













**Ans: The key assumption behind the Naive Bayes algorithm is the conditional independence assumption. Specifically, it assumes that all features (or predictors) are independent of each other given the class label.**





**Ans:  Gaussian Naive Bayes**

* **Assumes that the continuous features follow a normal (Gaussian) distribution.**
* **Commonly used when features are continuous-valued.**
* **Example: Predicting a class based on features like height, weight, or age.**

** Multinomial Naive Bayes**

* **Suitable for discrete count data, such as word counts in text classification problems.**
* **Assumes features represent frequencies or counts.**
* **Widely used in document classification (e.g., spam detection, topic classification).**

** Bernoulli Naive Bayes**

* **Assumes binary features (features that are either 0 or 1).**
* **Works with boolean/yes-no type features.**
* **Used in binary occurrence scenarios, such as whether a word occurs in a document or not.**

** Complement Naive Bayes (an extension of Multinomial)**

* **Designed to work better on imbalanced datasets by using statistics from the complement of each class.**
* **Often improves performance for text classification.**







**Ans:  Scalability with Many Features:  
Since Naive Bayes assumes features are independent given the class, it simplifies the joint probability calculation into a product of individual feature probabilities. This means the model doesn’t need to estimate complex joint distributions of features, which grows exponentially with more features. Instead, it handles each feature separately, making it scalable to datasets with thousands or even millions of features (e.g., text data with many words).**

** Less Data Required to Train:  
Because it models each feature independently, it requires fewer parameters to estimate, so it can perform well even when the number of training samples is small relative to the number of features.**

** Fast Training and Prediction:  
The simplicity of calculations allows for very fast model training and prediction, which is crucial in high-dimensional spaces.**

** Robustness to Irrelevant Features:  
Naive Bayes can handle irrelevant features fairly well since they have minimal impact individually due to the multiplication of probabilities**









**Ans: The main purpose of using k-fold cross-validation is to evaluate the performance and generalizability of a machine learning model more reliably by reducing the bias and variance that can occur when using a single train-test split**







Ans: In a traditional train-test split, the dataset is divided once into two parts- one for training and one for testing – which may not fully represent the dataset variability.

In k-fold cross-validation, the dataset is divided into k equal parts(folds). The model is trained and tested k times, each time using a different fold as the test set and the remaining k-1 folds as the training set. This results in a more reliable estimate of model performance by reducing the impact of how the data is split.







Ans: Stratified k-fold cross-validation ensures that each fold has approximately the same class distribution as the original dataset. This is especially important for imbalanced datasets, as it maintains the representativeness of each class in every fold, leading to more accurate and fair evaluation of the model’s performance.







Ans: n k-fold cross-validation, k represents the number of equal-sized folds (subsets) that the dataset is divided into. The model is trained and evaluated k times, each time using a different fold as the validation set and the remaining folds for training.

-Effect of changing k:

-Smaller k (e.g., k=5):

-Faster computation

-Higher bias, lower variance in evaluation

-Less reliable estimate of model performance

-Larger k (e.g., k=10 or k=n, where n is the number of data points – known as Leave-One-Out):

-Slower computation

-Lower bias, higher variance

-More reliable performance estimate, but more computationally expensive

-So, changing k balances the trade-off between computational cost and accuracy of the model evaluation.







Ans: K-fold cross-validation helps prevent overfitting by testing the model on multiple different validation sets instead of just one. This ensures that the model performs well across various subsets of the data, not just the training data. It discourages the model from learning patterns that are specific to only one part of the data, leading to better generalization







Ans: K-fold cross-validation allows for a fair comparison of different models (or hyperparameter settings) by providing an average performance score across all folds. This helps in selecting the model that performs consistently well, rather than one that happens to perform well on a single train-test split.





Ans: A major disadvantage of k-fold cross-validation is its high computational cost, especially with large datasets or complex models. Since the model is trained and validated k times, it can significantly increase the time and resources required for model evaluation.









Ans: A confusion matrix is a table used to evaluate the performance of a classification model by comparing actual and predicted labels. It shows how many instances were correctly or incorrectly classified across different classes, helping to understand the types of errors the model is making.







Ans: True Positives (TP): Cases where the model correctly predicted the positive class.

True Negatives (TN): Cases where the model correctly predicted the negative class.







Ans: False Positives (FP): Cases where the model incorrectly predicted positive, but the actual class was negative.

False Negatives (FN): Cases where the model incorrectly predicted negative, but the actual class was positive.





Ans: Accuracy is the proportion of correct predictions (both true positives and true negatives) among the total number of predictions.

Accuracy= TP+TN/TP+TN+FP+FN

Where: TP = True Positives

TN = True Negatives

FP = False Positive

FN = False Negative







Ans: Precision (also called Positive Predictive Value) measures how many of the positively predicted cases were actually positive.

Formula:

Precision = TP/TP+FP

Where:

-TP = True Positive

-FP=False Positive







Ans: Recall (also called Sensitivity or True Positive Rate) measures how many of the actual positive cases were correctly identified.

Formula: Recall= TP/TP+FN

Where:

TP = True Positives

FN = False Negatives







Ans: F1 Score is the harmonic mean of precision and recall, and it provides a single metric that balances both.

This is a correct and concise explanation. If you want to make it slightly more informative, you could revise the answer like this:

The F1 score is the harmonic mean of precision and recall, calculated as:

Formula:F1 =2× Precision+Recall/Precision×Recall







Ans: A high False Positive Rate (FPR) indicates that the model is frequently misclassifying negative instances as positive. In other words, it is incorrectly predicting many negatives as positives. This suggests poor specificity, meaning the model struggles to correctly identify actual negatives.

Implications:

This can be problematic in applications like spam detection or medical diagnostics, where falsely flagging something as positive (e.g., marking a legitimate email as spam or diagnosing a healthy person with a disease) may lead to serious consequences.







Ans: A low False Negative (FN) rate means that the model is correctly identifying most of the actual positive cases. In other words, very few positive instances are being misclassified as negative. This indicates high sensitivity or recall, which is important in situations where missing a positive case (like a disease diagnosis) could be costly or dangerous.







Ans: Accuracy can be misleading in imbalanced datasets because it may be high even if the model is performing poorly on the minority class. For example, in a dataset where 95% of the samples are negative, a model that always predicts "negative" will be 95% accurate — but it fails completely to identify the positive cases. Therefore, metrics like precision, recall, F1-score, or AUC-ROC are better suited for evaluating performance in imbalanced scenarios.

