24_May_ML_Assignment2

July 13, 2025

1. What is regression analysis

Regression analysis is a statistical technique used to model and analyze the relationship between a dependent variable (target) and one or more independent variables (features).

`Purpose:`

Predict numeric outcomes

Understand the influence of input variables on the output

2. Explain the difference between linear and nonlinear regression

Aspect	Linear Regression	Nonlinear Regression
Relationship	Assumes a **linear relationship**	Assumes a **nonlinear relationship**
Equation	$y = b_0 + b_1 x$	$y = b_0 + b_1 x + b_2 x^2 + ...$ or exponential, etc.
Interpretability	Easier to interpret	Often complex to interpret
Computation	Simpler and faster	More computationally intensive

3. What is the difference between simple linear regression and multiple linear regression

Feature	Simple Linear Regression	Multiple Linear Regression
Number of features	One independent variable	Two or more independent variables
Equation	$y = b_0 + b_1 x$	$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n$
Complexity	Lower	Higher

4. How is the performance of a regression model typically evaluated

`Metrics used:`

Mean Absolute Error (MAE): Average of absolute differences between predicted and actual values.

Mean Squared Error (MSE): Average of squared differences.

Root Mean Squared Error (RMSE): Square root of MSE.

R-squared (R^2): Proportion of variance in the target explained by the model.

5. What is overfitting in the context of regression models

Overfitting occurs when a model learns the training data too well, including noise and outliers, and fails to generalize to new/unseen data.

`Symptoms:`

High accuracy on training data

Poor performance on validation/test data

`Common causes:`

Too many features

Complex model

Small dataset

6. What is logistic regression used for

Logistic Regression is used for binary or multi-class classification, not regression.

`Examples:`

Spam detection (Spam vs. Not Spam)

Disease prediction (Yes/No)

Credit default (Defaulted/Not Defaulted)

7. How does logistic regression differ from linear regression

Feature	Linear Regression	Logistic Regression
Output	Continuous values	Probability between 0 and 1
Use Case	Regression (predicting numbers)	Classification (predicting classes)
Output Function	Linear equation	**Sigmoid function**
Target Variable	Real number	Categorical (e.g., 0 or 1)

8. Explain the concept of odds ratio in logistic regression

The odds ratio compares the odds of an event happening to it not happening.

Odds= P(event)/1−P(event)

`Odds Ratio:`

1: Positive association

<1: Negative association

=1: No effect

In logistic regression, the exponentiated coefficients (e^) represent odds ratios for each predictor.

9. What is the sigmoid function in logistic regression

The sigmoid function maps any real-valued number to a value between 0 and 1, representing probability.

(z)= 1/1+e^−z

Where z=b0+b1x1+b2x2+...

Used in logistic regression to convert linear output to a probability of class membership.

10. How is the performance of a logistic regression model evaluated

Evaluation Metrics:

Accuracy: % of correctly predicted labels

Precision: Correct positive predictions / total predicted positives

Recall (Sensitivity): Correct positive predictions / actual positives

F1-score: Harmonic mean of precision and recall

ROC-AUC Score: Measures classification performance across thresholds

Confusion Matrix: Shows TP, TN, FP, FN

11. What is a decision tree

A Decision Tree is a supervised learning algorithm used for classification and regression. It splits data into branches based on feature values, forming a tree-like structure where:

Internal nodes → decision based on a feature

Leaf nodes → prediction (class/label or value)

12. How does a decision tree make predictions

Prediction is made by traversing the tree from the root to a leaf:

At each node, the algorithm checks a feature condition (e.g., X < 5)

Based on the outcome, it moves left or right

When a leaf node is reached, the associated value (class/number) is returned as the prediction

13. What is entropy in the context of decision trees

Entropy measures the impurity or randomness in a dataset.

Entropy(S)=−n i=1 (pilog base2(pi))

pi= proportion of class Lower entropy → purer data → better split

Used in ID3 algorithm to decide the best feature for splitting.

14. What is pruning in decision trees

Pruning reduces the size of a decision tree by removing branches that provide little or no value.

Types:

Pre-pruning: Stop tree growth early based on conditions (e.g., max depth, min samples).

Post-pruning: Grow the full tree and then remove nodes using validation performance.

Prevents overfitting and improves generalization

15. How do decision trees handle missing values

Decision trees can handle missing values by:

Skipping missing features during split decisions

Surrogate splits: Use another feature that closely mimics the original split

Assigning probabilities: Split instance based on observed distributions of known values

16. What is a support vector machine (SVM)

SVM is a supervised learning algorithm used for binary classification, multi-class classification, and regression.

It finds the optimal hyperplane that best separates classes with the maximum margin.

17. Explain the concept of margin in SVM

Margin is the distance between the hyperplane and the nearest data points from each class.

Maximum Margin Classifier: Chooses the hyperplane that maximizes this margin

Larger margin = better generalization

18. What are support vectors in SVM

Support Vectors are the data points closest to the separating hyperplane.

They are critical in defining the position and orientation of the hyperplane.

Only support vectors affect the model; removing others won't change the decision boundary.

19. How does SVM handle non-linearly separable data

Two main techniques:

1. Soft Margin SVM: Allows some misclassifications using a penalty parameter C.
2. Kernel Trick:

   Transforms data into higher dimensions using a kernel (e.g., RBF, polynomial).

   Makes non-linear data linearly separable in transformed space.

20. What are the advantages of SVM over other classification algorithms

Advantages of SVM:

Works well in high-dimensional spaces

Effective when number of features > number of samples

Robust to overfitting (especially with proper regularization)

Can handle non-linear data using kernel trick

Uses only support vectors, making it memory-efficient

Limitations:

Not ideal for large datasets (training is slow)

Performance depends on choice of kernel and parameters

21. What is the Naïve Bayes algorithm

Naïve Bayes is a supervised learning algorithm based on Bayes' Theorem, used primarily for classification tasks.

It assumes that features are conditionally independent given the class label.

`Bayes' Theorem:`

$P(Y X)= P(X Y) P(Y)/P(X)$

```
Where:
Y = target class
X = features
P(Y X) = posterior
P(X Y) = likelihood
P(Y) = prior
P(X) = evidence
```

22. Why is it called "Naïve" Bayes

It's called "Naïve" because it assumes all features are independent of each other given the class label.

This assumption is rarely true in practice, hence "naïve", but it still performs surprisingly well for many tasks.

23. How does Naïve Bayes handle continuous and categorical features

`Categorical features:`

Uses frequency-based probabilities (e.g., count of class-label matches)

`Continuous features:`

Assumes a distribution (typically Gaussian/normal)

Computes probabilities using the probability density function (PDF) of that distribution

24. Explain the concept of prior and posterior probabilities in Naïve Bayes

Prior (P(Y)): Probability of a class before seeing the data.

`Example: If 70 out of 100 emails are spam = 0.7`

Posterior (P(Y X)): Updated probability of the class after seeing the data/features.

`Calculated using Bayes’ Theorem`

25. What is Laplace smoothing and why is it used in Naïve Bayes

Laplace smoothing (add-one smoothing) is used to handle zero probabilities when a category in a feature is not present in the training data for a class.

`Formula (for categorical data):`

$P(xi y) = count(xi,y)+1/count(y)+k$

Where:

```
k = number of categories in feature
Prevents multiplication by zero in likelihood calculation
```

26. Can Naïve Bayes be used for regression tasks

While Naïve Bayes is primarily used for classification, there is a variant called Naïve Bayes regression, but it's rarely used in practice.

Other models like linear regression or decision tree regression are typically preferred for continuous outputs.

27. How do you handle missing values in Naïve Bayes

Ignoring missing features during probability calculation

Imputing missing values using:

Mean/median (for continuous features)

Mode (for categorical features)

Using probabilistic imputation based on class-specific statistics

28. What are some common applications of Naïve Bayes

Text classification (spam detection, sentiment analysis)

Email filtering

Medical diagnosis

Document categorization

Recommendation systems

It works well with high-dimensional data, like text.

29. Explain the concept of feature independence assumption in Naïve Bayes.

The feature independence assumption means that Naïve Bayes assumes all features contribute independently to the probability of a class.

P(x1,x2,…,xn y)=P(x1 y) P(x2 y) … P(xn y)

Simplifies computation

Often violated in real-world data, but the model still performs well

30. How does Naïve Bayes handle categorical features with a large number of categories

Challenges:

Sparse data: Many categories may appear rarely

Zero-frequency problem: More chances of categories not being seen in training data

Solutions:

Laplace smoothing to avoid zero probabilities

Category grouping (combine rare categories into "Other")

Use feature hashing to reduce dimensionality

Use embedding-based techniques (in advanced models)

31. What is the curse of dimensionality, and how does it affect machine learning algorithms

The curse of dimensionality refers to the various problems that arise when working with high-dimensional data (many features).

`Effects:`

Data becomes sparse, making patterns harder to detect.

Distance metrics become less meaningful (important in KNN, clustering).

Increases computation time and risk of overfitting.

`Solution: Use dimensionality reduction (e.g., PCA, feature selection).`

32. Explain the bias-variance tradeoff and its implications for machine learning models

Component	Description	Effect
Bias	Error due to overly **simple assumptions** in the model	Leads to **underfitting**
Variance	Error due to model's **sensitivity to fluctuations** in the training data	Leads to **overfitting**

`Tradeoff:`

Low bias → high variance

Low variance → high bias

`Goal: Find the right balance for good generalization on unseen data`

33. What is cross-validation, and why is it used

Cross-validation is a technique for evaluating model performance by dividing the dataset into training and validation sets multiple times.

`Common type: K-Fold Cross-Validation`

Split data into k folds

Train on (k-1) folds, test on 1 fold

Repeat for all folds and average results

`Prevents overfitting and ensures model generalization`

34. Explain the difference between parametric and non-parametric machine learning algorithms

Aspect	Parametric Algorithms	Non-Parametric Algorithms
Assumptions	Assumes a **fixed form** for the model	No assumption about data distribution
Parameters	**Finite** number of parameters	Parameters grow with data

Aspect	Parametric Algorithms	Non-Parametric Algorithms
Examples	Linear regression, Logistic regression, SVM	KNN, Decision Trees, Random Forest
Flexibility	Less flexible	More flexible, captures complex patterns

35. What is feature scaling, and why is it important in machine learning

Feature scaling is the process of normalizing or standardizing features to bring them to a common scale.

`Importance:`

Required for models that use distance (e.g., KNN, SVM, K-Means) or gradient descent.

Prevents dominance of high-scale features.

Helps models converge faster.

`Common methods: Min-Max Scaling, Standardization, Normalization`

36. What is regularization, and why is it used in machine learning

Regularization adds a penalty to the loss function to discourage complex models, helping prevent overfitting.

`Types:`

L1 Regularization (Lasso): Adds absolute values of coefficients → leads to feature selection

L2 Regularization (Ridge): Adds squared values of coefficients → shrinks coefficients but keeps all features

`Encourages simpler, more generalizable models`

37. Explain the concept of ensemble learning and give an example

Ensemble learning combines multiple models (weak learners) to produce better performance than individual models.

`Improves accuracy, robustness, and stability`

`Example:`

Random Forest: Ensemble of decision trees using bagging

Gradient Boosting: Ensemble of weak learners trained sequentially

38. What is the difference between bagging and boosting

Feature	Bagging	Boosting
Model Training	Trains models **in parallel**	Trains models **sequentially**
Goal	Reduce **variance**	Reduce **bias and variance**

Feature	Bagging	Boosting
Example	Random Forest	AdaBoost, XGBoost, Gradient Boosting
Data Sampling	Bootstrap sampling (with replacement)	Weighted sampling based on errors

39. What is the difference between a generative model and a discriminative model

Model Type	Generative Model	Discriminative Model
Learns	**Joint probability** $P(X, Y)$	**Conditional probability** (P(Y X))
Examples	Naïve Bayes, GANs	Logistic Regression, SVM, Decision Trees
Goal	Learn how data is generated	Learn decision boundary between classes
Can generate data?	Yes	No

40. Explain the concept of batch gradient descent and stochastic gradient descent

Batch Gradient Descent:

Uses entire dataset to compute gradients

More stable, but slower and memory-intensive

Stochastic Gradient Descent (SGD):

Uses one data point at a time to update weights

Faster, but has more variance in updates (noisy path to convergence)

Mini-batch Gradient Descent: A compromise → uses a small batch of data points

41. What is the K-nearest neighbors (KNN) algorithm, and how does it work

KNN is a non-parametric, instance-based learning algorithm used for both classification and regression.

How it works:

Choose the number of neighbors k.

Compute distance (e.g., Euclidean) between test point and all training points.

Select the k-nearest points.

For classification: use majority vote.

For regression: use mean/median of neighbors' values.

42. What are the disadvantages of the K-nearest neighbors algorithm

Computationally expensive: Slow during prediction (no training phase)

Sensitive to irrelevant features and feature scaling

Memory intensive: Stores the entire dataset

Performs poorly with high-dimensional data (curse of dimensionality)

No model interpretability

43. Explain the concept of one-hot encoding and its use in machine learning

One-hot encoding converts categorical variables into binary columns, with one column per category.

`Example:`

Color: Red, Blue, Green → [Red, Blue, Green] Red → [1, 0, 0] Blue → [0, 1, 0]

`Why it’s used:`

Converts non-numeric data to numeric

Prevents models from assuming ordinal relationships

44. What is feature selection, and why is it important in machine learning

Feature selection is the process of identifying and using only the most relevant features for training a model.

`Importance:`

Reduces overfitting

Improves accuracy and training speed

Makes models simpler and interpretable

Helps in dealing with high-dimensional data

45. Explain the concept of cross-entropy loss and its use in classification tasks

Cross-entropy loss measures the difference between predicted probabilities and actual class labels.

Formula (binary):

L=−[y log(p)+(1−y) log(1−p)]

`Where:`

y = true label (0 or 1)

p = predicted probability of class 1

`Commonly used in logistic regression, neural networks, and other classification models`

46. What is the difference between batch learning and online learning

Feature	Batch Learning	Online Learning
Data Processing	Uses **entire dataset** at once	Uses **one sample or mini-batch** at a time
Memory Requirement	High	Low
Flexibility	Static – retraining required for new data	Dynamic – updates incrementally
Use Case	Stable datasets	Streaming data, real-time systems

47. Explain the concept of grid search and its use in hyperparameter tuning

Grid search is a brute-force method to find the best combination of hyperparameters by exhaustively trying all combinations in a predefined grid.

Steps:

1. Define ranges/sets of hyperparameters.
2. Train model on each combination using cross-validation.
3. Select the one with best performance.

   Guarantees optimal result (if search space is small)

   Computationally expensive

48. What are the advantages and disadvantages of decision trees

Advantages:

Easy to understand and visualize

No need for feature scaling

Handles both categorical and numerical data

Non-parametric → flexible to data shapes

Disadvantages:

Prone to overfitting (especially deep trees)

Unstable to small changes in data

Biased toward features with more levels

Can be less accurate compared to ensemble methods

49. What is the difference between L1 and L2 regularization

Feature	L1 Regularization (Lasso)	L2 Regularization (Ridge)		
Penalty term	(	w_i	)	$\lambda \sum w_i^2$
Effect	Can shrink some weights to **zero**	Shrinks all weights, none to zero		
Use case	**Feature selection**	Handles multicollinearity well		

L1 → sparse models

L2 → smoother solutions

50. What are some common preprocessing techniques used in machine learning

1. Missing value handling: Imputation (mean, median, mode)
2. Encoding categorical variables: One-hot, label, ordinal encoding
3. Feature scaling: Min-Max, Standardization
4. Outlier detection/removal

5. Dimensionality reduction: PCA, t-SNE
6. Text preprocessing: Tokenization, stemming, TF-IDF
7. Data transformation: Log, sqrt, Box-Cox
8. Balancing classes: SMOTE, undersampling, oversampling

51. What is the difference between a parametric and non-parametric algorithm? Give examples of each

Feature	Parametric Algorithms	Non-Parametric Algorithms
Assumption	Fixed number of parameters	No fixed structure; grows with data
Flexibility	Less flexible, faster	More flexible, slower
Memory usage	Lower	Higher
Examples	Linear Regression, Logistic Regression, Naïve Bayes	KNN, Decision Trees, SVM (with kernel), Random Forest

52. Explain the bias-variance tradeoff and how it relates to model complexity

Bias: Error from simplistic models that can't capture data complexity → leads to underfitting.

Variance: Error from complex models that learn noise in training data → leads to overfitting.

Relation to complexity:

Simple models → high bias, low variance

Complex models → low bias, high variance

Goal: Find the sweet spot where total error (bias^2 + variance + noise) is minimized

53. What are the advantages and disadvantages of using ensemble methods like random forests

Advantages:

Higher accuracy and robustness than individual models

Handles non-linear relationships well

Reduces overfitting (compared to a single decision tree)

Handles missing values and imbalanced data

Disadvantages:

Less interpretable

Requires more computation and memory

Slower in training and prediction than a single tree

54. Explain the difference between bagging and boosting

Aspect	Bagging	Boosting
Model Training	Trains models **in parallel**	Trains models **sequentially**
Focus	Reduces **variance**	Reduces **bias and variance**
Data Sampling	Bootstrap sampling (random with replacement)	Weighted sampling, focuses on errors
Examples	Random Forest	AdaBoost, XGBoost, Gradient Boosting

55. What is the purpose of hyperparameter tuning in machine learning

Hyperparameter tuning is the process of optimizing the external configuration (not learned during training) to improve model performance.

`Examples:`

k in KNN

C, gamma in SVM

learning rate in gradient boosting

`Helps achieve better accuracy, less overfitting, and faster training`

56. What is the difference between regularization and feature selection

Feature	Regularization	Feature Selection
Purpose	**Shrink coefficients** to reduce overfitting	**Select subset of features**
Method	Penalty terms added to loss function	Manual/statistical/model-based selection
Example	L1/L2 regularization	RFE, Chi-square, feature importance
Outcome	Keeps all features but reduces impact	Removes irrelevant/redundant features

57. How does the Lasso (L1) regularization differ from Ridge (L2) regularization?

Feature	Lasso (L1)	Ridge (L2)		
Penalty	(	w_i	)	$\lambda \sum w_i^2$
Effect	Shrinks some weights to **zero**	Shrinks all weights, none to zero		
Feature Selection	**Yes** – leads to sparse models	No – all features retained		
Use Case	When feature selection is needed	When multicollinearity exists		

58. Explain the concept of cross-validation and why it is used

Cross-validation is a technique to assess model generalization by dividing the data into multiple train-test splits.

`K-Fold Cross-Validation:`

Data is split into k folds

Each fold is used once as test, remaining as train

Performance is averaged

`Ensures reliable model evaluation, reduces overfitting risk`

59. What are some common evaluation metrics used for regression tasks

Mean Absolute Error (MAE): Average of absolute differences

Mean Squared Error (MSE): Average of squared differences

Root Mean Squared Error (RMSE): Square root of MSE

R-squared (R^2): Proportion of variance explained

Adjusted R^2: R^2 adjusted for number of predictors

`Helps compare model performance and select the best regressor`

60. How does the K-nearest neighbors (KNN) algorithm make predictions

`For classification:`

Compute distance between query point and all training points.

Select k-nearest neighbors.

Return the most frequent class among neighbors (majority vote).

`For regression:`

Compute distances.

Select k-nearest.

Return the average or median of target values.

`No training phase → predictions are made based on stored training data`

61. What is the curse of dimensionality, and how does it affect machine learning algorithms

The curse of dimensionality refers to problems that arise when data has too many features (dimensions).

`Effects:`

Data sparsity: More dimensions → more space → fewer data points in any region.

Distance metrics lose meaning, affecting models like KNN or clustering.

Overfitting increases, especially in models with high complexity.

`Solution: Apply feature selection or dimensionality reduction techniques (like PCA).`

62. What is feature scaling, and why is it important in machine learning

Feature scaling transforms features so they fall within a similar range.

`Importance:`

Prevents features with large values from dominating distance-based models (KNN, SVM, etc.).

Helps gradient descent converge faster in optimization-based models.

Common methods:

Min-Max Scaling: Scales values between 0 and 1

Standardization: Centers around mean 0 with unit variance

63. How does the Naïve Bayes algorithm handle categorical features

Naïve Bayes handles categorical features using frequency-based probability estimation:

It calculates:

P(feature value class)

Probabilities are learned from the frequency of each feature-class combination in the training data.

Simple and efficient for text, categorical, and discrete data.

64. Explain the concept of prior and posterior probabilities in Naïve Bayes

Prior Probability (P(C)): The probability of a class before observing the feature values.

Posterior Probability (P(C X)): The updated probability of a class after seeing the data.

Bayes' Theorem:

P(C X)= P(X C) P(C)/P(X)

Used to predict the most probable class given the input features.

65. What is Laplace smoothing, and why is it used in Naïve Bayes

Laplace Smoothing (also called add-one smoothing) is used to handle zero probabilities when a feature-category combination doesn't appear in training data.

P(xi y) = count(xi,y)+1/count(y)+k

k = number of possible categories for the feature

Prevents probabilities from becoming zero, which would invalidate the entire product in Naïve

66. Can Naïve Bayes handle continuous features

Yes. Gaussian Naïve Bayes is used for continuous features.

Assumes features follow a normal distribution.

Uses the probability density function (PDF):

P(x y)= 1/sq.rt.(2 ^2) e^(-(x-)^2/2 ^2)

Where:

= mean of feature for class y

= standard deviation

67. What are the assumptions of the Naïve Bayes algorithm

1. Feature Independence: All features are conditionally independent given the class.
2. Equal importance: All features contribute equally and independently to the outcome.

3. Class-conditional probability: Each feature follows a known probability distribution (like Gaussian for continuous).

    Though often violated, Naïve Bayes still performs well.

68. How does Naïve Bayes handle missing values

Approaches:

Ignore missing features in the probability computation.

Use probabilistic estimates based on known values.

Apply imputation techniques (mean, mode, etc.) before training or prediction.

Naïve Bayes is relatively robust to missing data.

69. What are some common applications of Naïve Bayes

Spam filtering

Sentiment analysis

Document classification

Medical diagnosis

Recommendation systems

Credit scoring

Especially effective for text-based and high-dimensional data.

70. Explain the difference between generative and discriminative models

Feature	Generative Model	Discriminative Model
Learns	Joint probability $P(X, Y)$	Conditional probability (P(Y X))
Goal	How data is generated	How to separate/classify the data
Examples	Naïve Bayes, Hidden Markov Models, GANs	Logistic Regression, SVM, Decision Trees
Can generate data?	Yes	No
Use case	Modeling full distribution or creating samples	Predicting labels accurately

71. How does the decision boundary of a Naïve Bayes classifier look like for binary classification tasks

In binary classification, the decision boundary of Naïve Bayes is typically:

Linear when using Gaussian Naïve Bayes with normally distributed features.

Non-linear if distributions are non-Gaussian or features are not independent.

The decision boundary is defined by where the posterior probabilities of both classes are equal:

P(C1 X)=P(C2 X)

72. What is the difference between multinomial Naïve Bayes and Gaussian Naïve Bayes

Feature	Multinomial Naïve Bayes	Gaussian Naïve Bayes
Used For	**Discrete/counted features** (e.g., word counts)	**Continuous features** (e.g., measurements)
Assumes	Features follow **multinomial distribution**	Features follow **normal distribution**
Example Use Cases	Text classification, spam filtering	Iris dataset, medical diagnosis

73. How does Naïve Bayes handle numerical instability issues

Naïve Bayes uses logarithmic probabilities to avoid underflow caused by multiplying many small probabilities:

Instead of:

P(C X)=P(x1 C) P(x2 C) … P(xn C)

It computes:

logP(C X)=logP(C)+ logP(xi C)

Prevents numerical underflow and allows better handling of small probabilities.

74. What is the Laplacian correction, and when is it used in Naïve Bayes

Laplacian correction (Laplace smoothing) is used to:

Avoid zero probabilities for unseen feature-class combinations.

Add a small constant (typically 1) to frequency counts:

P(xi y)= count(xi,y)+1/count(y)+k

Used when dealing with categorical or sparse data, such as in text classification.

75. Can Naïve Bayes be used for regression tasks

Yes, but rarely. There's a variant called Naïve Bayes Regression, where:

The target variable is continuous.

Conditional distributions are assumed (e.g., Gaussian).

Not commonly used in practice because standard regression methods (linear regression, decision

76. Explain the concept of conditional independence assumption in Naïve Bayes

The conditional independence assumption means:

P(x1,x2,...,xn C) = i=1 n P(xi C)

That is, all features are assumed to be independent of each other given the class label.

This simplifies computation significantly

Often violated in practice, but Naïve Bayes still performs well.

77. What are some drawbacks of the Naïve Bayes algorithm

Strong independence assumption rarely holds in real-world data

Performs poorly when features are highly correlated

Cannot learn interactions between features

Not ideal for small datasets with high feature cardinality

Sensitive to imbalanced data without prior adjustment

78. Explain the concept of smoothing in Naïve Bayes

Smoothing refers to techniques like Laplace (add-1) or Lidstone (add-) smoothing that adjust p

Avoid zero probabilities for unseen data

Improve generalization

Handle sparse datasets (common in text data)

Example: Instead of assigning 0 to unseen word frequency, give it a small non-zero value

79. How does Naïve Bayes handle categorical features with a large number of categories

Challenges:

Sparse probabilities

Overfitting on rare categories

Computational overhead

Solutions:

Use Laplace smoothing

Group rare categories as "Other"

Apply feature hashing or embedding

Use dimensionality reduction or frequency-based binning

80. How does Naïve Bayes handle imbalanced datasets?

By default, Naïve Bayes can be biased toward the majority class.

Solutions:

Use class priors to adjust probability toward minority class.

Apply resampling techniques (oversample minority or undersample majority).

Use SMOTE for synthetic sampling.

Adjust decision threshold based on predicted probabilities.