Random Forest Algorithm

Function RANDOM_FOREST (Data D, Integer T,
                        Integer m):

    Trees ← [ ]

    for t from 1 to T do:

        Bootstrap Sample ← Sample-with-Replacement

        Tree ← Build-Decision-tree (Bootstrap Sam

        Append Tree to Trees

    Return Trees

Function Build-Decision tree (Data D, Integer m):

    At Each node in tree:

    Randomly Select m features from all feature

    Find the best split using only these m features

    Recursively Split until Stopping Condition is met

    Return decision tree

Function PREDICT_RANDOM_FOREST (Trees, Instance x

    votes ← {]

    for Each Tree In Trees do:

        Prediction ← Tree (x)

        Append Prediction to Votes

    Return MAJORITY_VOTE (votes)

2) k-Means Clustering

Function k-Means (Data D, Integer k, Integer max-iters):

Centroids : $M_1, M_2 \ldots M_k$

Initialize k random

for Iter from 1 to max-iters do:

    Clusters ← empty lists for each cluster

    for each data point x in D do:

        Find the closest centroid M_j

        Assign x to cluster j

    for each cluster j do:

        M_j ← MEAN of all points assigned to cluster j

    If centroids do not change:

        Break.

Return clusters, Centroids.