Logo

Description automatically generated with low confidence

# Top 50 data science interview questions with answers

|  |  |
| --- | --- |
| Q # 1 | Why is Naive Bayes referred to as Naive? |
| Answer | Naïve Bayes is a probabilistic algorithm based on Bayes’ theorem of Conditional probability.  It is called "naive" because Naive Bayes assumes that each feature in the dataset is independent of all the other features given the class label. This assumption is considered "naive" because it oversimplifies the relationship between the features and the class label.  In reality, features are often correlated, meaning that the presence or absence of one feature can affect the likelihood of other features being present or absent as well.  In Mathematical term, it makes the math simple by simplifying the likelihood terms in the expression.  Example:  Instead of this,    We calculate,    More detailed explanation here in my video:  <https://www.youtube.com/watch?v=-b9deWb8oaI&t=1733s> |

|  |  |
| --- | --- |
| Q # 2 | Explain bias, variance tradeoff. |
| Answer | The bias-variance tradeoff is a key concept in machine learning that describes the balance between two sources of error in a model: bias and variance.  Bias represents the error that occurs when a model is too simple and cannot capture the complexity of the underlying data. This results in poor performance on both the training and test data. [underfit]  On the other hand, variance represents the error that occurs when a model is too complex and overfits to the training data, leading to high performance on the training data but poor performance on the test data.[overfit]  The goal of machine learning is to find a model that generalizes well to new data, which requires striking a balance between bias and variance.  Models with high bias tend to underfit the data, while models with high variance tend to overfit the data.  The key is to find the optimal level of model complexity that minimizes both bias and variance, resulting in a model that can accurately and reliably predict outcomes on new data.  Techniques such as regularization and cross-validation can be used to help balance bias and variance and improve the generalizability of a model. |

|  |  |
| --- | --- |
| Q # 3 | What is Kernel and Kernel Trick in SVM |
| Answer | The "kernel" or "Kernel function" or "Kernel trick" is a method that allows Support Vector Machines (SVMs) to efficiently operate in high-dimensional feature spaces without explicitly computing the coordinates of data in that space.  The kernel trick works by implicitly mapping the input data into a high-dimensional feature space using a kernel function.  A kernel function is a mathematical function that takes in two input vectors and computes a scalar value (dot product) representing the similarity between them.  👉 But why it is called a trick ?  Think of feature engineering: In case of FE, We would "actually" transform the data to higher dimension (lot of computation). But in case of Kernel Trick we are not actually calculating the co-ordinates in higher dimension. We are only calculating the dot product of two vector in higher dimension(Comparatively less computation).  We will evaluate this dot product for every pair of observation. This dot product ( a scaler quantity) acts as a similarity matrix and help SVM compute a decision boundary , or hyperplane, in the high-dimensional feature space, which can then be used to classify new data points.  Details here in my linkedIn Post  <https://www.linkedin.com/posts/anas-p-296b38258_datascience-statistics-analytics-activity-7044161638446297088-Q_kS/?utm_source=share&utm_medium=member_ios>  Youtube Video explaining the same:  https://www.youtube.com/watch?v=VBCk8bdMjR8 |

|  |  |
| --- | --- |
| Q # 4 | Why doesn’t gradient descent methods always converge / doesn’t converge to the same point? |
| Answer | Gradient descent is an iterative optimization algorithm used in machine learning to find the minimum of a cost function.  In theory, gradient descent should always converge to the same minimum point, assuming certain conditions are met, such as the cost function being convex (Linear regression ) and the learning rate being properly tuned.  However, in practice, there are several reasons why gradient descent may not converge to the same point every time. These include:  Non-convex cost functions: In some cases, the cost function may have multiple local minima, and the choice of initial parameters and learning rate may cause the algorithm to converge to different local minima. [Deep Learning]  Stochasticity: In stochastic gradient descent, the algorithm randomly samples a subset of the training data at each iteration, which can introduce randomness into the optimization process. As a result, the algorithm may converge to slightly different points on different runs.  Learning rate: The learning rate determines the step size of the algorithm at each iteration. If the learning rate is too high, the algorithm may overshoot the minimum, while if the learning rate is too low, the algorithm may converge too slowly or get stuck in a local minimum.  Youtube video where I have explained Gradient Descent :  <https://www.youtube.com/watch?v=xs-DoKGbHtk&list=PL47eid5T-F5TKdsfhQolCHhyhnCAZVclC> |

|  |  |
| --- | --- |
| Q # 5 | How is AUC different from ROC? |
| Answer | Confusion Matrix    FP = Type 1 error  FN = Type 2 error          ROC is a plot of the True Positive Rate (TPR) against the False Positive Rate (FPR) for different threshold values.  The area under the ROC curve (AUC) is a summary statistic that measures the classifier's overall performance regardless of the threshold value chosen. The AUC ranges from 0 to 1, with 0.5 indicating a random classifier and 1 indicating a perfect classifier.  On the other hand, AUC is a scalar value that represents the probability that a randomly chosen positive example will be ranked higher than a randomly chosen negative example. It is calculated by integrating the ROC curve and ranges from 0 to 1, with higher values indicating better performance.  Let’s say, we have two use cases:   1. Medical model – Disease detection – Here, we should lower false negative. ( Disease left undetected ) 2. Spam – Here we should lower false positive. ( Avoid sending some important email into SPAM folder )   In summary, ROC is a plot of the TPR against FPR – This is used to chose which threshold is better for your particular usecase and dataset.  while AUC is a scalar value that summarizes the overall performance of a binary classification model based on the ROC curve. Let’s see if you have logistic regression and random forest. You can compare their performance using AUC. A higher AUC represents better classifier.  . |