ML_Assignment_2

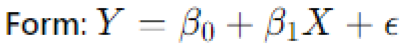
1. What is Regression Analysis?

Regression Analysis is a statistical technique used to model and analyze the relationship between a dependent variable and one or more independent variables. It helps in understanding how the dependent variable changes as the independent variables change, and it can be used for prediction and forecasting.

2. Explain the Difference Between Linear and Nonlinear Regression

Linear Regression:

Description: Models the relationship between variables using a linear function.



Form: $Y = \beta_0 + \beta_1 X + \epsilon$

Example: Predicting house prices based on square footage.

Nonlinear Regression:

Description: Models the relationship using a nonlinear function.

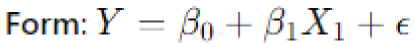
Form: The relationship is described by a nonlinear equation (e.g., exponential, logarithmic).

Example: Modeling population growth with an exponential function.

3. What is the Difference Between Simple Linear Regression and Multiple Linear Regression?

Simple Linear Regression:

Description: Models the relationship between two variables using a linear equation.

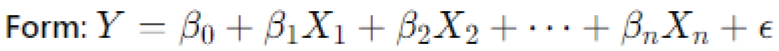


Form: $Y = \beta_0 + \beta_1 X_1 + \epsilon$

Variables: One independent variable.

Multiple Linear Regression:

Description: Models the relationship between one dependent variable and multiple independent variables.



Form: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$

Variables: Multiple independent variables.

4. How is the Performance of a Regression Model Typically Evaluated?

Performance Metrics:

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

Mean Squared Error (MSE): Average of squared differences between predicted and actual values.

Root Mean Squared Error (RMSE): Square root of MSE, providing error in the same units as the dependent variable.

R-squared: Proportion of variance in the dependent variable that is predictable from the independent variables.

5. What is Overfitting in the Context of Regression Models?

Overfitting occurs when a regression model learns the noise or random fluctuations in the training data rather than the underlying pattern. This leads to high accuracy on the training data but poor generalization to new, unseen data.

6. What is Logistic Regression Used For?

Logistic Regression is used for binary classification tasks. It models the probability of a binary outcome based on one or more predictor variables.

7. How Does Logistic Regression Differ from Linear Regression?

Logistic Regression:

Purpose: Used for binary classification.

Output: Probability of the positive class, transformed by the sigmoid function.

Linear Regression:

Purpose: Used for regression tasks.

Output: Continuous value, not bounded between 0 and 1.

8. Explain the Concept of Odds Ratio in Logistic Regression

Odds Ratio is a measure of association between a predictor variable and the outcome in logistic regression. It represents how the odds of the outcome increase or decrease with a one-unit change in the predictor variable.

9. What is the Sigmoid Function in Logistic Regression?

Sigmoid Function is used to map predicted values to probabilities in logistic regression. It is defined as:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

where

z

z is the linear combination of predictors.

10. How is the Performance of a Logistic Regression Model Evaluated?

Performance Metrics:

Accuracy: Proportion of correctly classified instances.

Precision: Proportion of true positives among predicted positives.

Recall: Proportion of true positives among actual positives.

F1 Score: Harmonic mean of precision and recall.

AUC-ROC: Area under the receiver operating characteristic curve, measuring the model's ability to distinguish between classes.

11. What is a Decision Tree?

Decision Tree is a flowchart-like tree structure used for classification and regression tasks. It splits the data into subsets based on feature values, leading to a decision at each node and producing a final output at the leaves.

12. How Does a Decision Tree Make Predictions?

Prediction Process:

Splitting: The tree splits the data based on feature values to form branches.

Decision Nodes: At each node, a decision is made based on a feature's value.

Leaf Nodes: The final decision or prediction is made based on the majority class or average value in the leaf node.

13. What is Entropy in the Context of Decision Trees?

Entropy is a measure of the impurity or disorder in a dataset. In decision trees, it quantifies the uncertainty in the dataset, guiding the splits by choosing features that reduce entropy.

14. What is Pruning in Decision Trees?

Pruning is the process of removing branches from a decision tree that have little importance to improve the model's generalization and reduce overfitting.

15. How Do Decision Trees Handle Missing Values?

Surrogate Splits: Use alternative splits if the primary feature is missing.

Imputation: Fill missing values with the mean or mode of the feature.

Tree Structure Adjustments: Modify the tree to handle missing values during training and prediction.

16. What is a Support Vector Machine (SVM)?

Support Vector Machine (SVM) is a classification algorithm that finds the optimal hyperplane that separates different classes in the feature space with the maximum margin.

17. Explain the Concept of Margin in SVM

Margin is the distance between the hyperplane and the closest data points from each class (support vectors). SVM aims to maximize this margin to ensure better separation between classes.

18. What are Support Vectors in SVM?

Support Vectors are the data points that lie closest to the decision boundary (hyperplane) and are used to define the optimal margin. They are critical for constructing the hyperplane.

19. How Does SVM Handle Non-Linearly Separable Data?

Kernel Trick: SVM uses kernel functions (e.g., polynomial, radial basis function) to transform the feature space into higher dimensions where the data becomes linearly separable.

20. What are the Advantages of SVM Over Other Classification Algorithms?

Effective in High Dimensions: Works well with a large number of features.

Robust to Overfitting: Effective in high-dimensional spaces with a proper choice of kernel and regularization.

Margin Maximization: Provides a clear margin of separation.

21. What is the Naïve Bayes Algorithm?

Naïve Bayes is a probabilistic classification algorithm based on Bayes' theorem with the assumption of feature independence. It predicts the class based on the probability of features given the class.

22. Why is it Called "Naïve" Bayes?

It is called "Naïve" Bayes because it assumes that all features are independent of each other given the class label, which is a simplification that often does not hold in practice.

23. How Does Naïve Bayes Handle Continuous and Categorical Features?

Categorical Features: Uses frequency counts to calculate probabilities.

Continuous Features: Often assumes a Gaussian distribution and uses the mean and variance to estimate probabilities.

24. Explain the Concept of Prior and Posterior Probabilities in Naïve Bayes

Prior Probability: Probability of each class before considering the features.

Posterior Probability: Probability of a class given the features, computed using Bayes' theorem.

25. What is Laplace Smoothing and Why is it Used in Naïve Bayes?

Laplace Smoothing (or add-one smoothing) adjusts probabilities to avoid zero probabilities for unseen features by adding a small constant to frequency counts. It is used to handle cases where some feature-class combinations are not present in the training data.

26. Can Naïve Bayes be Used for Regression Tasks?

No, Naïve Bayes is primarily used for classification tasks. For regression tasks, other algorithms like linear regression or support vector regression are more appropriate.

27. How Do You Handle Missing Values in Naïve Bayes?

Imputation: Replace missing values with the mean or mode of the feature.

Ignore: Exclude instances with missing values from training.

Feature Engineering: Treat missing values as a separate category.

28. What are Some Common Applications of Naïve Bayes?

Text Classification: Spam detection, sentiment analysis.

Document Classification: Categorizing documents into topics.

Medical Diagnosis: Predicting disease presence based on symptoms.

29. Explain the Concept of Feature Independence Assumption in Naïve Bayes

Feature Independence Assumption assumes that all features are independent of each other given the class label. This simplifies computation but may not always hold in practice.

30. How Does Naïve Bayes Handle Categorical Features with a Large Number of Categories?

Naïve Bayes can handle large numbers of categories by using frequency counts or probability distributions. For high-cardinality categorical features, techniques like feature hashing or embedding may be used to reduce dimensionality.

31. What is the Curse of Dimensionality, and How Does it Affect Machine Learning Algorithms?

Curse of Dimensionality refers to the problems that arise when working with high-dimensional data, including increased computational complexity and sparsity of data. It can lead to overfitting and reduced model performance.

32. Explain the Bias-Variance Tradeoff and Its Implications for Machine Learning Models

Bias: Error due to overly simplistic models that cannot capture the underlying patterns (underfitting).

Variance: Error due to overly complex models that fit noise in the training data (overfitting).

Tradeoff: Balancing bias and variance is crucial to achieving a model that generalizes well to new data.

33. What is Cross-Validation, and Why is it Used?

Cross-Validation is a technique for evaluating model performance by dividing the data into multiple folds, training on some folds, and validating on others. It helps assess the model’s ability to generalize to new data and reduces overfitting.

34. Explain the Difference Between Parametric and Non-Parametric Machine Learning Algorithms

Parametric Algorithms: Assume a specific form for the model and have a fixed number of parameters (e.g., linear regression).

Non-Parametric Algorithms: Do not assume a specific form and can have an infinite number of parameters (e.g., k-nearest neighbours, decision trees).

35. What is Feature Scaling, and Why is it Important in Machine Learning?

Feature Scaling involves normalizing or standardizing feature values so they are on a similar scale. It is important because many machine learning algorithms perform better or converge faster with scaled features.

36. What is Regularization, and Why is it Used in Machine Learning?

Regularization is a technique to prevent overfitting by adding a penalty to the loss function for large coefficients. It helps improve model generalization by controlling model complexity.

37. Explain the Concept of Ensemble Learning and Give an Example

Ensemble Learning combines predictions from multiple models to improve overall performance. Examples include:

Random Forests: Combines multiple decision trees.

Gradient Boosting Machines: Sequentially adds models to correct errors of previous models.

38. What is the Difference Between Bagging and Boosting?

Bagging: Builds multiple models independently and combines their predictions (e.g., Random Forest).

Boosting: Builds models sequentially, where each new model corrects errors of the previous ones (e.g., Gradient Boosting, AdaBoost).

39. What is the Difference Between a Generative Model and a Discriminative Model?

Generative Model: Models the joint probability $P(X,Y)$and can generate new data points (e.g., Naïve Bayes, Gaussian Mixture Models).

Discriminative Model: Models the conditional probability $P(Y|X)$ and focuses on distinguishing between classes (e.g., Logistic Regression, SVM).

40. Explain the Concept of Batch Gradient Descent and Stochastic Gradient Descent

Batch Gradient Descent: Updates model parameters using the average gradient of the entire training dataset.

Stochastic Gradient Descent (SGD): Updates model parameters using the gradient of a single training example, which can be more computationally efficient but noisier.

41. What is the K-Nearest Neighbors (KNN) Algorithm, and How Does it Work?

K-Nearest Neighbors (KNN) is a classification algorithm that assigns a class to a data point based on the majority class among its k-nearest neighbors in the feature space. It works by calculating distances between points and voting based on the nearest neighbors.

42. What are the Disadvantages of the K-Nearest Neighbors Algorithm?

Computational Complexity: Can be slow with large datasets due to distance calculations.

Memory Usage: Requires storing the entire training dataset.

Sensitive to Noisy Data: Performance can degrade with noisy features or outliers.

43. Explain the Concept of One-Hot Encoding and Its Use in Machine Learning

One-Hot Encoding converts categorical variables into binary vectors. Each category is represented by a binary vector with a single "1" in the position corresponding to the category and "0"s elsewhere. It is used to handle categorical data in algorithms that require numerical input.

44. What is Feature Selection, and Why is it Important in Machine Learning?

Feature Selection involves choosing a subset of relevant features from the original set. It is important because it can improve model performance, reduce overfitting, and decrease computational cost.

45. Explain the Concept of Cross-Entropy Loss and Its Use in Classification Tasks

Cross-Entropy Loss measures the difference between the true class labels and the predicted probabilities. It is used in classification tasks to penalize incorrect predictions, especially in binary and multi-class classification problems.

46. What is the Difference Between Batch Learning and Online Learning?

Batch Learning: Trains the model using the entire dataset at once. It is suitable for smaller datasets but may be impractical for large, streaming data.

Online Learning: Trains the model incrementally with each new data point or batch. It is suitable for large or streaming datasets and adapts to new data over time.

47. Explain the Concept of Grid Search and Its Use in Hyperparameter Tuning

Grid Search is a hyperparameter tuning technique that exhaustively searches over a predefined set of hyperparameters to find the best combination. It evaluates all possible combinations and selects the one with the best performance.

48. What are the Advantages and Disadvantages of Decision Trees?

Advantages:

Interpretability: Easy to understand and visualize.

No Need for Feature Scaling: Handles different types of features without scaling.

Disadvantages:

Overfitting: Can become too complex and overfit the training data.

Instability: Small changes in the data can lead to different tree structures.

49. What is the Difference Between L1 and L2 Regularization?

L1 Regularization (Lasso): Adds the absolute value of the coefficients to the loss function. It can lead to sparse models by setting some coefficients to zero.

L2 Regularization (Ridge): Adds the squared value of the coefficients to the loss function. It discourages large coefficients but does not lead to sparsity.

50. What are Some Common Preprocessing Techniques Used in Machine Learning?

Normalization: Scaling features to a range, e.g., [0,1].

Standardization: Scaling features to have zero mean and unit variance.

Imputation: Filling missing values with mean, median, or mode.

Encoding: Converting categorical variables into numerical formats.

51. What is the Difference Between a Parametric and Non-Parametric Algorithm? Give Examples of Each

Parametric Algorithm:

Definition: Assumes a specific form for the model with a fixed number of parameters.

Examples: Linear Regression, Logistic Regression.

Non-Parametric Algorithm:

Definition: Does not assume a specific form and can have an infinite number of parameters.

Examples: K-Nearest Neighbors, Decision Trees.

52. Explain the Bias-Variance Tradeoff and How it Relates to Model Complexity

Bias-Variance Tradeoff:

Bias: Error due to oversimplified models that miss important patterns (high bias, low complexity).

Variance: Error due to models that capture noise or fluctuations in the training data (high variance, high complexity).

Balancing bias and variance involves choosing a model complexity that minimizes total error.

53. What are the Advantages and Disadvantages of Using Ensemble Methods Like Random Forests?

Advantages:

Improved Accuracy: Combines multiple models to improve prediction accuracy.

Reduced Overfitting: Aggregates predictions from various models to generalize better.

Disadvantages:

Complexity: Can be computationally intensive and harder to interpret.

Training Time: Requires more time and resources for training compared to individual models.

54. Explain the Difference Between Bagging and Boosting

Bagging: Combines predictions from multiple models built independently using bootstrap samples. It reduces variance and helps prevent overfitting (e.g., Random Forest).

Boosting: Builds models sequentially, where each new model corrects errors of the previous ones. It reduces bias and improves model performance (e.g., Gradient Boosting).

55. What is the Purpose of Hyperparameter Tuning in Machine Learning?

Hyperparameter Tuning aims to find the best set of hyperparameters for a machine learning model to improve its performance. It involves adjusting settings that control the learning process and model complexity.

56. What is the Difference Between Regularization and Feature Selection?

Regularization: Adds a penalty to the loss function to discourage large coefficients and prevent overfitting. It is applied during model training.

Feature Selection: Involves selecting a subset of relevant features from the original set to improve model performance and reduce overfitting.

57. How Does the Lasso (L1) Regularization Differ from Ridge (L2) Regularization?

Lasso (L1) Regularization: Adds the absolute value of coefficients to the loss function, which can lead to sparse models with some coefficients being zero.

Ridge (L2) Regularization: Adds the squared value of coefficients to the loss function, which discourages large coefficients but does not lead to sparsity.

58. Explain the Concept of Cross-Validation and Why it is Used

Cross-Validation involves partitioning the dataset into multiple subsets (folds), training the model on some folds, and validating it on the remaining folds. It is used to assess model performance and ensure that the model generalizes well to new data.

59. What are Some Common Evaluation Metrics Used for Regression Tasks?

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

Mean Squared Error (MSE): Average of squared differences between predicted and actual values.

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

R-squared: Proportion of variance in the dependent variable that is predictable from the independent variables.

60. How Does the K-Nearest Neighbors (KNN) Algorithm Make Predictions?

K-Nearest Neighbors (KNN) makes predictions by identifying the k-nearest data points to the test instance and assigning the majority class (for classification) or averaging the values (for regression) among those neighbors.

61. What is the Curse of Dimensionality, and How Does it Affect Machine Learning Algorithms?

Curse of Dimensionality refers to problems arising from high-dimensional data, including sparsity, increased computational complexity, and difficulty in visualizing and understanding data. It can lead to overfitting and reduced model performance.

62. What is Feature Scaling, and Why is it Important in Machine Learning?

Feature Scaling involves normalizing or standardizing features so they are on a similar scale. It is important because many machine learning algorithms rely on distances or assume features are on a similar scale, improving convergence and performance.

63. How Does the Naïve Bayes Algorithm Handle Categorical Features?

Naïve Bayes handles categorical features by calculating the probability of each category given the class label using frequency counts or probabilities.

64. Explain the Concept of Prior and Posterior Probabilities in Naïve Bayes

Prior Probability: Probability of each class before considering the features.

Posterior Probability: Probability of a class given the features, calculated using Bayes' theorem.

65. What is Laplace Smoothing, and Why is it Used in Naïve Bayes?

Laplace Smoothing adds a small constant to frequency counts to handle cases where some feature-class combinations are not present in the training data. It prevents zero probabilities and ensures smooth probability estimates.

66. Can Naïve Bayes Handle Continuous Features?

Yes, Naïve Bayes can handle continuous features by assuming a probability distribution (e.g., Gaussian) and estimating the distribution parameters (mean and variance).

67. What are the Assumptions of the Naïve Bayes Algorithm?

Feature Independence Assumption: Features are independent of each other given the class label.

Probability Distribution: Assumes a specific distribution for continuous features, such as Gaussian.

68. How Does Naïve Bayes Handle Missing Values?

Imputation: Replace missing values with the mean, median, or mode of the feature.

Ignore: Exclude instances with missing values from training.

Separate Category: Treat missing values as a distinct category.

69. What are Some Common Applications of Naïve Bayes?

Text Classification: Spam detection, sentiment analysis.

Document Classification: Categorizing documents into topics.

Medical Diagnosis: Predicting disease presence based on symptoms.

70. Explain the Difference Between Generative and Discriminative Models

Generative Models: Model the joint probability $P(X,Y)$ and can generate new data points (e.g., Naïve Bayes, Gaussian Mixture Models).

Discriminative Models: Model the conditional probability $P(Y|X)$ and focus on distinguishing between classes (e.g., Logistic Regression, SVM).

71. How Does the Decision Boundary of a Naïve Bayes Classifier Look Like for Binary Classification Tasks?

Decision Boundary: For binary classification, the decision boundary is often nonlinear and determined by the level curves of the posterior probability. It depends on the assumed distribution of the features and class probabilities.

72. What is the Difference Between Multinomial Naïve Bayes and Gaussian Naïve Bayes?

Multinomial Naïve Bayes: Assumes feature values are counts or frequencies (e.g., text classification).

Gaussian Naïve Bayes: Assumes features follow a Gaussian distribution (e.g., continuous data).

73. How Does Naïve Bayes Handle Numerical Instability Issues?

Numerical Instability: To handle numerical instability, Naïve Bayes uses techniques like log-transforming probabilities and applying Laplace smoothing to prevent zero probabilities.

74. What is the Laplacian Correction, and When is it Used in Naïve Bayes?

Laplacian Correction (or Laplace Smoothing) is used to adjust probability estimates by adding a small constant to frequency counts, preventing zero probabilities for unseen feature-class combinations.

75. Can Naïve Bayes Be Used for Regression Tasks?

No, Naïve Bayes is designed for classification tasks. For regression, other algorithms like linear regression or support vector regression are more appropriate.

76. Explain the Concept of Conditional Independence Assumption in Naïve Bayes

Conditional Independence Assumption: Assumes that features are independent of each other given the class label. This simplifies probability calculations but may not always hold in practice.

77. How Does Naïve Bayes Handle Categorical Features with a Large Number of Categories?

Naïve Bayes handles high-cardinality categorical features by calculating probabilities for each category or using techniques like feature hashing to manage large numbers of categories efficiently.

78. What are Some Drawbacks of the Naïve Bayes Algorithm?

Feature Independence Assumption: The assumption of feature independence is often unrealistic and can affect performance.

Limited to Probabilistic Models: Not suitable for complex relationships between features and classes.

79. Explain the Concept of Smoothing in Naïve Bayes

Smoothing involves adjusting probability estimates to handle cases where some feature-class combinations are not present in the training data. It ensures that probabilities are not zero and are more robust.

80. How Does Naïve Bayes Handle Imbalanced Datasets?

Adjust Class Weights: Use class weights to compensate for imbalanced classes.

Resampling: Use techniques like oversampling the minority class or under sampling the majority class.

Performance Metrics: Use metrics like precision, recall, and F1-score that are more informative in imbalanced scenarios.