Problem 1.

Algorithm:

Information Gain:

$IG(X|Y) = H(X) - H(X|Y)$

$= \sum_{i=1}^{n} -p(x_i)log_2 p(x_i) - \sum_{j=1}^{m} p(y_j)H(X|y_j)$

The approaches used in this problem comes from the Lecture 19.

Input: Training set $(x_1, y_1), ..., (x_n, y_n) \in \chi \times \{-1, 1\}$

Algorithm: Initialize $w_1(i) = 1/n$

For $t = 1, ..., T$

(1) Select a weak learner using information gain.

(2) Get weak hypothesis G_t with error $\epsilon_t = \sum_i w_t(i) 1[G_t(x_i) \neq y_i]$

(3) Choose $\alpha_i = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$

(4) Update $w_{t+1}(i) = \frac{w_t(i) exp(-\alpha_t y_i G_t(x_i))}{Z_t}$, where Z_t is the normalization factor.

Output:

$g(x) = sign[\sum_{t=1}^{T} \alpha_t G_t(x)]$

Result:

For the dataset, I entirely removed the features with missing data and the first attribute(id).

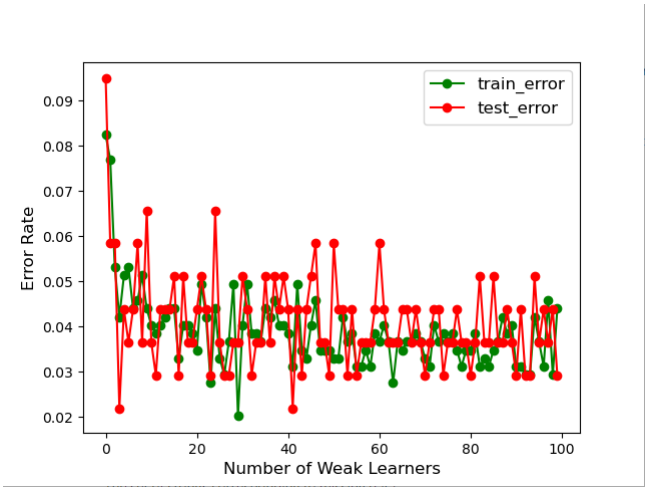

Figure 1: Adaboost Plot

Problem 2

Algorithm:

The approaches used in this problem comes from the Lecture 20.

Boostrap Sampling:

(1)Consider training set D with m training examples.

(2)Create D^i by drawing m examples with replacement.

(3)In expectation, D^i will leave out a fraction of examples.

(4)Each D^i will approximate the distribution underlying D.

For **Random Forest**, we would build a forest of decision trees:

(1)Create k bootstrap samples $D^1, ..., D^k$.

(2)Learn an un-pruned decision tree on each sample.

(3)Learning: At each internal node:

Randomly select $m < d$ features.

Determine the best split using only these features.

(4)Prediction: Use output from all trees in the forest.

Classification: Majority vote.

Result:

For the dataset, I entirely removed the features with missing data and the first attribute(id).

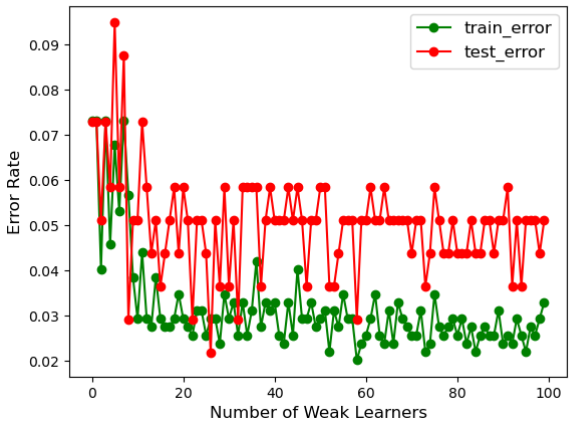

Figure 2: Random Forest with 3 attributes and vary decision stumps

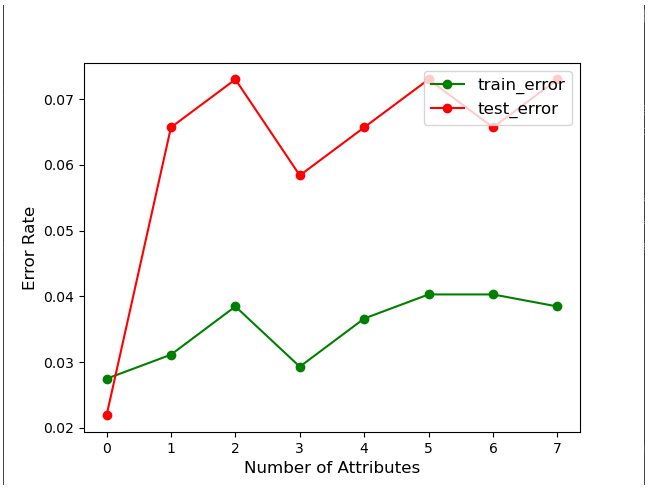

Figure 3: Random Forest with vary attributes and 100 decision stumps

Problem 3

Algorithm:

The approaches used in this problem comes from the Lecture 22.

K-means:

(1)Consider a dataset $\chi = \{x_1, ..., x_N\}, x_i \in R^d$

(2)Assume there are K clusters $C_1, ..., C_K$

(3)Associate a prototype $\mu_h, h = 1, ..., K$ with each cluster.

(4)Let $r_{nk} \in \{0, 1\}$ be the indicator of $x_n \in C_k$.

(5)The goal is to minimize the following distortion measure.

$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2$

Then, we optimize over $\{r_{nk}\}$ for a fixed $\{\mu_k\}$ and optimize over $\{\mu_k\}$ for a fixed $\{r_{nk}\}$.

Result:

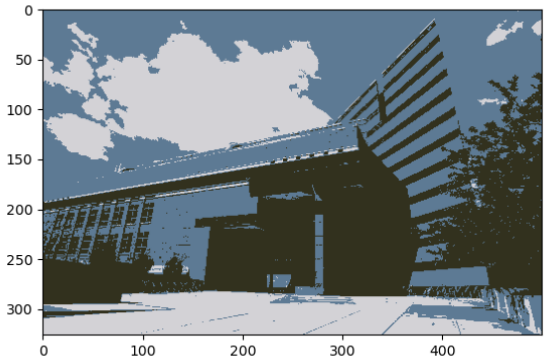

Figure 4: K means with 3 centroids

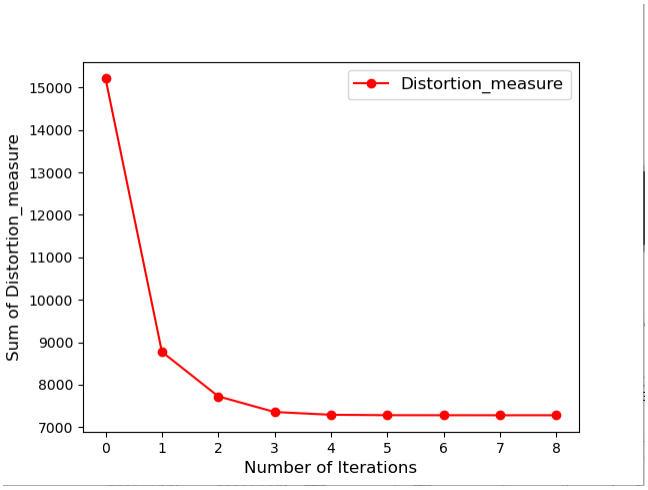

Figure 5: K means with 3 centroids

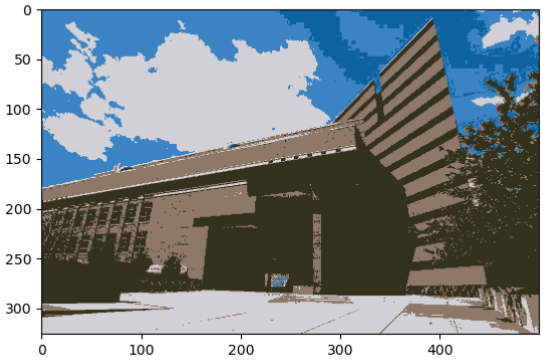

Figure 6: K means with 5 centroids

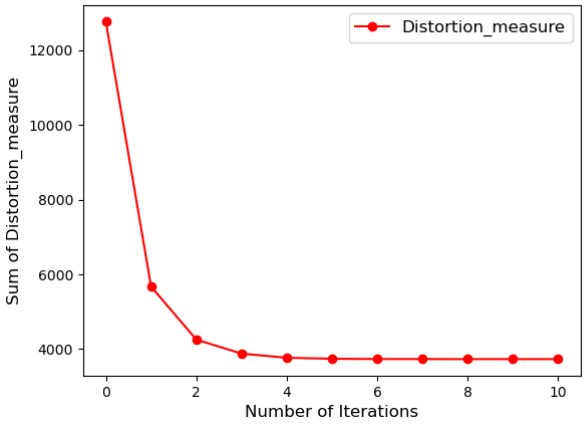

Figure 7: K means with 5 centroids

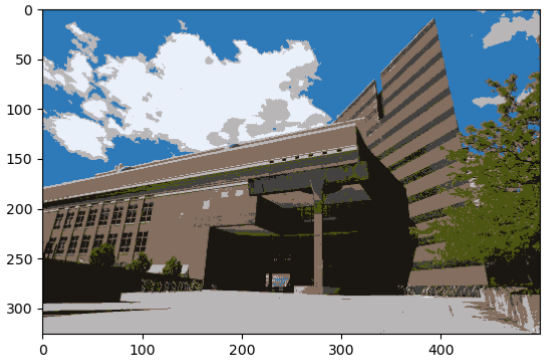

Figure 8: K means with 7 centroids

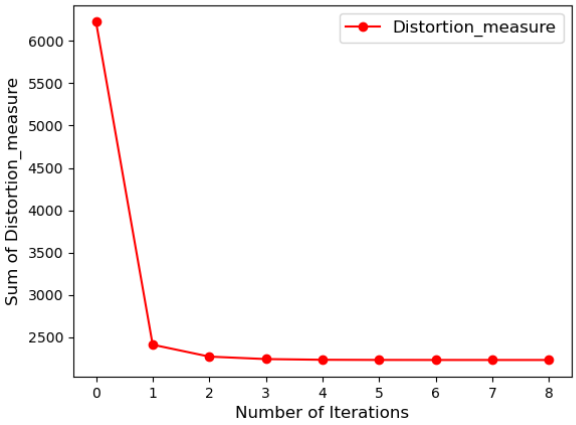

Figure 9: K means with 7 centroids

Reference:

1.Pattern Recognition and Machine Learning,CHRISTOPHER M.BISHOP

2.Lecture in Canvas,Nicholas Johnson