Assignment_2

1. What is the key idea behind bagging? Can bagging deal both with high variance (overfitting) and high bias (underfitting)?

A. Bagging is an ensemble machine learning approach, often known as bootstrap aggregation. To create many samples of the same training dataset, the fundamental principle of bagging is to choose different samples of the training set and substitute them. After the various training set samples are trained, the various samples provide distinct outcomes. One forecast is created by combining all of the data from the various. K times through this procedure can be done until no further progress is made. Yes, the primary goal of using the bagging approach is to lessen overfitting or excessive variance. As a result of the model knowing too much about the data, each classifier is overfitted. Using replacements to produce distinct samples, bagging aids in reducing variance or overfitting in the final ensemble model. Bagging performs differently when there is strong bias or underfitting because when individual classifiers are underfitting, the total ensemble model also underfits.

2. Why bagging models are computationally more efficient when compared to boosting models with the same number of weak learners?

A. In bagging, the various training set samples are trained separately before being combined into the final mode to help the model provide better results. As a result, parallelization is used in the bagging model training. When using boosting, models are sequentially trained one after the other using the output of the prior model. Based on the accuracy of the previous model, the model generates fresh iterations of the training set to minimize the error of the previous model. As a result, accuracy is increased, and mistakes are reduced by raising the computational complexity for each repetition.

3. James is thinking of creating an ensemble mode to predict whether a given stock will go up or down in the next week. He has trained several decision tree models, but each model is not performing any better than a random model. The models are also very similar to each other. Do you think creating an ensemble model by combining these tree models can boost the performance? Discuss your answer.

A. The goal of the ensemble model is to increase overall performance by combining many independent models. The ensemble model combines varying strengths of each to increase accuracy. Several decision tree models are inadequate in the James situation. There wouldn't be a noticeable performance gain when James ‘models were used to ensemble models since the model's output would similarly be poor. Another reason the James ensemble is unable to improve performance is that the decision tree's primary flaw is that it is unsuitable for continuous numerical variables. James is trying to forecast the stock values, which are a continuous numerical variable.

4. Consider the following Table that classifies some objects into two classes of edible (+) and non- edible (-), based on some characteristics such as the object color, size and shape. What would be the Information gain for splitting the dataset based on the "Size" attribute?

A. Information gain = entropy(parent) – [average entropy(children)]

Calculating Size(parent entropy):

– 9/16.log2 (9/16) – 7/16.log2(7/16)

– 9/16(-0.83) – 7/16(–1.1926)

Parent(Size) entropy = 0.98

Calculating "Small" and "Large" (Children entropy):

Calculating Child ("Small" entropy):

– 6/8.log2(6/8) – 2/8.log2(2/8)

– 6/8(-0.415) –2/8.log2(-2)

Child ("Small" entropy) = 0.81

Calculating Child ("Large" entropy):

– 3/8.log2(3/8) – 5/8.log2(5/8)

–3/8(-1.415) – 5/8(-0.6781)

Child ("Large" entropy) = 0.95

Weighted Average Entropy of Children =

8/16(0.81) + 8/16(0.95) = 0.88

Information gain = 0.98 – 0.88 = 0.10.

5. Why is it important that the m parameter (number of attributes available at each split) to be optimally set in random forest models? Discuss the implications of setting this parameter too small or too large.

A. While selecting the attribute at each split in the decision tree model, it is crucial to select the best value for m as this adds diversity. Additionally, since each decision tree contains model alternatives, it is not limited. The qualities chosen at each split are overly constrained when the m is chosen too tiny. The decision tree model's performance would thus be below par. The model lacks variety because of the excessively large m selection; all decision tree models are the same, and variability is solely produced by bootstrapping.