K-NEAREST NEIGHBORS

1. What is the K-Nearest Neighbor algorithm?

The K-Nearest Neighbor algorithm is a supervised learning algorithm that can be used for both classification and regression tasks. The algorithm works by finding the K nearest neighbors to a given data point, and then using those neighbors to predict the class or value of the data point.

2. Can you explain how to implement a simple kNN algorithm in code?

The kNN algorithm is a simple classification algorithm that can be used for a variety of tasks. To implement it, you will need to first calculate the distance between the new data point and all of the training data points. Once you have the distances, you will then need to find the k nearest neighbors and take the majority vote of those neighbors to determine the class of the new data point.

3. How does an implementation of kNN differ from other classification and regression algorithms like Gradient Descent, Random Forest, or Logistic Regression?

The main difference between kNN and other classification or regression algorithms is that kNN is a non-parametric algorithm, meaning that it does not make any assumptions about the underlying data. This makes kNN more flexible, but also means that it can be more computationally expensive. Other algorithms like Gradient Descent or Logistic Regression make assumptions about the data that allow them to be more efficient, but also less flexible.

4. What are some advantages of using kNN instead of decision trees?

Some advantages of using kNN instead of decision trees include the fact that kNN is often more accurate than decision trees, kNN can be used for regression as well as classification, and kNN is relatively simple to implement.

5. Can you give me some examples of where you would use the kNN algorithm?

kNN can be used for a variety of tasks, including classification, regression, and outlier detection. A few specific examples include:

– Classifying images of handwritten digits
– Predicting the price of a house based on its location and other features
– Detecting fraudulent credit card transactions
– Identifying genes that are related to a particular disease

6. When should you not use kNN?

kNN can be a very resource-intensive algorithm, so it is not always practical to use. Additionally, kNN can be less accurate than other algorithms when the data is not evenly distributed or when there are outliers in the data.

7. What is the difference between supervised and unsupervised learning?

Supervised learning is where the data is labeled and the algorithm is told what to do with it. Unsupervised learning is where the data is not labeled and the algorithm has to figure out what to do with it.

8. Why is Euclidean distance considered to be the best method for determining distances in most cases?

Euclidean distance is the best method for determining distances in most cases because it is the shortest distance between two points.

9. What do you understand about data normalization? Why is it important when working with kNN?

Data normalization is the process of scaling data so that it is within a certain range, usually between 0 and 1. This is important when working with kNN because if the data is not normalized, then the kNN algorithm will not work correctly.

10. Is there any way to improve the performance of kNN by performing feature selection before training our model? If yes, then what methods can be used for this purpose?

There are a few ways to improve the performance of kNN by performing feature selection before training our model. One way is to use a technique called feature selection which can help us select the most relevant features for our model. Another way is to use a technique called feature transformation which can help us transform our data into a more suitable form for kNN.

11. What's your understanding of locality sensitive hashing? How does it relate to kNN?

Locality sensitive hashing is a method of creating a hash function that is sensitive to the local structure of the data. This means that similar data points will tend to hash to the same value, while dissimilar data points will tend to hash to different values. This is useful for kNN because it means that we can quickly find the nearest neighbors of a given data point by simply looking at the points that hash to the same value.

12. Can you give me some examples of real world applications that use kNN?

kNN can be used for a variety of tasks, including but not limited to:
-Predicting whether or not a loan applicant will default
-Classifying types of plants
-Detecting fraudulent activity
-Recommending similar products to customers

13. What is the importance of choosing the right value for k?

The value of k is important because it determines how many neighbors will be used to make predictions. A lower value for k will make the model more sensitive to outliers, while a higher value will make the model more resistant to them. The right value for k will depend on the data set and the specific problem you are trying to solve.

14. How can you determine which features to include in a machine learning model?

One way to determine which features to include in a machine learning model is to use a technique called feature selection. This is a process where you select a subset of the features available to you that you believe will be most predictive of the target variable. There are a number of different methods for doing feature selection, but they all essentially boil down to trying to find the best combination of features that will result in the most accurate predictions.

15. What's your opinion on curse of dimensionality and its implications on kNN?

The curse of dimensionality is a real problem when working with kNN. As the number of dimensions increases, the data becomes more and more sparse. This can lead to issues with overfitting, as well as problems with the algorithm not being able to find the nearest neighbors at all. It's important to be aware of this issue and to take steps to mitigate it, such as using dimensionality reduction techniques.

16. What is the importance of splitting data into training and test sets?

One of the key steps in using the k-nearest neighbor algorithm is to split your data into a training set and a test set. The training set is used to train the model, while the test set is used to evaluate the performance of the model. It is important to split the data in this way so that you can get an accurate assessment of how well the model is performing.

17. What are some situations where kNN performs poorly?

kNN can perform poorly when the data is very noisy or when the data is not linearly separable. Additionally, kNN can be computationally intensive if there are a large number of training examples or if the dimensionality of the data is high.

18. What are some good guidelines for preparing data for kNN algorithms?

In general, you want to make sure that your data is as clean as possible before using kNN. This means getting rid of any missing values, outliers, and making sure that all of your features are on the same scale. Additionally, you may want to consider performing some dimensionality reduction techniques if you have a large number of features, as kNN can be computationally intensive.

19. What is cross validation? Why is it necessary to perform cross validation?

Cross validation is a technique used to assess the accuracy of a machine learning model. It does this by splitting the data into a training set and a test set, then training the model on the training set and testing it on the test set. This allows you to see how well the model performs on data it hasn't seen before, which is important in order to gauge its real-world accuracy.

20. What types of errors can occur when classifying data using Naive Bayes?

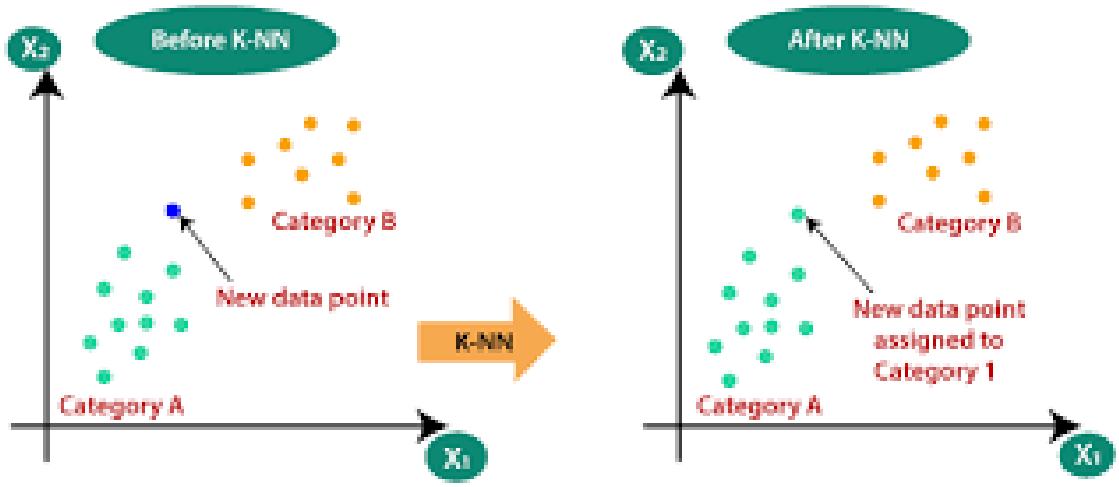
The two main types of errors that can occur are false positives and false negatives. A false positive occurs when a data point is classified as belonging to a certain class when it actually does not. A false negative occurs when a data point is classified as not belonging to a certain class when it actually does.

21. 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.



22. 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.

23. What is "K" 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.

24. 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.

25. 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:

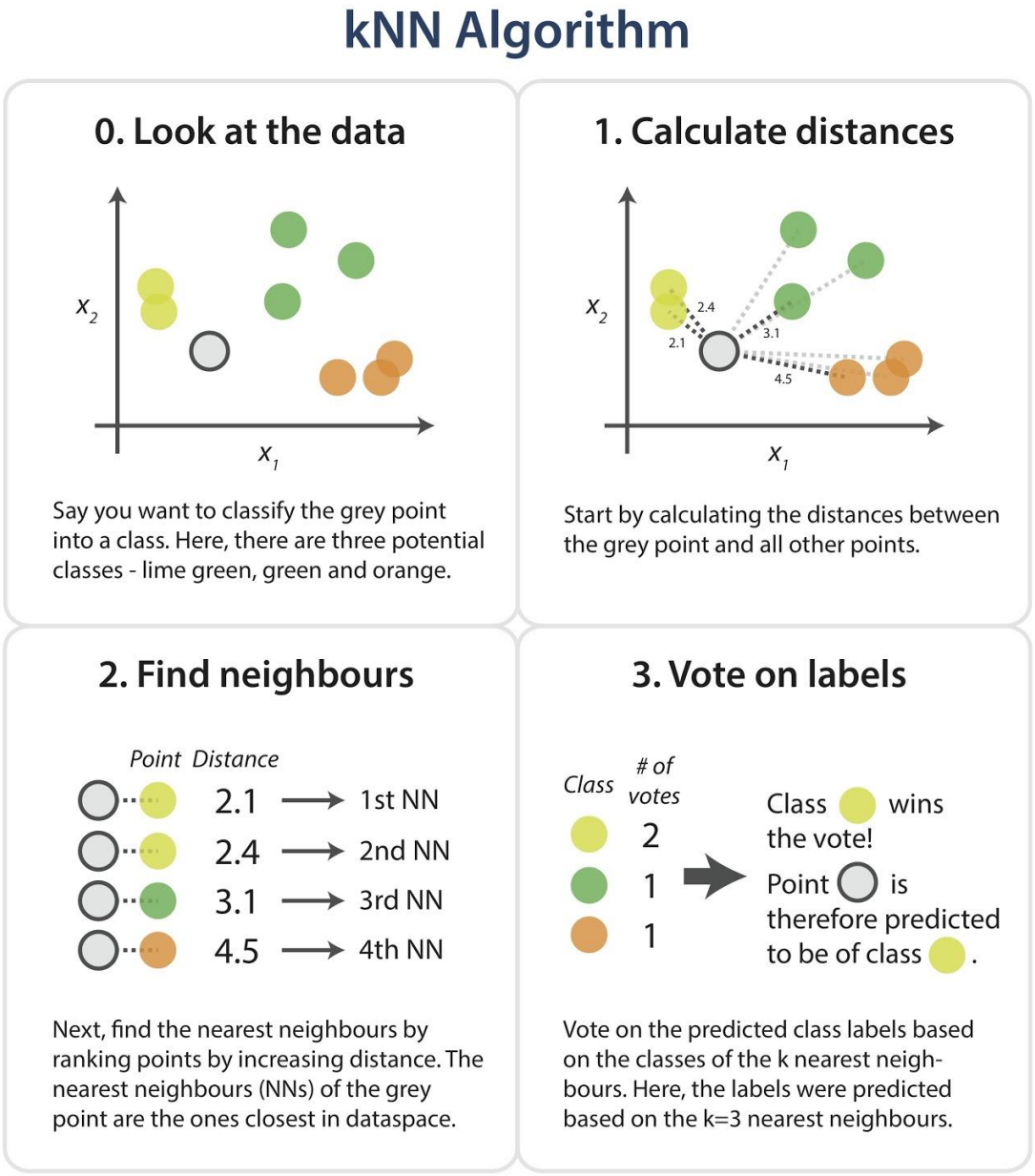
Step-1: Calculate the distances of test point to all points in the training set and store them

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



26. 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.

27. 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(N^3 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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:

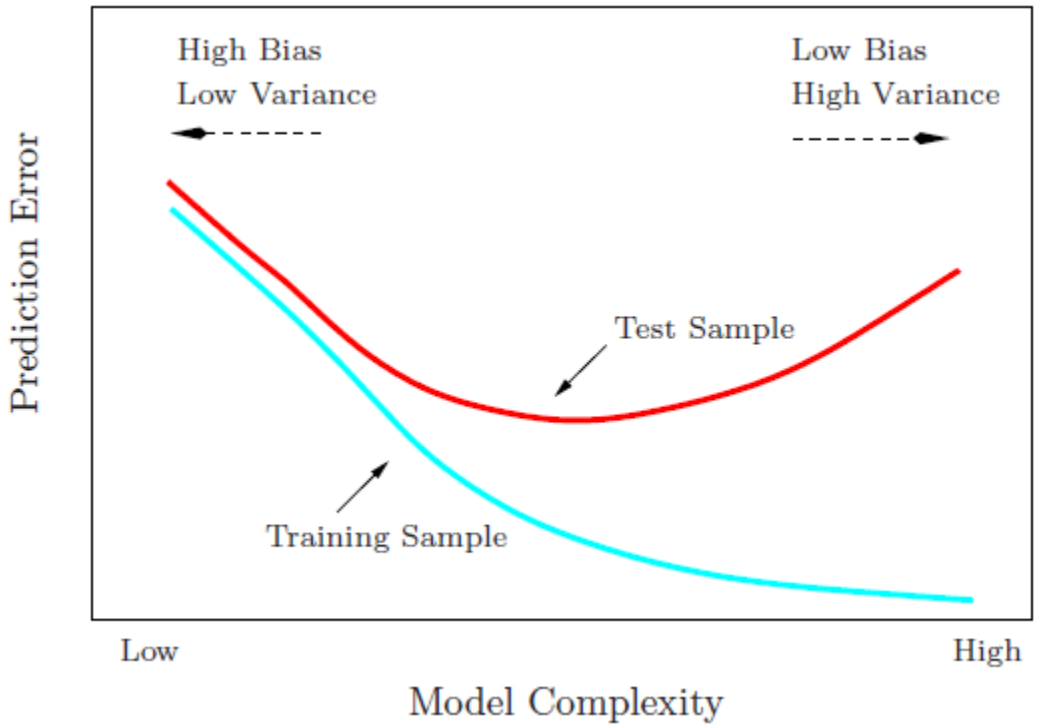
The larger the value of K, the higher is the accuracy. If K is too large, then our model is under-fitted. As a result, the error will go up again. So, to prevent your model from under-fitting it should retain the generalization capabilities otherwise there are fair chances that your model may perform 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**.



34. 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".

35. 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.

36. 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.

37. 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.

38. 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.

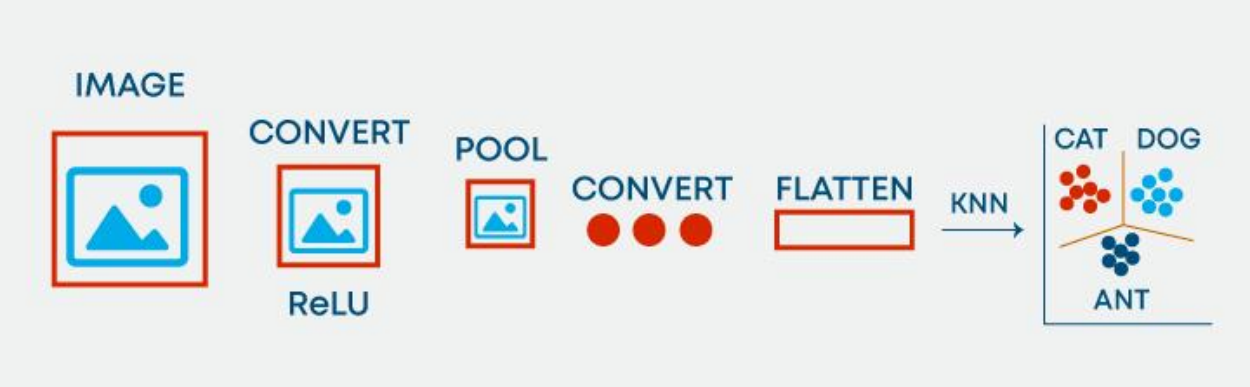
2. Does not work well with high dimensions: KNN algorithms generally do not work well with high dimensional data since, with the increasing number of dimensions, it becomes difficult to calculate the distance for each dimension.

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.

39. 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.



40. 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**.

41) [True or False] k-NN algorithm does more computation on test time rather than train time.

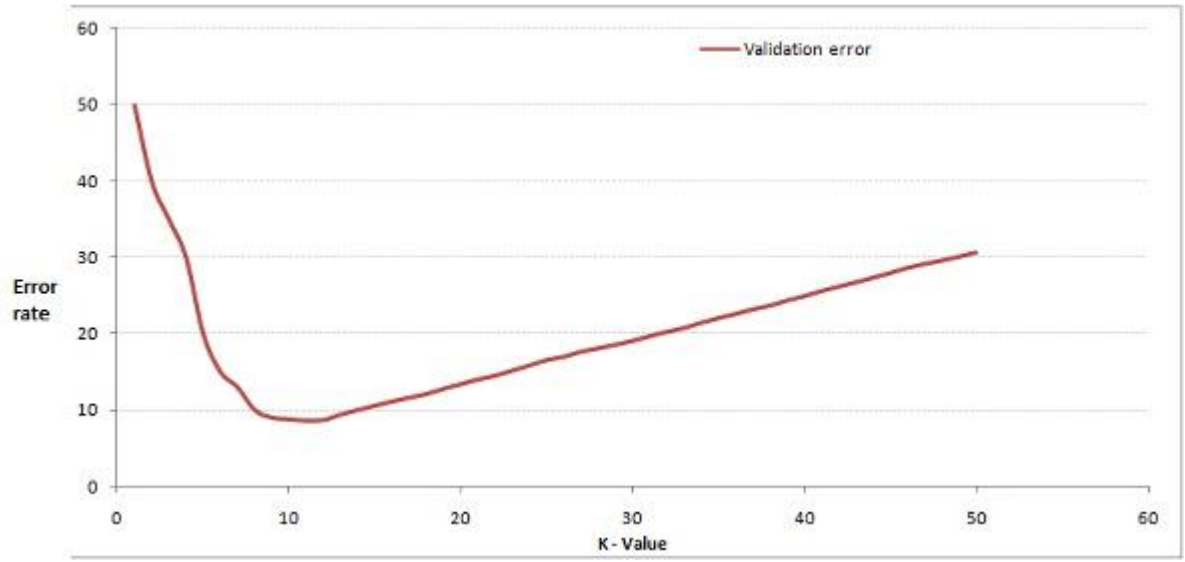
A) TRUE

B) FALSE

Solution: A The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.

In the testing phase, a test point is classified by assigning the label which are most frequent among the K training samples nearest to that query point – hence higher computation.

42) In the image below, which would be the best value for k assuming that the algorithm you are using is k-Nearest Neighbor.



A) 3
B) 10
C) 20
D 50

Solution: B Validation error is the least when the value of k is 10. So it is best to use this value of k

43) Which of the following distance metric can not be used in k-NN?

A) Manhattan

B) Minkowski

C) Tanimoto

D) Jaccard

E) Mahalanobis

F) All can be used

Solution: F All of these distance metric can be used as a distance metric for k-NN.

44) Which of the following option is true about k-NN algorithm?

A) It can be used for classification

B) It can be used for regression

C) It can be used in both classification and regression

Solution: C We can also use k-NN for regression problems. In this case the prediction can be based on the mean or the median of the k-most similar instances.

45) Which of the following statement is true about k-NN algorithm?

1. k-NN performs much better if all of the data have the same scale
2. k-NN works well with a small number of input variables (p), but struggles when the number of inputs is very large
3. k-NN makes no assumptions about the functional form of the problem being solved

A) 1 and 2

B) 1 and 3

C) Only 1

D) All of the above

Solution: D The above mentioned statements are assumptions of kNN algorithm

46) Which of the following machine learning algorithm can be used for imputing missing values of both categorical and continuous variables?

A) K-NN

B) Linear Regression

C) Logistic Regression

Solution: A

k-NN algorithm can be used for imputing missing value of both categorical and continuous variables.

47) Which of the following is true about Manhattan distance?

A) It can be used for continuous variables

B) It can be used for categorical variables

C) It can be used for categorical as well as continuous

D) None of these

Solution: A

Manhattan Distance is designed for calculating the distance between real valued features.

48) Which of the following distance measure do we use in case of categorical variables in k-NN?

1. Hamming Distance
2. Euclidean Distance
3. Manhattan Distance

A) 1

B) 2

C) 3

D) 1 and 2

E) 2 and 3

F) 1,2 and 3

Solution: A

Both Euclidean and Manhattan distances are used in case of continuous variables, whereas hamming distance is used in case of categorical variable.

49) Which of the following will be Euclidean Distance between the two data point A(1,3) and B(2,3)?

A) 1
B) 2
C) 4
D) 8

Solution: A

sqrt((1-2)^2 + (3-3)^2) = sqrt(1^2 + 0^2) = 1

50) Which of the following will be Manhattan Distance between the two data point A(1,3) and B(2,3)?

A) 1
B) 2

C) 4
D) 8

Solution: A

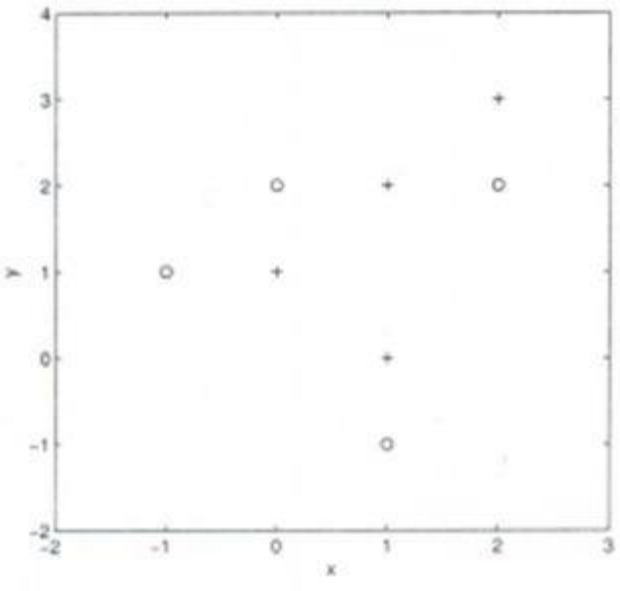
sqrt(mod((1-2)) + mod((3-3))) = sqrt(1 + 0) = 1

Context: 51-52

Suppose, you have given the following data where x and y are the 2 input variables and Class is the dependent variable.

x	y	Class
-1	1	$-$
0	1	$+$
0	2	$-$
1	-1	$-$
1	0	$+$
1	2	$+$
2	2	$-$
2	3	$+$

Below is a scatter plot which shows the above data in 2D space.



51) Suppose, you want to predict the class of new data point x=1 and y=1 using eucludian distance in 3-NN. In which class this data point belong to?

A) + ClassB) – ClassC) Can't say

D) None of these

Solution: A

All three nearest point are of +class so this point will be classified as +class.

52) In the previous question, you are now want use 7-NN instead of 3-KNN which of the following x=1 and y=1 will belong to?

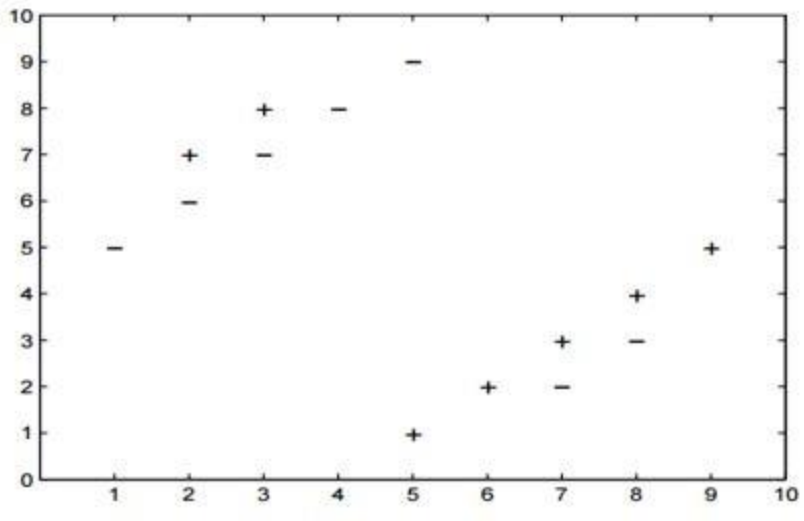
A) + ClassB) – ClassC) Can't say

Solution: B

Now this point will be classified as – class because there are 4 – class and 3 +class point are in nearest circle.

Context 53-54:

Suppose you have given the following 2-class data where "+" represent a postive class and "" is represent negative class.



53) Which of the following value of k in k-NN would minimize the leave one out cross validation accuracy?

A) 3
B) 5
C) Both have same
D) None of these

Solution: B

5-NN will have least leave one out cross validation error.

54) Which of the following would be the leave on out cross validation accuracy for k=5?

A) 2/14
B) 4/14
C) 6/14
D) 8/14
E) None of the above

Solution: E In 5-NN we will have 10/14 leave one out cross validation accuracy.

55) Which of the following will be true about k in k-NN in terms of Bias?

A) When you increase the k the bias will be increases
B) When you decrease the k the bias will be increases
C) Can't say
D) None of these

Solution: A

large K means simple model, simple model always condider as high bias

56) Which of the following will be true about k in k-NN in terms of variance?

A) When you increase the k the variance will increases
B) When you decrease the k the variance will increases
C) Can't say
D) None of these

Solution: B

Simple model will be consider as less variance model

57) The following two distances(Eucludean Distance and Manhattan Distance) have given to you which generally we used in K-NN algorithm. These distance are between two points A(x1,y1) and B(x2,Y2).

Your task is to tag the both distance by seeing the following two graphs. Which of the following option is true about below graph ?

A) Left is Manhattan Distance and right is euclidean Distance
B) Left is Euclidean Distance and right is Manhattan Distance

C) Neither left or right are a Manhattan Distance
D) Neither left or right are a Euclidian Distance

Solution: B Left is the graphical depiction of how euclidean distance works, whereas right one is of Manhattan distance.

58) When you find noise in data which of the following option would you consider in k-NN?

A) I will increase the value of k
B) I will decrease the value of k
C) Noise can not be dependent on value of k
D) None of these

Solution: A To be more sure of which classifications you make, you can try increasing the value of k.

59) In k-NN it is very likely to overfit due to the curse of dimensionality. Which of the following option would you consider to handle such problem?

1. Dimensionality Reduction
2. Feature selection

A) 1

B) 2

C) 1 and 2

D) None of these

Solution: C

In such case you can use either dimensionality reduction algorithm or the feature selection algorithm

60) Below are two statements given. Which of the following will be true both statements?

1. k-NN is a memory-based approach is that the classifier immediately adapts as we collect new training data.
2. The computational complexity for classifying new samples grows linearly with the number of samples in the training dataset in the worst-case scenario.

A) 1

B) 2

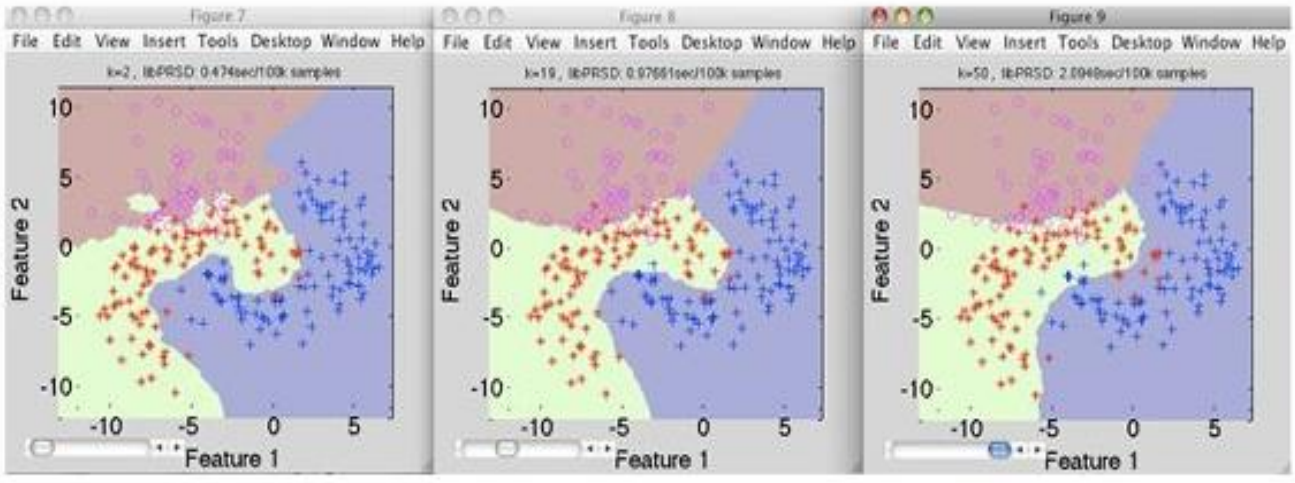
C) 1 and 2

D) None of these

Solution: C

Both are true and self explanatory

61) Suppose you have given the following images(1 left, 2 middle and 3 right), Now your task is to find out the value of k in k-NN in each image where k1 is for 1st, k2 is for 2nd and k3 is for 3rd figure.



A) k1 > k2> k3
B) k1<k2
C) k1 = k2 = k3
D) None of these

Solution: D
Value of k is highest in k3, whereas in k1 it is lowest

62) Which of the following value of k in the following graph would you give least leave one out cross validation accuracy?

A) 1
B) 2
C) 3
D) 5

Solution: B

If you keep the value of k as 2, it gives the lowest cross validation accuracy. You can try this out yourself.

63) A company has build a kNN classifier that gets 100% accuracy on training data. When they deployed this model on client side it has been found that the model is not at all accurate. Which of the following thing might gone wrong?

Note: Model has successfully deployed and no technical issues are found at client side except the model performance

A) It is probably a overfitted model

B) It is probably a underfitted model
C) Can't say
D) None of these

Solution: A In an overfitted module, it seems to be performing well on training data, but it is not generalized enough to give the same results on a new data.

64) You have given the following 2 statements, find which of these option is/are true in case of k-NN?

1. In case of very large value of k, we may include points from other classes into the neighborhood.
2. In case of too small value of k the algorithm is very sensitive to noise

A) 1

B) 2

C) 1 and 2

D) None of these

Solution: C

Both the options are true and are self explanatory.

65) Which of the following statements is true for k-NN classifiers?

A) The classification accuracy is better with larger values of k
B) The decision boundary is smoother with smaller values of k
C) The decision boundary is linear
D) k-NN does not require an explicit training step

Solution: D Option A: This is not always true. You have to ensure that the value of k is not too high or not too low. Option B: This statement is not true. The decision boundary can be a bit jagged

Option C: Same as option B

Option D: This statement is true

66) True-False: It is possible to construct a 2-NN classifier by using the 1-NN classifier?

A) TRUE
B) FALSE

Solution: A You can implement a 2-NN classifier by ensembling 1-NN classifiers

67) In k-NN what will happen when you increase/decrease the value of k?

A) The boundary becomes smoother with increasing value of K
B) The boundary becomes smoother with decreasing value of K
C) Smoothness of boundary doesn't dependent on value of K
D) None of these

Solution: A
The decision boundary would become smoother by increasing the value of K

68) Following are the two statements given for k-NN algorthm, which of the statement(s) is/are true?

1. We can choose optimal value of k with the help of cross validation
2. Euclidean distance treats each feature as equally important

A) 1

B) 2

C) 1 and 2

D) None of these

Solution: C

Both the statements are true

Context 69-70:

Suppose, you have trained a k-NN model and now you want to get the prediction on test data. Before getting the prediction suppose you want to calculate the time taken by k-NN for predicting the class for test data.
Note: Calculating the distance between 2 observation will take D time.

69) What would be the time taken by 1-NN if there are N(Very large) observations in test data?

A) N*D
B) N*D*2
C) (N*D)/2
D) None of these

Solution: A The value of N is very large, so option A is correct

70) What would be the relation between the time taken by 1-NN,2-NN,3-NN.

A) 1-NN >2-NN >3-NN
B) 1-NN < 2-NN < 3-NN
C) 1-NN ~ 2-NN ~ 3-NN
D) None of these

Solution: C The training time for any value of k in kNN algorithm is the same.