Interview questions

04 March 2021 08:13 PM

1. ***What Are the Basic Assumption?***

   No assumptions because it is a non-parametric algorithm

2. ***2. Advantages***
   - No training stage
   - New data can be added to the training dataset and it will not impact the accuracy of the algorithm.
   - It is easy to implement. There are only 2 parameters required to implement KNN (value of k and the distance function).

3. ***Disadvantages***
   - It takes more time to predict if the dataset is huge and has high dimensions.
   - Need feature scaling: KNN needs to do feature scaling otherwise it will negatively affect the predictions.
   - Sensitive to noisy data, missing values and outliers.
   - Test stage is slow
   - Algorithm is sensitive to outliers, since a single mislabeled example dramatically changes the class boundaries. Anomalies affect the method significantly, because k-NN gets all the information from the input, rather than from an algorithm that tries to generalize data.

4. ***Whether Feature Scaling is required?***

   Yes

5. ***Impact of Missing Values?***
   - We need a complete features vector for each instance in order to compute distance. So, any missing value has to be filled.

     Proposal: Setting the average value of the feature across the entire dataset for missing values, or restrict the distance calculation to a subspace may be reasonable choices.

   From <https://quantdare.com/10-reasons-for-loving-nearest-neighbors-algorithm/>

6. ***Impact of outliers?***
   - Algorithm is sensitive to outliers, since a single mislabeled example dramatically changes the class boundaries. Anomalies affect the method significantly, because k-NN gets all the information from the input, rather than from an algorithm that tries to generalize data.
   - *Proposal:* Avoid very small number of neighbors (k=1, for example), especially if you are in front of noisy data, so always.

7. **Affect of imbalance data?**
   - k-NN doesn't perform well on imbalanced data. If we consider two classes, A and B, and the majority of the training data is labeled as A, then the model will ultimately give a lot of preference to A. This might result in getting the less common class B wrongly classified.
   - *Proposal:* This can be mitigated by bagging class A & B samples separately: We could classify a new sample using several k-nearest neighbors' sets, where each is obtained by random sampling the available data; we should force the random sampling to select examples with equilibrated classes frequencies. Another solution could be to

implement a **correction of frequencies** by weighting the neighbors' decisions: more to those labeled with the less frequent class. **SMOTE** (Synthetic Minority Over-Sampling Technique) and **MSMOTE** are also methods specifically designed for learning from imbalanced data sets.

? ***8. Types of Problems it can solve(Supervised)***
? ***9. Overfitting And Underfitting***
? ***10. Different Problem statement you can solve using Linear Regression***