error of $h'$ is larger as compared to $h$, but the generalisation error (test error) of $h'$ is smaller than that of $h$.
  - There exist two kinds of approaches to deal with high variance problem:
    - By limiting the growth of the decision tree up to a certain level.
    - By creating full fledged decision tree which may overfit the training set, and then pruning some of its nodes to increase the accuracy of the tree on test set.

    Method 1 seems more direct, but it has a problem that which depth is to be considered optimal. Hence, Method 2 is used much more often.
  - **Pruning:** While pruning an internal node of a decision tree, its subtree is removed, and the node is converted into a leaf node with the class label which is most frequent amongst the sample space of the particular node.
  - There are various ways one can apply pruning to the decision tree. Following are some most widely used strategies:
    - **Reduce-error pruning:** This method is used when we have enough data with us to train the decision tree. The dataset is divided into three parts: training set, validation set, and test set. The decision tree is learned while training on the training set and its accuracy is measured on the validation set. In this method, pruning is applied iteratively to the nodes of the decision tree until it is impossible to increase the accuracy further on the validation set.
    - **Rule post pruning:** This method is used when the dataset is very limited and we don't want to keep a part of data as validation set. In this method, decision tree is viewed as a set of rules. These rules, having conjunctive normal form, are basically derived from the various branches of the decision tree (root to leaf node). Each rule is pruned independently from each other. Basically, the algorithm tries to relax some precondition in turn of increasing the accuracy. Then, the rules are sorted as per their accuracy measure and are considered in this order when we need to classify any unseen instance. As no validation set is constructed in this approach, accuracy of rules is measured by using probabilistic methods on the training set itself.

- **Dealing with various types of attributes**
  - **Attributes having continuous values:** For continuous valued attribute $A$ taking values in space $X$ , new boolean attributes $A_{c1}, A_{c2}, \dots$ are considered for building the tree. The parameter $c_1, c_2, \dots$ are chosen so as to split $X$ into disjoint subsets as per the information gain principle.
  - **Attributes having many discrete values:** Problem with such attribute is that given that they partition any sample space $S$ into many subspaces, the entropy is largely decreased and hence information gain is very high. Hence, such attributes will always be #1 choice of naïve gain functions. However, the high gain has nothing to do with how well the attribute helps in categorizing the sample instances. To deal with such attributes, a partitioning penalty is introduced in the information gain function.
  - **Attributes having no values:** When an instance is being checked at a node for a particular attribute and it doesn't have any value corresponding to that attribute, the value of the attribute is approximated to:
    - The most frequent value of attribute of sample instances which have the same class label
    - The most frequent value of attribute of sample instances which are being checked on that node
    - Expected value from the probabilistic analysis of that attribute.
  - **Attributes having costs:** Cost penalty is introduced in the gain functions.
  - There are several other types of gain function that we can used to define information gain, one such function is *Gini Index* which is defined as sum of product of probabilities of class labels taken pair wise.

- **Regression Trees**
  - This involves computing squared errors instead of information gain, and partitioning the input space in rectangular subspaces, each with its own predictor.
  - Impurity of a sample defined by the variance of the output in that learning sample, and the best split is one that reduces the variance the most.
  - Pruning is more important in regression trees because full trees are much more complex.