22nd April Quiz

7 out of 7 correct

1. Which of the following is NOT a variant of KNN algorithm?

- [x] Naive Bayes
- [ ] Brute Force KNN
- [ ] K-Dimension Tree
- [ ] Ball Tree

Explanation: Naive Bayes is a classification algorithm that uses Bayes' theorem to predict the probability of a certain class based on the features of the input data. It is not a variant of KNN algorithm.

2. Which variant of KNN algorithm uses a brute force approach to compute the distances between the new data point and all the training data points?

- [x] Brute Force KNN
- [ ] K-Dimension Tree
- [ ] Ball Tree
- [ ] None of the above

Explanation: In the Brute Force KNN algorithm, the distances between the new data point and all the training data points are computed directly, without any optimization techniques such as data structures.

3. Which variant of KNN algorithm partitions the data points into a binary tree based on their features?

- [ ] Brute Force KNN

- [x] K-Dimension Tree
- [ ] Ball Tree
- [ ] None of the above

Explanation: In the K-Dimension Tree algorithm, the data points are partitioned into a binary tree based on their features. This allows for faster computation of the distances between the new data point and the training data points.

4. Which of the following is an advantage of using K-Dimension Tree over Brute Force KNN?

- [x] K-Dimension Tree can handle high-dimensional data more efficiently.
- [ ] Brute Force KNN is always faster than K-Dimension Tree
- [ ] K-Dimension Tree requires less memory than Brute Force KNN.
- [ ] Brute Force KNN is more accurate than K-Dimension Tree.

Explanation: Brute Force KNN has a high computational cost for high-dimensional data because it requires the computation of the distance between the new data point and all the training data points. K-Dimension Tree partitions the data points into a binary tree, allowing for faster computation of the distances and making it more efficient for high-dimensional data.

5. Which variant of KNN algorithm is more appropriate for data with unevenly distributed data points?

- [ ] Brute Force KNN
- [ ] K-Dimension Tree
- [x] Ball Tree
- [ ] All of the above

Explanation: Ball Tree is more appropriate for data with unevenly distributed data points because it partitions the data into hyperspheres, which can better capture the geometry of the data.

6. In Brute Force KNN, if the number of training data points is 1000 and the number of features is 10, how many distances will be computed to classify a single new data point?

- ◯ 1000
- ◉ 10,000
- ◯ 1,000,000
- ◯ 10,000,000

Explanation: In Brute Force KNN, the distance between the new data point and all the training data points needs to be computed to classify a single new data point. Therefore, the number of distances to be computed is equal to the number of training data points multiplied by the number of features, which is 1000 x 10 = 10,000.

7. In K-Dimension Tree, if the number of training data points is 1000 and the number of features is 10, how many splits will be performed to construct the binary tree?

- ◯ 10
- ◉ 20
- ◯ 30
- ◯ 1000

Explanation: In K-Dimension Tree, the binary tree is constructed by recursively splitting the data points along each feature dimension. The number of splits required to construct the binary tree is equal to the logarithm of the number of data points to the base 2, multiplied by the number of features. Therefore, the number of splits in this case is log2(1000) x 10 = 20.

Submit