STATISTICS WORKSHEET-5

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

Ans. d) Expected

2. Chisquare is used to analyse?

Ans. c) Frequencies

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

Ans. c) 6

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

Ans. b) Chisqared distribution

5. Which of the following distributions is Continuous?

Ans. c) F Distribution

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

Ans. b) Hypothesis

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

Ans. a) Null Hypothesis

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

Ans. a) Two tailed

8. Alternative Hypothesis is also called as?

Ans. b) Research Hypothesis

9. 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

Ans. a) np

ASSIGNMENT- 5

MACHINE LEARNING

Q.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?

Ans. R-squared is a better measure of goodness of fit model in regression

Reasons:-

Interpretability: R-squared provides a more intuitive interpretation. It represents the proportion of variance in the dependent variable that is explained by the independent variables in the model. Higher R-squared values indicate a better fit of the model to the data.

Normalized Scale: R-squared is a normalized measure, ranging from 0 to 1. This makes it easier to compare the goodness of fit across different models or datasets. In contrast, RSS is an absolute measure and doesn't provide a clear indication of how well the model fits the data relative to the total variability.

Model Comparison: R-squared allows for direct comparison between models. When comparing models with different numbers of predictors, R-squared can help identify which model provides a better balance between explanatory power and model complexity.

Incorporation of Variability: R-squared considers both the explained and unexplained variability in the data. It quantifies the proportion of total variability that is accounted for by the model, providing a comprehensive assessment of model performance.

Q.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.

Ans. In regression, TSS, ESS and RSS are important metrics used to access the goodness of fit of a regression model.

- Total sum of Squares (TSS): TSS represents the total variability in the dependent variable (Y) and is calculated as the sum of the squared differences between each observed dependent variable value and the mean of the dependent variable.
- Mathematically:
  TSS=∑i=1n(Yi−Y¯)2
- Explained Sum of Squares (ESS): ESS represents the variability in the dependent variable (Y) that is explained by the regression model. It is calculated as the sum of the squared differences between the predicted values of the dependent

variable (obtained from the regression model) and the mean of the dependent variable.

- Mathematically:
  ESS=∑i=1n(Y^i−Y¯)2

- **Residual Sum of Squares (RSS)**: RSS represents the variability in the dependent variable (Y) that is not explained by the regression model, i.e., the discrepancy between the observed values and the predicted values of the dependent variable.
- Mathematically:
  RSS=∑i=1n(Yi−Y^i)2
-

These three metrics are related by the following equation, known as the decomposition of the Total Sum of Squares: **TSS=ESS**

Q3. What is the need of regularization in machine learning?

Ans. Regularization is a technique used in machine learning to prevent overfitting and improve the generalization performance of a model.

- **Preventing Overfitting**: Overfitting occurs when a model learns the training data too well, capturing noise or irrelevant patterns that do not generalize well to unseen data. Regularization introduces additional constraints on the model parameters, discouraging it from fitting the noise in the training data and promoting simpler models that generalize better.
- **Improving Model Stability**: Regularization helps improve the stability of the model by reducing the variance in the parameter estimates.
- **Dealing with Multicollinearity**: In regression problems, multicollinearity occurs when predictor variables are highly correlated with each other. Regularization techniques like Ridge regression can mitigate the effects of multicollinearity by imposing penalties on the magnitude of the coefficients, leading to more stable and interpretable models.

Q.4 What is Gini-impurity index?

Ans. The Gini impurity index, often referred to simply as Gini impurity, is a measure of the impurity or randomness in a set of data.

For a given set of data with K classes, the Gini impurity index IG is calculated as: IG=1−∑i=1Kpi2

Where:
pi is the probability of an element in the set belonging to class i .

K is the total number of classes.

The Gini impurity index ranges between 0 and 1. A value of 0 indicates perfect purity, meaning all the elements in the set belong to the same class. A value of 1 indicates maximum impurity, meaning the elements in the set are evenly distributed across all classes.

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

Ans. Yes, unregularized decision trees are prone to overfitting.

Reasons:-

- High Variance: Unregularized decision trees have high variance, meaning they are sensitive to the specific training data used to build them.
- Complexity: Decision trees have the ability to create complex decision boundaries with many splits, especially if the data is noisy or contains irrelevant features. As the tree grows deeper, it becomes more tailored to the training data, increasing the likelihood of overfitting
- No Constraints: Unregularized decision trees have no constraints on their structure or complexity during training. This lack of constraints allows them to grow freely until they perfectly fit the training data. Without regularization techniques, such as pruning or limiting the depth of the tree, decision trees can become overly complex and overfit the training data.

Q.6 What is an ensemble technique in machine learning?

Ans. An ensemble technique in machine learning involves combining multiple individual models (often referred to as base learners or weak learners) to create a stronger, more robust model that typically outperforms any of the individual models alone.

Ensemble techniques are widely used across various machine learning tasks and algorithms due to their ability to improve predictive performance, reduce overfitting, and increase stability.

There are several ensemble techniques:

- Bagging
- Boosting
- Random Forest
- Stacking
- Voting

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

S.NO	Bagging	Boosting
1.	The simplest way of combining predictions that belong to the same type.	A way of combining predictions that belong to the different types.
2.	Aim to decrease variance, not bias.	Aim to decrease bias, not variance.
3.	Each model receives equal weight.	Models are weighted according to their performance.
4.	Each model is built independently.	New models are influenced by the performance of previously built models.
5.	Different training data subsets are selected using row sampling with replacement and random sampling methods from the entire training dataset.	Every new subset contains the elements that were misclassified by previous models.
6.	Bagging tries to solve the over-fitting problem.	Boosting tries to reduce bias.
7.	If the classifier is unstable (high variance), then apply bagging.	If the classifier is stable and simple (high bias) the apply boosting.
8.	In this base classifiers are trained parallelly.	In this base classifiers are trained sequentially.
9	Example: The Random forest model uses Bagging.	Example: The AdaBoost uses Boosting techniques

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

Ans. The out-of-bag (OOB) error in random forests is an estimate of the model's performance on unseen data, calculated during the training process without the need for cross-validation or a separate validation dataset. It is a unique characteristic of random forests and is a consequence of the bagging technique used in their construction.

Q.9 What is K-fold cross-validation?

Ans. K-fold cross-validation is a technique used to assess the performance and generalization ability of a machine learning model. It involves splitting the original dataset into K equal-sized subsets (or "folds"), then iteratively training the model K times, each time using K-1 folds for training and the remaining fold for validation.

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

Hyperparameter tuning, also known as hyperparameter optimization, refers to the process of selecting the optimal hyperparameters for a machine learning algorithm. Hyperparameters are configuration settings external to the model itself that control its learning process.

Common examples of hyperparameters include learning rate, regularization strength, tree depth, number of hidden layers, and kernel type in support vector machines.

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

Ans. Using a large learning rate in gradient descent optimization algorithms can lead to several issues, including:

- Divergence: With a large learning rate, the updates to the model parameters can become too large, causing the optimization process to overshoot the minimum of the loss function.
- Instability: Large learning rates can cause instability in the optimization process, making it sensitive to small changes in the training data or initialization of model parameters.
- Difficulty in Convergence: Large learning rates may prevent the optimization algorithm from converging to the minimum of the loss function, even if it exists.
- Generalization: Training a model with a large learning rate can lead to poor generalization performance on unseen data.

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

Ans. Yes, we can use Logistic Regression for classification of non-linear data, but it may not capture complex non-linear relationships as effectively as some other algorithms designed specifically for non-linear classification tasks, such as decision trees, support vector machines (SVMs)

To address non-linear classification problems with Logistic Regression, several approaches can be used:

- Feature Engineering

- Ensemble Methods
- Kernel Tricks
- Neural Networks

Q.13 Differentiate between Adaboost and Gradient Boosting.

Ans. AdaBoost (Adaptive Boosting) and Gradient Boosting are both ensemble learning techniques used for boosting, a sequential approach to combining multiple weak learners (usually decision trees) to create a stronger predictive model.

Comparison:

1. Training Procedure:
    - AdaBoost: AdaBoost works by iteratively training a sequence of weak learners, where each learner is trained on a modified version of the dataset. In each iteration, AdaBoost assigns higher weights to misclassified observations from the previous iteration, focusing on the harder-to-classify instances. The final prediction is obtained by combining the predictions of all the weak learners, with more weight given to the predictions of the more accurate models.
    - Gradient Boosting: Gradient Boosting also follows a sequential training process, but it focuses on minimizing the loss function of the model. In each iteration, a new weak learner (typically a decision tree) is trained to fit the residual errors of the previous model. The final prediction is obtained by summing the predictions of all the weak learners, with each learner's contribution weighted by a learning rate parameter.

2. Loss Function:
    - AdaBoost: AdaBoost uses an exponential loss function, which penalizes misclassifications exponentially. It aims to minimize the overall error rate of the ensemble by focusing on the instances that are difficult to classify correctly.
    - Gradient Boosting: Gradient Boosting can be used with various loss functions, such as squared error loss for regression problems or logistic loss for classification problems. It aims to directly minimize the loss function of the model, leading to improved predictive performance.

3. Complexity:
    - AdaBoost: AdaBoost typically uses shallow decision trees (weak learners) as base models, often with a maximum depth of one. This makes AdaBoost models simpler and more interpretable, but they may be prone to overfitting with noisy or complex datasets.

- **Gradient Boosting**: Gradient Boosting can use more complex weak learners, such as deeper decision trees or even neural networks, allowing it to capture more complex patterns in the data. However, this increased complexity may lead to longer training times and a higher risk of overfitting.

Q.14 What is Bias-Variance Tradeoff in machine learning?

Ans. The bias-variance tradeoff is a fundamental concept in machine learning that describes the relationship between the model's bias, variance, and overall predictive performance. It refers to the tradeoff between the model's ability to capture the underlying patterns in the data (bias) and its sensitivity to fluctuations in the training data (variance). Finding the right balance between bias and variance is crucial for building models that generalize well to unseen data.

Bias:

- Bias refers to the error introduced by the simplifying assumptions made by the model to approximate the true relationship between the features and the target variable. A high bias model tends to underfit the training data, meaning it fails to capture the underlying patterns and systematically makes errors.
- Example: A linear regression model applied to non-linear data may exhibit high bias because it cannot capture the curvature of the relationship between the features and the target variable.

Variance:

- Variance refers to the model's sensitivity to fluctuations in the training data. A high variance model captures the noise or random fluctuations in the training data, leading to an overly complex model that fits the training data too closely.
- Example: A high-degree polynomial regression model may have high variance because it fits the training data closely, but it may fail to generalize well to new data due to overfitting.

Tradeoff:

- The bias-variance tradeoff arises because reducing bias often increases variance, and vice versa. Models with high complexity (e.g., many parameters or features) tend to have low bias but high variance, as they can fit the training data closely but may overfit.
- Conversely, models with low complexity (e.g., few parameters or features) tend to have high bias but low variance, as they make simplifying assumptions that may result in underfitting.
- The goal is to find the optimal balance between bias and variance that minimizes the model's overall prediction error on unseen data, known as the irreducible error.

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

Ans. **Linear Kernel**:

- The linear kernel is the simplest kernel function used in SVM.

- It computes the dot product between the feature vectors in the original feature space.
- The decision boundary generated by the linear kernel is a straight line in the feature space.
- Linear kernels are suitable for linearly separable data or when the number of features is very large.

Radial Basis Function (RBF) Kernel:

- The RBF kernel is a popular choice in SVM for handling non-linear decision boundaries.
- It maps the original feature space into a high-dimensional space using a Gaussian function.
- The RBF kernel has two hyperparameters: �γ, which controls the spread of the Gaussian function, and �C, which controls the regularization strength.
- The decision boundary generated by the RBF kernel is non-linear and can capture complex relationships in the data.
- RBF kernels are versatile and can handle a wide range of data distributions.

Polynomial Kernel:

- The polynomial kernel is used in SVM to handle data with polynomial decision boundaries.
- It maps the original feature space into a higher-dimensional space using polynomial functions.
- The polynomial kernel has a hyperparameter �d, which specifies the degree of the polynomial.
- The decision boundary generated by the polynomial kernel is non-linear and can have different shapes depending on the polynomial degree.
- Polynomial kernels are suitable for data with non-linear relationships but may require careful tuning of the degree parameter to avoid overfitting or underfitting.