WORKSHEET SET 2

STATISTICS WORKSHEET-5

Q1 to Q10 are MCQs with only one correct answer. Choose the correct option.

1. Using a goodness of fit, we can assess whether a set of obtained frequencies differ from a set of frequencies.

a) Mean

b) Actual

c) Predicted

d) Expected

Answer: d) Expected

2. Chi-square is used to analyse

a) Score

b) Rank

c) Frequencies

d) All of these

Answer: c) Frequencies

3. What is the mean of a Chi Square distribution with 6 degrees of freedom?

a) 4

b) 12

c) 6

d) 8

Answer: c) 6

4. Which of these distributions is used for a goodness of fit testing?

a) Normal distribution

b) Chi-squared distribution

c) Gamma distribution

d) Poisson distribution

Answer: b) Chi-squared distribution

5. Which of the following distributions is Continuous

a) Binomial Distribution

b) Hypergeometric Distribution

c) F Distribution

d) Poisson Distribution

Answer: c) F Distribution

6. A statement made about a population for testing purpose is called?

a) Statistic

b) Hypothesis

c) Level of Significance

d) Test Statistic

Answer: b) Hypothesis

7. If the assumed hypothesis is tested for rejection considering it to be true is called?

a) Null Hypothesis

b) Statistical Hypothesis

c) Simple Hypothesis

d) Composite Hypothesis

Answer: a) Null Hypothesis

8. If the Critical region is evenly distributed then the test is referred as?

a) Two tailed

b) One tailed

c) Three tailed

d) Zero tailed

Answer: a) Two tailed

9. Alternative Hypothesis is also called as?

a) Composite hypothesis

b) Research Hypothesis

c) Simple Hypothesis

d) Null Hypothesis

Answer: b) Research Hypothesis

10. In a Binomial Distribution, if ‘n’ is the number of trials and ‘p’ is the probability of success, then the mean value is given by

a) np

b) n

Answer: a) np

MACHINE LEARNING

ASSIGNMENT - 5

Q1 to Q15 are subjective answer type questions, Answer them briefly.

1. R-squared or Residual Sum of Squares (RSS) which one of these two is a better measure of goodness of fit model in regression and why?

Answer: R-squared (R^2) is generally considered a better measure of goodness of fit in regression compared to Residual Sum of Squares (RSS). Here's why; R-squared is preferred over Residual Sum of Squares (RSS) as it not only measures goodness of fit but also provides interpretability, standardization, and a relative comparison to baseline models, making it a more comprehensive measure in regression analysis and decision making.

2. What are TSS (Total Sum of Squares), ESS (Explained Sum of Squares) and RSS (Residual Sum of Squares) in regression. Also mention the equation relating these three metrics with each other.

Answer: In regression analysis, TSS (Total Sum of Squares), ESS (Explained Sum of Squares), and RSS (Residual Sum of Squares) are important metrics used to assess the goodness of fit of a regression model. Here's an explanation of each term and the equation relating them:

Total Sum of Squares (TSS): TSS represents the total variability in the dependent variable (Y) around its mean ($\bar{Y}$). It measures the total dispersion of observed data points around the mean without considering the regression model.

Explained Sum of Squares (ESS): ESS represents the variability in the dependent variable that is explained by the regression model. It quantifies how much of the total variation in the dependent variable is accounted for by the independent variables in the model.

Residual Sum of Squares (RSS): RSS represents the variability in the dependent variable that is not explained by the regression model, often referred to as the residuals or errors of the model.

Relationship Equation: The relationship between TSS, ESS, and RSS can be expressed using the following equation: TSS = ESS + RSS

This equation shows that the total variation in the dependent variable (TSS) can be decomposed into two parts: the variation explained by the regression model (ESS) and the unexplained variation (RSS).

3. What is the need of regularization in machine learning?

Answer: Regularization is a technique used in machine learning to prevent overfitting, improve the generalizability of models, and enhance their performance on unseen data. The key reasons for the need of regularization in machine learning are stated below:

Regularization is crucial for building machine learning models that are not only accurate but also generalizable to new, unseen data. By controlling model complexity, preventing overfitting, avoiding multicollinearity, enhancing generalization, and enhancing feature selection, regularization techniques ensure that models are both effective and reliable in practical applications.

4. What is Gini–impurity index?

Answer: Gini-impurity index is a metric used to measure the impurity or diversity of a dataset. It is commonly used in decision trees for classification tasks to determine the best split at each node. Gini-impurity index is also said to be a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. It helps in making optimal decisions at each node in the construction of decision trees.

5. Are unregularized decision-trees prone to overfitting? If yes, why?

Answer: Yes, unregularized decision trees are prone to overfitting. Here's why;

Complexity and Depth: Unregularized decision trees can grow very deep, with many branches and nodes. They can perfectly fit the training data by creating very specific rules for each training example, capturing noise and outliers in the process.

High Variance: By fitting the training data too closely, unregularized decision trees can exhibit high variance. This means that small changes in the training data can result in large changes in the structure of the tree, leading to poor generalization to new, unseen data.

Lack of Constraints: Without regularization, there are no constraints on the growth of the tree (e.g., no limit on the depth, number of leaves, or minimum samples per leaf). This allows the tree to keep splitting until each leaf node contains only one data point, which results in a model that is overly tailored to the training data.

Noise Fitting: Decision trees without regularization can fit the random noise present in the training data, leading to an overly complex model that does not perform well on test data.

6. What is an ensemble technique in machine learning?

Answer: An ensemble technique in machine learning involves combining multiple models to create a more robust and accurate predictive model. The idea is that a group of weak learners, when combined, can produce a stronger learner that performs better than any individual model alone. Ensemble techniques leverage the strengths of multiple models to create a more powerful predictive model. They are widely used in various machine learning competitions and practical applications due to their ability to improve model performance and generalization.

7. What is the difference between Bagging and Boosting techniques?

Answer: Here is a tabular differentiation between Bagging and Boosting techniques:

Feature	Bagging	Boosting
Methodology	Multiple models trained independently in parallel on bootstrapped samples	Models trained sequentially, with each new model correcting the errors of the previous models
Focus	Reducing variance	Reducing both bias and variance
Training Data	Random subsets of training data with replacement (bootstrapping)	Entire dataset with adjusted weights for misclassified instances
Model Independence	Independent	Dependent (sequential learning)
Aggregation	Majority voting (classification) or averaging (regression)	Weighted sum of predictions based on model performance
Complexity	Simpler to implement and parallelize	More complex, harder to parallelize due to sequential nature
Error Handling	Averages out errors across models	Focuses on correcting errors of previous models

Weighting	Each model has equal weight	Models are weighted based on their accuracy
Examples	Random Forest	AdaBoost, Gradient Boosting, XGBoost
Strengths	Reduces overfitting, improves stability	Reduces both bias and variance, often leads to significant performance improvements
Weaknesses	May not perform well if base models are weak learners with high bias	More prone to overfitting if not properly regularized, computationally intensive
Performance	Better with strong, high-variance models (e.g., deep trees)	Effective with weak learners (e.g., shallow trees), improving overall model performance

This table highlights the fundamental differences between Bagging and Boosting, illustrating their methodologies, focus, complexity, strengths, and weaknesses.

Both bagging and boosting aim to improve model performance, but they do so in different ways; bagging focuses on reducing variance through parallel model training and aggregation, while boosting focuses on reducing both bias and variance by sequentially training models to correct errors.

8. What is out-of-bag error in random forests?

Answer: Out-of-bag (OOB) error is a method for evaluating the performance of a Random Forest model without the need for a separate validation dataset. It leverages the unique structure of Random Forests, which are built using bootstrap aggregating (bagging). OOB Error can also be said to be an internal validation metric in Random Forests calculated using data points not included in the bootstrap samples used to train individual trees.

9. What is K-fold cross-validation?

Answer: K-fold cross-validation is a technique used to assess the performance and generalizability of a machine learning model. It involves dividing the dataset into K equal (or nearly equal) parts, or "folds," and then performing the training and validation process K times. K-fold cross-validation is a powerful and widely used method for model validation that helps in obtaining a reliable estimate of a model's performance. It makes efficient use of data and provides a robust mechanism to gauge how well a model is likely to generalize to unseen data.

10. What is hyper parameter tuning in machine learning and why it is done?

Answer: Hyperparameter tuning is the process of optimizing the hyperparameters of a machine learning model to improve its performance. Hyperparameters are the parameters of the learning algorithm itself that are set before the training process begins and are not learned from the data. Examples include the learning rate, the number of trees in a random forest, and the regularization parameter in regression models. Hyperparameter tuning is crucial for optimizing the performance of machine learning models. By systematically searching for the best hyperparameter values, we can improve model accuracy, enhance generalization, and ensure the model's complexity is appropriate for the task. This step is essential in building robust, high-performing machine learning systems.

Why Hyperparameter Tuning is Done:

Improve Model Performance: Properly tuned hyperparameters can significantly enhance the predictive accuracy and efficiency of a machine learning model. They help the model better learn the underlying patterns in the training data and generalize well to unseen data.

Avoid Overfitting and Underfitting: Correct hyperparameter settings can help prevent overfitting (where the model learns the noise in the training data) and underfitting (where the model is too simple to capture the underlying data structure).

Balance Bias-Variance Trade-off: Tuning hyperparameters helps in finding the right balance between bias (error due to overly simplistic assumptions in the learning algorithm) and variance (error due to excessive complexity in the learning algorithm).

Optimize Model Complexity: Hyperparameter tuning ensures that the model is neither too complex nor too simple. For instance, in a decision tree, tuning the maximum depth or minimum samples per leaf can control the tree's complexity.

11. What issues can occur if we have a large learning rate in Gradient Descent?

Answer: If we have a large learning rate in Gradient Descent, several issues can occur, affecting the performance and convergence of the optimization process. A large learning rate in Gradient Descent can lead to several issues, including overshooting the minimum, divergence, oscillations, unstable training, and missed minima.

12. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?

Answer: Logistic Regression is a linear classification model and is inherently limited in its ability to classify non-linear data directly, so, we can't use logistic regression for classification of non-linear data. While Logistic Regression is powerful for linearly separable data and provides interpretable probabilities, it is not suitable for directly handling non-linear data due to its inherent linear nature. Here's why:

Limitations with Non-Linear Data:

Inability to Capture Complex Patterns: Non-linear data often requires decision boundaries that are more complex than straight lines (e.g., curves, circles, or more intricate shapes). Logistic Regression cannot capture these non-linear relationships inherently, as it is constrained to linear decision boundaries.

Underfitting Non-Linear Patterns: When applied to non-linear data, Logistic Regression may underfit, meaning it will fail to capture important patterns in the data, leading to poor classification performance.

13. Differentiate between Adaboost and Gradient Boosting.

Answer: Here's a tabular differentiation between Adaboost and Gradient Boosting:

Feature	Adaboost	Gradient Boosting
Base Learner	Typically uses decision trees (weak learners)	Often uses decision trees (weak learners)
Training Method	Sequential	Sequential or Parallel
Weight Adjustment	Adjusts instance weights based on misclassification	Adjusts predictions based on gradients of the loss function
Focus	Correcting misclassifications	Minimizing the overall loss function
Error Minimization	Weighted sum of learners' predictions	Gradient descent optimization
Robustness	Sensitive to noisy data and	More robust to noisy data

	outliers	and outliers
Parallelization	Sequential training of learners	Parallel training of learners
Regularization	Limited options	Regularization parameters (learning rate, tree depth)
Learning Rate	Does not use a learning rate parameter	Uses a learning rate to control tree contributions
Performance	Can overfit without regularization	Effective at reducing both bias and variance
Usage	Effective for complex decision boundaries	Preferred for high predictive accuracy
Example	Face detection	Kaggle competitions, industrial applications

Adaboost: Sequentially adjusts instance weights to focus on correcting misclassifications, sensitive to noisy data.

Gradient Boosting: Uses gradient descent optimization to sequentially or in parallel build an ensemble of trees, more robust and effective at reducing both bias and variance.

14. What is bias-variance trade off in machine learning?

Answer: The bias-variance trade-off is a fundamental concept in machine learning that describes the delicate balance encountered when building models. It relates to how well a model can generalize to unseen data and is crucial for understanding model performance. Understanding and managing the bias-variance trade-off is essential for building effective machine learning models that generalize well to new data. It involves finding the right level of model complexity that balances underfitting and overfitting, ensuring optimal performance on unseen datasets.

15. Give short description each of Linear, RBF, Polynomial kernels used in SVM.

Answer:

Linear Kernel: The linear kernel is the simplest kernel function and is used when the data is linearly separable. It computes the dot product between the input features directly in the original feature space. It's suitable when the data can be separated by a straight line or hyperplane in the input feature space.

Radial Basis Function (RBF) Kernel: The RBF kernel (also known as Gaussian kernel) is widely used due to its flexibility in capturing complex relationships in the data. It

transforms the feature space into infinite dimensions, mapping each data point into a higher-dimensional space where it can be linearly separated. It's effective for non-linear classification tasks where the decision boundary is complex and not easily separable in the original feature space.

Polynomial Kernel: The polynomial kernel computes the dot product of polynomial combinations of the original features, allowing SVMs to capture non-linear relationships between the data points. It's useful for datasets where the decision boundary is polynomial and not linearly separable in the original feature space.

Choosing the appropriate kernel depends on the specific characteristics of the data and the complexity of the decision boundary required for the classification task in SVM.