ASSIGNMENT NO: 4

TITLE- Introduction to Random Forest.

Problem Definition- Using dataset of decision tree assignment build a random forest model with buys as a target variable.

Objectives:
1) Understanding of statistical analysis behind random forest.
2) Visualizing performance and accuracy of random forest and decision tree.

Outcome- After completion of this assignment students will be able to understand the implemntation and working of random forest algorithm.

Theory-

What Is Random Forest?

Random forest algorithm is a supervised classification and regression algorithm. As the name suggests, this algorithm randomly creates a forest with several trees,Generally, the more trees in the forest the more robust the forest looks like. Similarly, in the random forest classifier, the higher the number of trees in the forest, greater is the accuracy of the results.

Why Use Random Forest?

You might be wondering why we use Random Forest when we can solve the same problems using Decision trees.

- Even though Decision trees are convenient and easily implemented, they lack accuracy. Decision trees work very effectively with the training data that was used to build them, but they're not flexible when it comes to classifying the new sample. Which means that the accuracy during testing phase is very low.
- This happens due to a process called Over-fitting.

->Over-fitting occurs when a model studies the training data to such an extent that it negatively influences the performance of the model on new data.

The Random Forest Classifier

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

Creating A Random Forest:

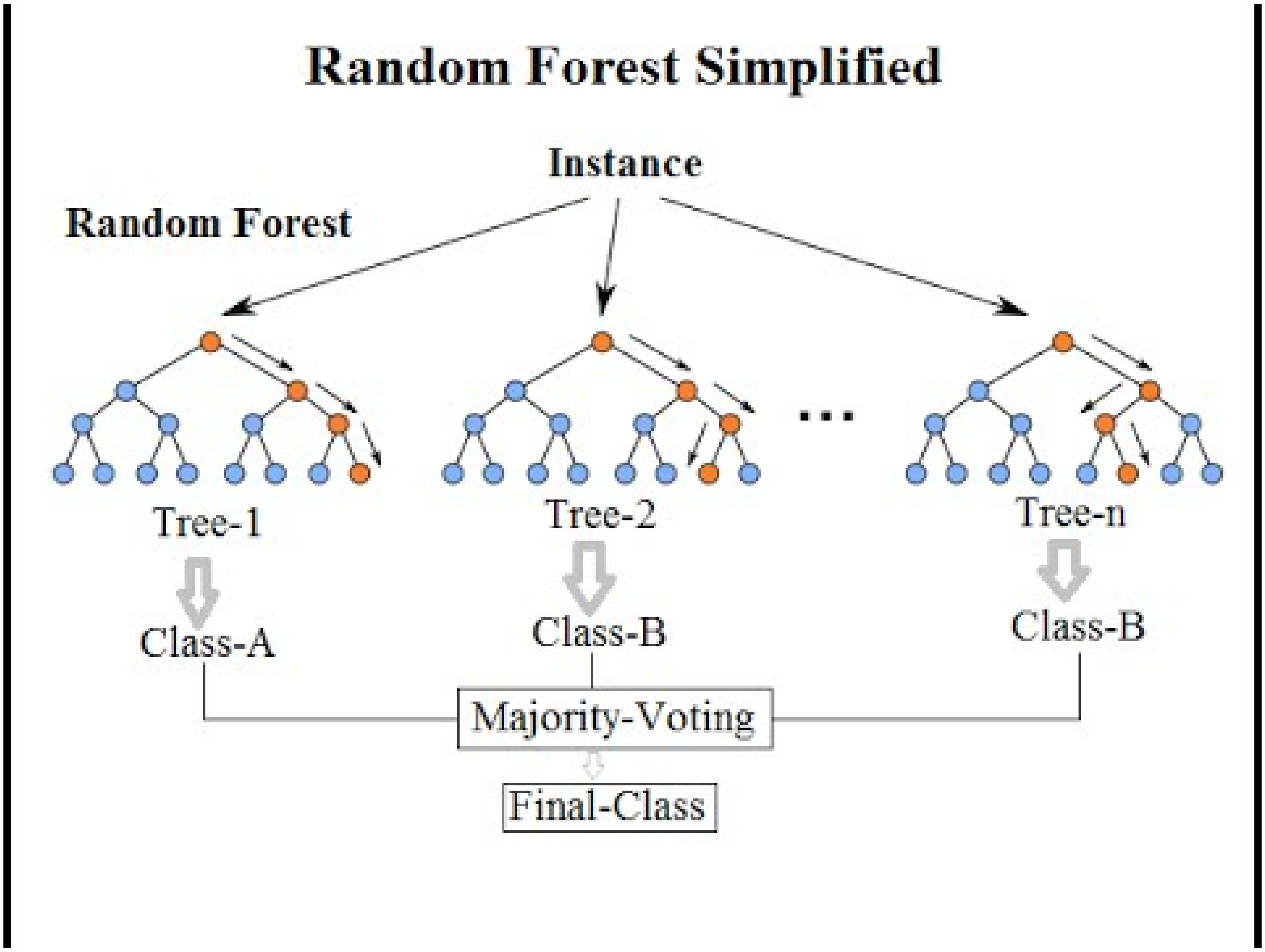
Step 1: Create a Bootstrapped Data Set

Step 2: Creating Decision Trees

Step 3: Go back to Step 1 and Repeat

Step 4: Predicting the outcome of a new data point.

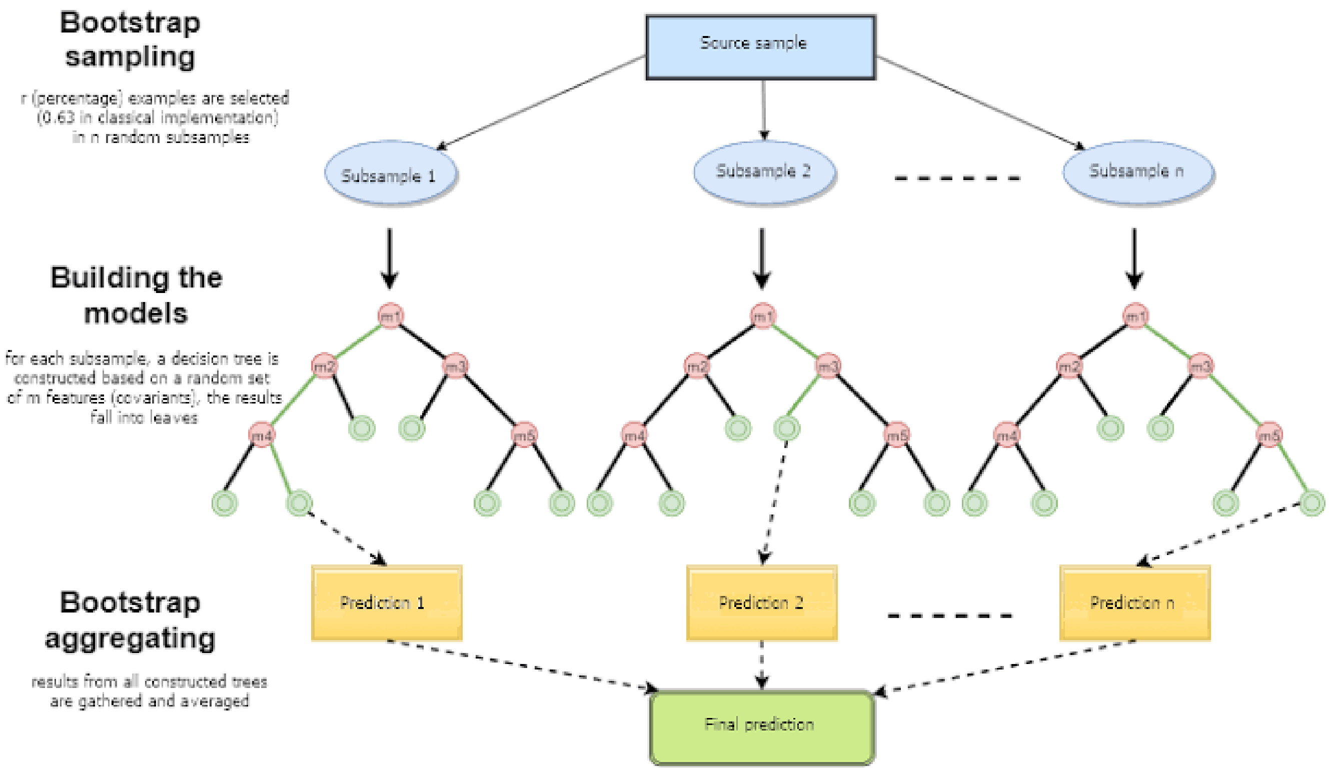
Step 5: Evaluate the Model



BOOTSTRAPPING:

Random Forests are trained via the bagging method. Bagging or Bootstrap Aggregating, consists of randomly sampling subsets of the training data, fitting a model to these smaller data sets, and aggregating the predictions. This method allows several instances to be used repeatedly for the training stage given that we are sampling with replacement. Tree bagging consists of sampling subsets of the training set, fitting a Decision Tree to each, and aggregating their result.

The Random Forest method introduces more randomness and diversity by applying the bagging method to the feature space. That is, instead of searching greedily for the best predictors to create branches, it randomly samples elements of the predictor space, thus adding more diversity and reducing the variance of the trees at the cost of equal or higher bias. This process is also known as “feature bagging” and it is this powerful method what leads to a more robust model.


Bootstrap sampling
r (percentage) examples are selected (0.63 in classical implementation) in n random subsamples
Source sample
Subsample 1
Subsample 2
Subsample n
Building the models
for each subsample, a decision tree is constructed based on a random set of m features (covariants), the results fall into leaves
m1
m2
m3
m4
m5
m1
m2
m3
m4
m5
m1
m2
m3
m4
m5
Bootstrap aggregating
results from all constructed trees are gathered and averaged
Prediction 1
Prediction 2
Prediction n
Final prediction

Problems with Decision Tree:

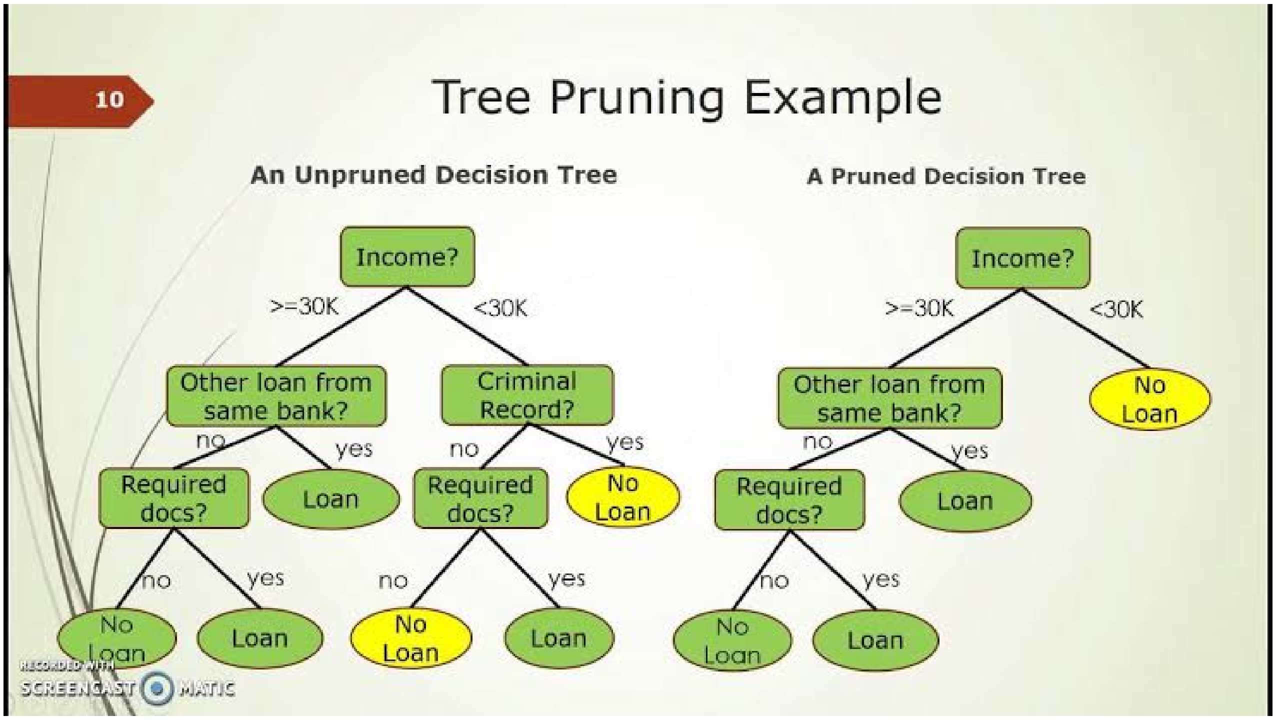
Overfitting is a significant practical difficulty for decision tree models and many other predictive models. Overfitting happens when the learning algorithm continues to develop hypotheses that reduce training set error at the cost of an increased test set error. There are several approaches to avoiding overfitting in building decision trees.

- **Pre-pruning:** that stop growing the tree earlier, before it perfectly classifies the training set.
- **Post-pruning:** that allows the tree to perfectly classify the training set, and then post prune the tree.

PRUNING DECISION TREE AND RANDOM FOREST:

A decision tree that is very deep or of full depth tend to learn the noise in the data. They overfit to the data leading to low bias but high variance. Pruning is a suitable approach used in decision trees to reduce overfitting.

However, generally random forests would give good performance with full depth. As random forests training use bootstrap aggregation (or sampling with replacement) along with random selection of features for a split, the correlation between the trees (or weak learners) would be low. That means although individual trees would have high variance but the ensemble output will be appropriate (lower variance and lower bias) because the trees are not correlated.
If you still want to control the training in random forest, go for controlling the tree depth instead of pruning.



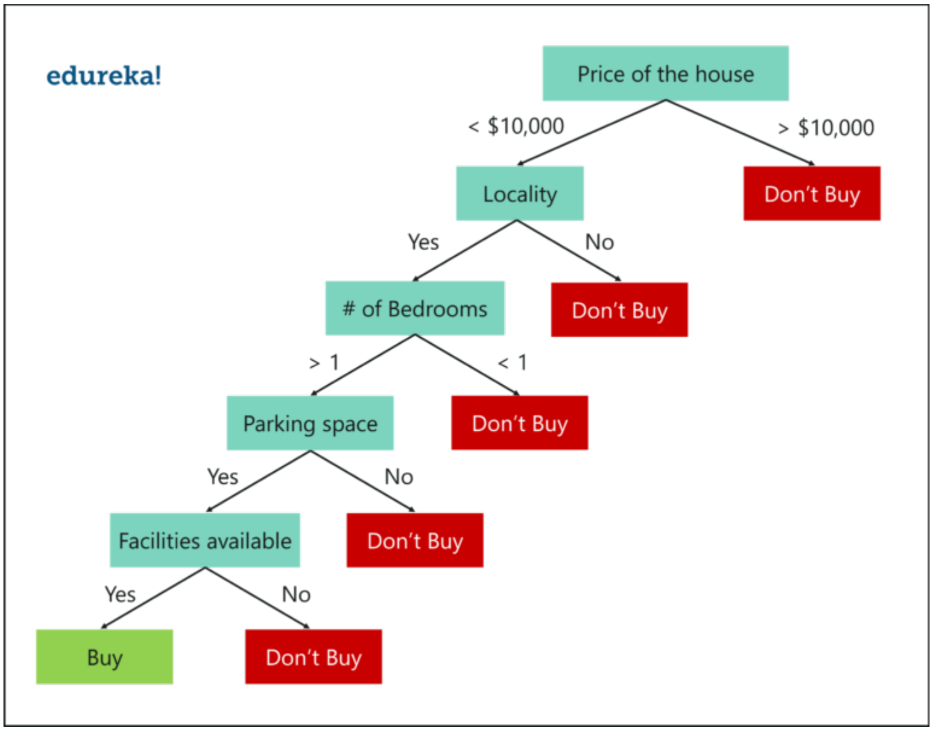
10

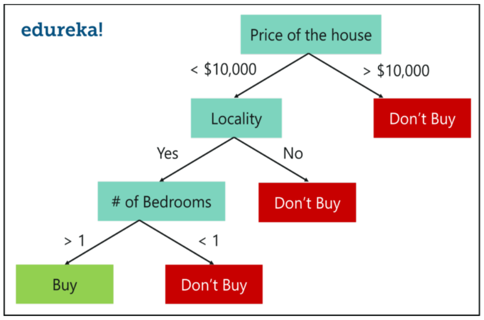
Tree Pruning Example

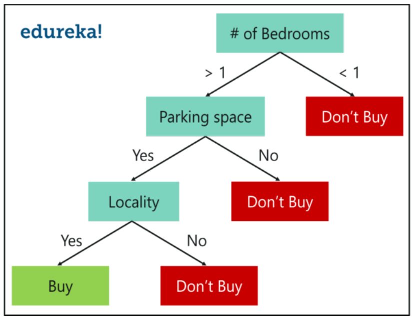
An Unpruned Decision Tree

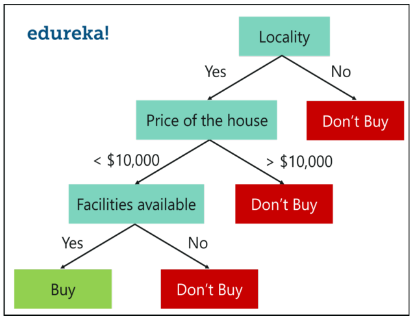
A Pruned Decision Tree

FORMATION OF DECISION TREE FROM FOREST.









Conclusion:Through this assignment we have understood the basic terminologies of random forest algorithm and comparative analysis with decision tree.