Test : Machine Learning Algorithms (Theory)

Linear Regression

1.What is the difference between simple linear regression and multiple linear regression?

Simple linear regression and multiple linear regression are both techniques used to model the relationship between one dependent variable and one or more independent variables. The main difference between the two lies in the number of independent variables involved:

Simple Linear Regression

- Simple linear regression involves only one independent variable and one dependent variable.
- The relationship between the independent variable and the dependent variable is modelled as a straight line.

The equation for simple linear regression is of the form:

$$y = mx + b$$

where y is the dependent variable, x is the independent variable, m is the slope of the line, and b is the intercept.

Multiple Linear Regression

- Multiple linear regression involves two or more independent variables and one dependent variable.
- The relationship between the dependent variable and multiple independent variables is modelled as a linear combination of the independent variables.

The equation for multiple linear regression is of the form:

$$y = b0 + b1x1 + b2x2 + ... + bnxn$$

where y is the dependent variable, x1, x2, ..., xn are the independent variables, and b0, b1, b2, ..., bn are the coefficients representing the impact of each independent variable on the dependent variable.

2.Explain the concept of the cost function in linear regression.

The cost function in linear regression is a function that measures the difference between the predicted values by the model and the actual values in the dataset. The goal of linear regression is to minimize this cost function, typically using a method like least squares, to find the best-fitting line that describes the relationship between the independent variables and the dependent variable.

In linear regression, the cost function is often defined as the **Mean Squared Error (MSE)**. For a simple linear regression model with one independent variable, the cost function can be expressed as:

$MSE = (1/2n) * \Sigma (yi - \hat{y}i) ^2$

Where,

n: Number of observations
yi: Actual value of the dependent variable for observation i
$\hat{y}i$: Predicted value of the dependent variable for observation i
Σ: Summation symbol (summing over all observations)

3.How do you interpret the coefficients in a linear regression model?

In a linear regression model, the coefficients represent the weights assigned to each independent variable in the model. These coefficients indicate how much the dependent variable is expected to change when the corresponding independent variable changes by one unit, assuming all other variables remain constant. So, interpreting the coefficients involves understanding the impact of each independent variable on the dependent variable.

4.What are the assumptions of linear regression?

- Linearity: The relationship between the independent and dependent variables is linear.
- Independence: The observations in the dataset are independent of each other.
- Homoscedasticity: The variance of the residuals (the differences between the observed and predicted values) is constant across all levels of the independent variables.
- Normality: The residuals are normally distributed.
- No multicollinearity: The independent variables are not highly correlated with each other.

Logistic Regression

1.How does logistic regression differ from linear regression?

- Linear regression is used for predicting continuous numerical values, while logistic regression is used for predicting binary outcomes (0 or 1).
- In linear regression, the output is a continuous value based on a linear relationship between the independent variables and the dependent variable. In logistic regression, the output is a probability score between 0 and 1, which is then transformed into a binary outcome using a threshold.
- Logistic regression uses a different cost function (log loss) and activation function (sigmoid function) compared to linear regression.

2.Explain the sigmoid function and its role in logistic regression.

The sigmoid function, also known as the logistic function, is a mathematical function that maps any real value into a value between 0 and 1. It is defined as:

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

where z is the input to the function.

In logistic regression, the sigmoid function is used to transform the output of the linear combination of the features into a probability score between 0 and 1. This probability score represents the likelihood that a given input belongs to a certain class (0 or 1).

3.What are the key performance metrics used to evaluate a logistic regression model?

- Accuracy: The proportion of correctly classified instances.
- Precision: The ratio of true positive predictions to the total number of positive predictions.
- Recall (Sensitivity): The ratio of true positive predictions to the total number of actual positive instances.
- F1 Score: The harmonic mean of precision and recall, providing a balance between the two metrics.
- ROC-AUC: Receiver Operating Characteristic - Area Under the Curve, which measures the model's ability to distinguish between classes.

4.How do you handle multicollinearity in logistic regression?

Multicollinearity is a statistical phenomenon in which two or more independent variables in a regression model are highly correlated with each other. To address multicollinearity, some strategies include:

- Remove one of the correlated variables.
- Use regularization techniques like Lasso or Ridge regression to penalize large coefficients.
- Perform dimensionality reduction techniques like Principal Component Analysis (PCA).
- Use Variance Inflation Factor (VIF) analysis to detect and address multicollinearity.
- Collect more data, if possible, to reduce the impact of multicollinearity.

Naive Bayes

1.What is the Naive Bayes algorithm based on?

The Naive Bayes algorithm is based on Bayes' theorem that describes the probability of an event based on prior knowledge or information. Bayes' Theorem is expressed as:

$P(A|B) = P(B) / \{P(B|A) * P(A)\}$

Where:

P(A|B) is the probability of event A occurring given that event B has occurred.

P(B|A) is the probability of event B occurring given that event A has occurred.

P(A) is the prior probability of event A.

P(B) is the prior probability of event B.

Naïve Bayes is a classification technique that is based on the assumption of independence between the features. Despite its simplicity, Naive Bayes is known for its effectiveness in text classification and other classification tasks.

2.Explain the concept of conditional probability in the context of Naive Bayes.

In the context of Naive Bayes, conditional probability refers to the probability of an event occurring given that another event has already occurred. In the Naive Bayes algorithm, conditional probability is used to calculate the probability of a class label given the values of the features. It assumes that the features are conditionally independent given the class label, which is where the "naive" assumption comes from.

3.What are the advantages and disadvantages of Naive Bayes?

Advantages of Naive Bayes:

- Simple and easy to implement.
- Efficient and fast for both training and prediction.
- Works well with high-dimensional data, such as text classification.
- Handles both numerical and categorical data.
- Robust to irrelevant features.

Disadvantages of Naive Bayes:

- Naive Bayes assumes independence between features, which may not hold true in real-world datasets.
- It can be overly simplistic and may not capture complex relationships in the data.
- Sensitive to the presence of irrelevant features.
- Requires a large amount of data to make accurate predictions.

4.How does Naive Bayes handle missing values and categorical features?

Handling missing values and categorical features in Naive Bayes:

- Naive Bayes can handle missing values by either ignoring instances with missing values during training or by imputing missing values with a suitable method (e.g., mean, median, mode).

- Naive Bayes can handle categorical features by converting them into numerical values through techniques like one-hot encoding or label encoding. This allows the algorithm to work with categorical data effectively.

Decision Trees

1.How does a decision tree make decisions?

A decision tree makes decisions by recursively partitioning the input space based on features, leading to splits at each node until reaching a leaf node for a decision or prediction. The process follows a flowchart-like structure and continues until a stopping criterion like maximum depth or purity level is met.

2.What are the main criteria for splitting nodes in a decision tree?

The main criteria for splitting nodes in a decision tree include measures like Gini Impurity, Entropy, and Information Gain. These criteria are used to determine the best feature and value to split the data at each node, aiming to maximize the homogeneity of the resulting child nodes.

- **Gini Impurity:** Measures the impurity or disorder in a node. It is minimized when the classes are perfectly mixed.
- **Entropy:** Measures the level of impurity or disorder in a set of examples. It is minimized when the classes are perfectly separated.
- **Information Gain:** Measures the reduction in entropy or impurity achieved by splitting a node based on a particular feature.

3.How do decision trees handle categorical variables?

Decision trees can handle categorical variables by employing techniques like one-hot encoding or label encoding to convert categorical variables into numerical values that the algorithm can work with effectively. Each category is represented as a binary (one-hot encoding) or ordinal (label encoding) variable.

4.What are some common techniques to prevent overfitting in decision trees?

- **Pruning:** Pruning involves cutting back parts of the tree that do not provide additional predictive power. Pre-pruning involves setting stopping criteria before building the tree, while post-pruning involves removing nodes from an already fully grown tree.

- **Limiting Tree Depth:** Restricting the maximum depth of the tree can prevent it from becoming too complex and overfitting to the training data.

- **Minimum Samples per Leaf:** Setting a minimum number of samples required to be at a leaf node can prevent the model from creating nodes with very few samples.

- **Feature Selection:** Limiting the number of features used in splitting nodes can help prevent overfitting by focusing on the most informative features.

- **Ensemble Methods:** Using ensemble methods like Random Forests or Gradient Boosting can help reduce overfitting by combining multiple decision trees and improving generalization.

Support Vector Machines (SVM)

1.What is the basic idea behind SVM?

SVM is a supervised machine learning algorithm used for classification and regression tasks. The basic idea behind SVM is to find the hyperplane that best separates the data into different classes while maximizing the margin between classes. SVM aims to find the optimal decision boundary that not only separates the classes but also generalizes well to unseen data.

2.Explain the concepts of margin and support vectors in SVM.

- **Margin:** The margin in SVM is the distance between the hyperplane (decision boundary) and the nearest data points from each class. The goal of SVM is to maximize this margin, as a larger margin indicates better generalization and robustness of the model.

- **Support Vectors:** Support vectors are the data points that lie closest to the decision boundary (hyperplane). These points play a crucial role in defining the decision boundary and determining the margin. They are the critical instances that support the definition of the hyperplane.

3.What are the different kernel functions used in SVM, and when would you use each?

In SVM (Support Vector Machine), different kernel functions are used to map the input data into a higher-dimensional space where it becomes easier to find a separating hyperplane. Some common kernel functions used in SVM are:

- **Linear Kernel:** The linear kernel is used for linearly separable data or when the number of features is large. It computes the dot product between two data points in the original feature space.

- **Polynomial Kernel:** The polynomial kernel function is used when the data is not linearly separable and can map the data into higher-dimensional space using a polynomial function.

- **RBF (Radial Basis Function) Kernel**: The RBF kernel, also known as the Gaussian kernel, is a popular choice in SVM. RBF kernel function maps the data into an infinite-dimensional space using a Gaussian function. The RBF kernel is versatile

and suitable when there is no prior knowledge about the data distribution. It can handle non-linear separable data effectively.

- **Sigmoid Kernel:** The sigmoid kernel maps the data into a higher-dimensional space using a hyperbolic tangent function. It can be used for non-linear decision boundaries and in scenarios where the data is not linearly separable.

4.How does SVM handle outliers?

SVM is sensitive to outliers as it aims to maximize the margin and find the best separation between classes. Outliers can significantly impact the position of the hyperplane.

To handle outliers in SVM, techniques such as using soft-margin SVM (allowing for some misclassification), adjusting the regularization parameter C to control the influence of outliers, or preprocessing data to detect and remove outliers can be employed.

Outlier detection methods like Isolation Forest or One-Class SVM can also be used before training an SVM model to mitigate the impact of outliers on the model's performance.
