K NEAREST NEIGHBHOUR Interview Questions

We believe that you have learned both theoritical and practical knowledge on Naive Bayes classification algorithm through your assignment.

So let's test your knowledge here. This will help you to be prepared for interviews too!

Best with Quest ¶

1. What is the KNN Algorithm?

```
KNN(K-nearest neighbours) is a supervised learning and non-parametric algorithm that can
be used to solve both classification and regression problem statements.

It uses data in which there is a target column present i.e, labelled data to model a
function to produce an output for the unseen data. It uses the euclidean distance formula
to compute the distance between the data points for classification or prediction.

The main objective of this algorithm is that similar data points must be close to each
other so it uses the distance to calculate the similar points that are close to each
other.
```

2. Why do you need to scale your data for the k-NN algorithm?

```
Imagine a dataset having m number of “examples” and n number of “features”. There is one
feature dimension having values exactly between 0 and 1. Meanwhile, there is also a
feature dimension that varies from -99999 to 99999. Considering the formula of Euclidean
Distance, this will affect the performance by giving higher weightage to variables having
a higher magnitude.
```

3. Is Euclidean Distance always the case in KNN?

```
Although Euclidean Distance is the most common method used and taught, it is not always
the optimal decision. In fact, it is hard to come up with the right metric just by looking
at data, so I would suggest trying a set of them. However, there are some special cases.
For instance, hamming distance is used in case of a categorical variable.
```

4. Why should we not use the KNN algorithm for large datasets?

```
Here is an overview of the data flow that occurs in the KNN algorithm:
1.Calculate the distances to all vectors in a training set and store them
2.Sort the calculated distances
3.Store the K nearest vectors
4.Calculate the most frequent class displayed by K nearest vectors
```

Imagine you have a very large dataset. Therefore, it is not only a bad decision to store a large amount of data but it is also computationally costly to keep calculating and sorting all the values.

5. What is "K" in the KNN Algorithm? Why is the odd value of "K" preferred over even values in the KNN Algorithm?

K represents the number of nearest neighbours you want to select to predict the class of a given item, which is coming as an unseen dataset for the model.

The odd value of K should be preferred over even values in order to ensure that there are no ties in the voting. If the square root of a number of data points is even, then add or subtract 1 to it to make it odd.

6. How does the KNN algorithm make the predictions on the unseen dataset?

The following operations have happened during each iteration of the algorithm. For each of the unseen or test data point, the kNN classifier must:

Step-1: Calculate the distances of test point to all points in the training set and store them

Step-2: Sort the calculated distances in increasing order

Step-3: Store the K nearest points from our training dataset

Step-4: Calculate the proportions of each class

Step-5: Assign the class with the highest proportion

7. What is space and time complexity of the KNN Algorithm?

Time complexity:

The distance calculation step requires quadratic time complexity, and the sorting of the calculated distances requires an O(N log N) time. Together, we can say that the process is an O(N3 log N) process, which is a monstrously long process.

Space complexity:

Since it stores all the pairwise distances and is sorted in memory on a machine, memory is also the problem. Usually, local machines will crash, if we have very large datasets.

8. Can the KNN algorithm be used for regression problem statements?

Yes, KNN can be used for regression problem statements.

In other words, the KNN algorithm can be applied when the dependent variable is continuous. For regression problem statements, the predicted value is given by the average of the values of its k nearest neighbours.

9. Why is the KNN Algorithm known as Lazy Learner?

When the KNN algorithm gets the training data, it does not learn and make a model, it just stores the data. Instead of finding any discriminative function with the help of the training data, it follows instance-based learning and also uses the training data when it actually needs to do some prediction on the unseen datasets.

As a result, KNN does not immediately learn a model rather delays the learning thereby being referred to as Lazy Learner.

10. How to choose the optimal value of K in the KNN Algorithm?

There is no straightforward method to find the optimal value of K in the KNN algorithm.

You have to play around with different values to choose which value of K should be optimal for my problem statement. Choosing the right value of K is done through a process known as Hyperparameter Tuning.

The optimum value of K for KNN is highly dependent on the data itself. In different scenarios, the optimum K may vary. It is more or less a hit and trial method.

There is no one proper method of finding the K value in the KNN algorithm. No method is the rule of thumb but you should try the following suggestions:

1. Square Root Method: Take the square root of the number of samples in the training dataset and assign it to the K value.

2. Cross-Validation Method: We should also take the help of cross-validation to find out the optimal value of K in KNN. Start with the minimum value of k i.e, K=1, and run cross-validation, measure the accuracy, and keep repeating till the results become consistent.

As the value of K increases, the error usually goes down after each one-step increase in K, then stabilizes, and then raises again. Finally, pick the optimum K at the beginning of the stable zone. This technique is also known as the Elbow Method.

How can we find the optimum K in K-Nearest Neighbor?

Image Source: Google Images

3. Domain Knowledge: Sometimes with the help of domain knowledge for a particular use case we are able to find the optimum value of K (K should be an odd number).

I would therefore suggest trying a mix of all the above points to reach any conclusion.

11. How can you relate KNN Algorithm to the Bias-Variance tradeoff?

Problem with having too small K:

The major concern associated with small values of K lies behind the fact that the smaller value causes noise to have a higher influence on the result which will also lead to a large variance in the predictions.

Problem with having too large K:

The larger the value of K, the higher is the accuracy. If K is too large, then our model is under-fitted. As a result, the error will go up again. So, to prevent your model from under-fitting it should retain the generalization capabilities otherwise there are fair chances that your model may perform well in the training data but drastically fail in the real data. The computational expense of the algorithm also increases if we choose the k very large.

So, choosing k to a large value may lead to a model with a large bias(error).

The effects of k values on the bias and variance is explained below :

As the value of k increases, the bias will be increases
As the value of k decreases, the variance will increases
With the increasing value of K, the boundary becomes smoother
So, there is a tradeoff between overfitting and underfitting and you have to maintain a balance while choosing the value of K in KNN. Therefore, K should not be too small or too large.

12. Explain the statement- "The KNN algorithm does more computation on test time rather than train time".

The above-given statement is absolutely true.

The basic idea behind the kNN algorithm is to determine a k-long list of samples that are close to a sample that we want to classify. Therefore, the training phase is basically storing a training set, whereas during the prediction stage the algorithm looks for k-neighbours using that stored data. Moreover, KNN does not learn anything from the training dataset as well.

13. What are the advantages of the KNN Algorithm?

Some of the advantages of the KNN algorithm are as follows:

1. No Training Period: It does not learn anything during the training period since it does not find any discriminative function with the help of the training data. In simple words, actually, there is no training period for the KNN algorithm. It stores the training dataset and learns from it only when we use the algorithm for making the real-time predictions on the test dataset.

As a result, the KNN algorithm is much faster than other algorithms which require training. For Example, SupportVector Machines(SVMs), Linear Regression, etc.

Moreover, since the KNN algorithm does not require any training before making predictions as a result new data can be added seamlessly without impacting the accuracy of the algorithm.

2. Easy to implement and understand: To implement the KNN algorithm, we need only two parameters i.e. the value of K and the distance metric(e.g. Euclidean or Manhattan, etc.). Since both the parameters are easily interpretable therefore they are easy to understand.

14. What are the disadvantages of the KNN Algorithm?

Some of the disadvantages of the KNN algorithm are as follows:

1. Does not work well with large datasets: In large datasets, the cost of calculating the distance between the new point and each existing point is huge which decreases the performance of the algorithm.

2. Does not work well with high dimensions: KNN algorithms generally do not work well with high dimensional data since, with the increasing number of dimensions, it becomes difficult to calculate the distance for each dimension.

3. Need feature scaling: We need to do feature scaling (standardization and normalization) on the dataset before feeding it to the KNN algorithm otherwise it may generate wrong predictions.

4. Sensitive to Noise and Outliers: KNN is highly sensitive to the noise present in the dataset and requires manual imputation of the missing values along with outliers removal.

15. What are the real-life applications of KNN Algorithms?

The various real-life applications of the KNN Algorithm includes:

1. KNN allows the calculation of the credit rating. By collecting the financial characteristics vs. comparing people having similar financial features to a database we can calculate the same. Moreover, the very nature of a credit rating where people who have similar financial details would be given similar credit ratings also plays an important role. Hence the existing database can then be used to predict a new customer's credit rating, without having to perform all the calculations.

2. In political science: KNN can also be used to predict whether a potential voter "will vote" or "will not vote", or to "vote Democrat" or "vote Republican" in an election.

Apart from the above-mentioned use cases, KNN algorithms are also used for handwriting detection (like OCR), Image recognition, and video recognition.

Now Rest with this Quest :)