30 Interview Questions to Test your Skills on KNN Algorithm

BEGINNER MACHINE LEARNING SUPERVISED

Introduction

The K Nearest Neighbor (KNN) algorithm is a cornerstone in the realm of supervised Machine Learning, renowned for its simplicity and effectiveness in tackling classification challenges. This algorithm's ease of understanding and implementation, coupled with its robust performance, makes it indispensable for anyone venturing into the field of Data Science and Machine Learning.

This comprehensive tutorial aims to equip aspiring Data Scientists and Machine Learning Engineers with a thorough understanding of the KNN algorithm. Through a curated selection of interview questions and answers, it navigates from the foundational principles to more complex aspects of KNN. The guide is meticulously designed for beginners and seasoned practitioners alike, ensuring a solid grasp of KNN's applications and intricacies. It provides the knowledge and confidence necessary to excel in learning journeys and professional endeavors, focusing on the essentials of the KNN algorithm.

Learning Objectives

- Prepare for Data Science interviews with a focus on KNN, enhancing knowledge from basic concepts to advanced applications.
- Understand the foundational principles and applications of the K Nearest Neighbor algorithm in supervised learning.

This article was published as a part of the Data Science Blogathon.

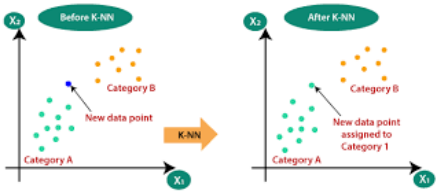
30 Interview Questions to Test your Skills on KNN Algorithm

1. What is the KNN Algorithm?

K-nearest neighbors algorithm (KNN)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.



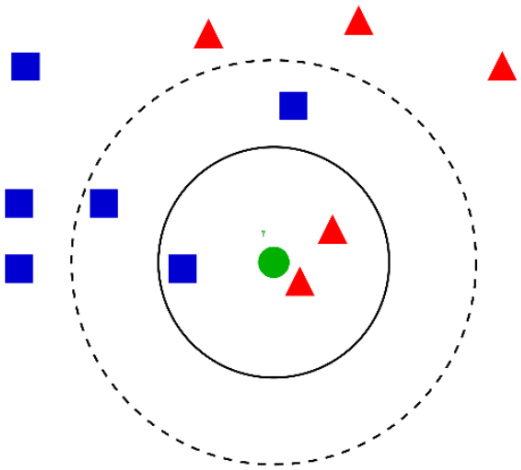
2. Why is KNN a non-parametric Algorithm?

The term "**non-parametric**" refers to not making any assumptions on the underlying data distribution. These methods do not have any fixed numbers of parameters in the model.

Similarly in KNN, the model parameters grow with the training data by considering each training case as a parameter of the model. So, KNN is a non-parametric algorithm.

3. What is "K" in the K-nearest neighbors 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.



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

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.

5. 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

6. Is Feature Scaling required for the KNN Algorithm? Explain with proper justification.

Yes, feature scaling is required to get the better performance of the KNN algorithm.

For Example, Imagine a dataset having n number of instances and N number of features. There is one feature having values ranging between **0 and 1**. Meanwhile, there is also a feature that varies from **-999 to 999**. When these values are substituted in the formula of Euclidean Distance, this will affect the performance by giving higher weightage to variables having a higher magnitude.

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. Why is it recommended not to use the KNN Algorithm for large datasets?

The Problem in processing the data:

KNN works well with smaller datasets because it is a lazy learner. It needs to store all the data and then make a decision only at run time. It includes the computation of distances for a given point with all other points. So if the dataset is large, there will be a lot of processing which may adversely impact the performance of the algorithm.

Sensitive to noise:

Another thing in the context of large datasets is that there is more likely a chance of noise in the dataset which adversely affects the performance of the KNN algorithm since the KNN algorithm is sensitive to the noise present in the dataset.

11. How to handle categorical variables in the KNN Algorithm?

To handle the categorical variables we have to create **dummy variables** out of a categorical variable and include them instead of the original categorical variable. Unlike regression, create k dummies instead of (k-1).

For example, a categorical variable named **"Degree"** has 5 unique levels or categories. So we will create 5 dummy variables. Each dummy variable has 1 against its degree and else 0.

12. 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.**

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.

13. 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**.

14. Which algorithm can be used for value imputation in both categorical and continuous categories of data?

KNN is the only algorithm that can be used for the imputation of both categorical and continuous variables. It can be used as one of many techniques when it comes to handling missing values.

To impute a new sample, we determine the samples in the training set “nearest” to the new sample and averages the nearby points to impute. A **Scikit learn library of Python** provides a quick and convenient way to use this technique.

Note: NaNs are omitted while distances are calculated. Hence we replace the missing values with the average value of the neighbours. The missing values will then be replaced by the average value of their “neighbours”.

15. 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.

16. What are the things which should be kept in our mind while choosing the value of k in the KNN Algorithm?

If K is small, then results might not be reliable because the noise will have a higher influence on the result. If K is large, then there will be a lot of processing to be done which may adversely impact the performance of the algorithm.

So, the following things must be considered while choosing the value of K:

- K should be the square root of n (number of data points in the training dataset).
- K should be chosen as the odd so that there are no ties. If the square root is even, then add or subtract 1 to it.

17. 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.

18. 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.

19. Is it possible to use the KNN algorithm for Image processing?

Yes, KNN can be used for image processing by converting a 3-dimensional image into a single-dimensional vector and then using it as the input to the KNN algorithm.

20. How does KNN perform regression tasks?

In regression tasks, KNN predicts the output for a new data point by averaging the values of the K nearest neighbors. This method is based on the assumption that similar data points (based on distance metrics) have similar outputs.

21. Can KNN be utilized in building recommendation systems, and if so, how?

Yes, KNN can be used in recommendation systems, especially in collaborative filtering. It identifies similar users or items by calculating the distances between them and recommends items by looking at the most similar items or users' preferences.

You can read this article to know more about how to create recommendation system using KNN:

Movie Recommendation and Rating Prediction using K-Nearest Neighbors

22. What optimization techniques can improve KNN's performance?

Optimization techniques include using efficient data structures like KD-trees for faster distance calculations, dimensionality reduction to mitigate the curse of dimensionality, and selecting an appropriate distance measure to improve accuracy and computation time.

23. How does adding a new data point affect the KNN algorithm?

Adding a new data point to KNN does not require model retraining since KNN is a lazy learner. However, it may alter the decision boundary slightly for future predictions, especially if the new point significantly differs from existing data points.

24. In what way does KNN differ from neural networks in solving classification problems?

KNN is a simple, instance-based learning algorithm that doesn't learn a discriminative function from the data. In contrast, neural networks learn a complex function through layers of neurons and are better suited for capturing non-linear relationships in high-dimensional data.

25. How does logistic regression compare to KNN in classification tasks?

Logistic regression is a parametric approach that models the probability of a binary outcome based on one or more predictor variables. It assumes a linear decision boundary. KNN, on the other hand, can adapt to more complex decision boundaries by considering the proximity of neighboring points, without making any assumptions about the form of the decision boundary.

26. Why is feature selection important in KNN?

Feature selection is crucial in KNN to eliminate irrelevant or redundant features, which can significantly impact the distance calculations. Effective feature selection helps in reducing dimensionality, improving accuracy, and decreasing computation time.

27. How do different distance measures affect KNN's performance?

The choice of distance measure (e.g., Euclidean, Manhattan, Minkowski) can greatly affect KNN's performance. Different metrics may be more suitable for different types of data; for instance, Manhattan distance can be preferable for high-dimensional data as it tends to be more robust against the curse of dimensionality.

28. How can KNN be integrated with k-means clustering for enhanced data analysis?

KNN can be integrated with k-means by first using k-means to cluster the dataset into groups and then applying KNN within each cluster to classify or predict outcomes. This approach can reduce computational costs and improve prediction accuracy by narrowing down the search space.

29. What is the significance of the decision boundary in KNN, and how does it compare to decision trees?

The decision boundary in KNN is determined by the class of the K nearest neighbors to a point, resulting in a boundary that can adapt to the data's distribution without any assumptions. Decision trees, however, partition the space into regions based on feature thresholds, leading to rectangular partitioning. While KNN's boundary can be highly irregular, adapting closely to the data, decision trees provide a more structured approach, which can be easier to interpret but might not capture complex patterns as effectively as KNN.

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

The K Nearest Neighbor (KNN) algorithm finds widespread application across various fields, showcasing its adaptability and efficacy:

1. **Credit Rating Evaluation**: KNN assesses creditworthiness by comparing individuals' financial profiles with historical data, streamlining credit rating processes for lenders.
2. **Voter Behavior Prediction**: Utilized in political science, KNN forecasts voting patterns, aiding campaign strategies by predicting voter participation and party preferences.
3. **Handwriting Detection and OCR**: In OCR and handwriting recognition, KNN identifies characters and words in images, facilitating automated digitization of handwritten texts.
4. **Image Recognition**: KNN is used in image recognition to identify objects within images, serving applications from medical imaging diagnostics to facial recognition in security.
5. **Recommendation Systems**: Enhancing digital platforms, KNN personalizes recommendations for products, movies, or music based on user history and preferences.
6. **Video Recognition**: In video analysis, KNN helps monitor security footage or categorize video content, analyzing frames to detect specific objects or behaviors.

Conclusion

In summary, the K Nearest Neighbor (KNN) algorithm emerges as a powerful yet straightforward approach in supervised machine learning. Throughout this tutorial, we've explored its fundamentals, from understanding 'K' selection to addressing challenges like feature scaling and noise sensitivity. Despite limitations in handling large datasets , KNN's versatility finds practical applications across various domains, including credit rating evaluation and image recognition. Mastering KNN equips data scientists

with a robust tool for tackling classification and regression tasks, making it indispensable in modern machine learning endeavors.

Key Takeaways

- KNN is a versatile non-parametric algorithm used for both classification and regression problems, adapting to data without fixed parameters.
- The choice of 'K' affects KNN's accuracy, with odd values preferred to avoid ties and the need for hyperparameter tuning to find the optimal 'K'.
- Effective implementation of KNN requires feature scaling to ensure equal distance weighting and improved algorithm performance.
- KNN is a lazy learner that doesn't build a model but stores data, making it efficient for small datasets but challenging for large datasets due to high computation and sensitivity to noise.
- KNN finds practical applications in fields like image processing, credit rating, and political science, demonstrating its real-world versatility and utility.

Article Url - https://www.analyticsvidhya.com/blog/2021/05/interview-questions-to-test-your-skills-on-knn-algorithm/

CHIRAG GOYAL

I am currently pursuing my Bachelor of Technology (B.Tech) in Computer Science and Engineering from the Indian Institute of Technology Jodhpur(IITJ). I am very enthusiastic about Machine learning, Deep Learning, and Artificial Intelligence. Feel free to connect with me on Linkedin.