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

Ans : Expected

2. Chisquare is used to analyse

a) Score

b) Rank

c) Frequencies

d) All of these

Ans.All of these

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

a) 4

b) 12

c) 6

d) 8

Ans : 6

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

a) Normal distribution

b) Chisqared distribution

c) Gamma distribution

d) Poission distribution

Ans : Chisqared distribution

5. Which of the following distributions is Continuous

a) Binomial Distribution

b) Hypergeometric Distribution

c) F Distribution

d) Poisson Distribution

Ans : Binomial Distribution

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

a) Statistic

b) Hypothesis

c) Level of Significance

d) TestStatistic

Ans : 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

Ans : 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

Ans : Two tailed

9. Alternative Hypothesis is also called as?

a) Composite hypothesis

b) Research Hypothesis

c) Simple Hypothesis

d) Null Hypothesis

Ans : 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

Ans : np

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 generally considered the better overall measure of goodness of fit in a regression model compared to Residual Sum of Squares (RSS) for a few key reasons:

R-squared:

Proportion of Explained Variance: R-squared tells you the proportion of the variance in the dependent variable that can be explained by the independent variables in your model. It ranges from 0 to 1, with a value closer to 1 indicating a better fit. It essentially tells you what percentage of the scatter in the data is captured by your model.

Unitless and Scalable: R-squared is a unitless value, making it easier to compare the fit of different models regardless of the units of your data. This allows for consistent interpretation across studies.

RSS:

Absolute Value on Scale: RSS is the sum of the squared errors between the predicted values from your model and the actual values in your data. While a lower RSS suggests a better fit (less error), it's an absolute value and depends on the scale of your data. A smaller dataset with small residuals will naturally have a lower RSS compared to a larger dataset, even if the proportional fit is similar.

Analogy:

Imagine comparing efficiency in two factories:

R-squared: This is like the percentage of raw materials a factory uses to create products (explained variance). A higher percentage indicates better efficiency.

RSS: This is like the total amount of leftover waste material. A lower amount suggests efficiency, but it depends on the total volume of materials processed.

However, RSS does have some advantages:

Outlier Sensitivity: RSS is more sensitive to outliers in your data. A single data point far from the fitted line can significantly increase the RSS, prompting you to investigate that point.

Early Evaluation: RSS is easier to calculate in the initial stages of model building, helping you assess the basic fit before finalizing the model.

In conclusion:

Use R-squared for a general assessment of how well your model fits the data, considering the proportion of variance explained.

Use RSS to identify potential outliers and for initial model evaluation.

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 analysis, TSS, ESS, and RSS are all measures of variation used to understand how well a model fits the data. Here's their breakdown and the relationship between them:

Total Sum of Squares (TSS): This represents the total variability of the dependent variable (Y) around its mean ($\bar{Y}$). It essentially captures how spread out the data points are from the average value.

Explained Sum of Squares (ESS): This represents the variation of the dependent variable that is explained by the regression model. In simpler terms, it shows how much better the predicted values from the model fit the data points compared to just using the mean of Y.

Residual Sum of Squares (RSS): This represents the unexplained variation, or the error between the actual values of the dependent variable and the predicted values from the model. It basically captures the "noise" in the data that the model couldn't account for.

The key relationship between these three is:

TSS = ESS + RSS

This equation states that the total variability in the data (TSS) can be divided into two parts:

The variation explained by the model (ESS)

The unexplained variation or error (RSS)

This decomposition of variance helps us assess how well the model fits the data. A higher proportion of ESS to TSS indicates a better fit, meaning the model explains a larger portion of the total variability.

Here are some additional points to note:

These terms are sometimes denoted differently, like SST, SSR, and SSE. The meaning remains the same.

The concept applies to various regression models, not just linear regression.

I hope this explanation clarifies the meaning and relationship between TSS, ESS, and RSS in regression analysis!

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

Ans : In machine learning, regularization is an essential approach that helps avoid overfitting, a typical issue. The following explains why it's crucial:

Overfitting: Envision a situation where a model tries to learn every nuance of the training set, even wiggles and bumps that don't correspond to the main pattern. As a result, the model performs poorly on unseen data but well on training data. Regularization approaches serve as a kind of safety net, keeping the model from growing too intricate and only identifying the noise present in the data.

Generalizability: An accurate prediction on fresh data, not simply the data it was trained on, is the aim of a machine learning model. This is made possible via regularization, which encourages the use of simpler, more broadly applicable models. A more basic model is unlikely to

4. What is Gini–impurity index?

Ans : The statistic known as Gini impurity index, or simply Gini impurity, is employed in decision tree techniques within machine learning. It aids in determining the node's purity or impureness within the tree.

Here's how the idea is broken down:

Calculates the probability that a random data point will be misclassified at a specific decision tree node.

Values: For binary classification, the range is 0 to 1.

Interpretation

0: Complete purity, meaning every data point in the node is a member of the same class.

0.5: Maximum impurity; all classes in the node are distributed equally.

Different levels of impurity are indicated by values between 0 and 0.5.

In essence, it indicates the degree of class separation at a certain node. Gini is used by the decision tree algorithm.

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

Ans : Unrestricted Growth: To classify data, decision trees naturally produce a sequence of yes/no questions. Until they precisely fit the training data, unregularized trees are able to add splits, or questions, to the data indefinitely. This builds a sophisticated tree that learns every nuance of the training set, but it might not adapt well to new data.

Concentration on Training Data: The tree gives top priority to fitting the training data as closely as feasible when there are no constraints. Because of this, the tree may end up catching noise or erratic swings in the data that are absent from fresh data.

Consider a decision tree that is attempting to categorize oranges and apples. A extremely specific query such as "Is the peel slightly bumpy and exactly 2 inches wide?" may be asked of an unregularized tree. This may work out brilliantly.

6. What is an ensemble technique in machine learning?

Ans : In machine learning, ensemble methods are a potent means of enhancing the accuracy of predictive models. The main idea is to build a single, more resilient model by combining the qualities of numerous models. Although there are other methods to accomplish this, the main objective is to lower errors and raise prediction accuracy.

The use of ensemble methods has the following benefits:

Increased Accuracy: Ensemble approaches, which combine several models, frequently produce results that are more accurate than those of any one model.

Diminished Variance: Variance is the amount that a model's predictions can vary when multiple training sets are used. By averaging the predictions made by several models, ensemble approaches can aid in the reduction of variation.

Decreased Bias: Bias is the term for the systematic mistakes that can

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

Ans : Both bagging and boosting are ensemble techniques in machine learning used to improve the performance of a model by combining predictions from multiple models. However, they differ in their approach:

Bagging (Bootstrap aggregating):

Training data: Creates multiple random subsets of the original training data (with replacement). Each model is trained on a different subset.

Model type: Typically uses the same type of weak learner (e.g., decision trees) for all models in the ensemble.

Model weights: All models are given equal weight in the final prediction (usually averaging the predictions).

Focus: Reduces variance by averaging out errors from individual models. This is effective for unstable models with high variance.

Boosting:

Training data: Sequentially trains models, where each new model focuses on the data points that the previous models struggled with.

Model type: Can use different types of models in the ensemble.

Model weights: Models with better performance on the training data are given higher weights in the final prediction.

Focus: Reduces bias by sequentially improving on the predictions of prior models. This is effective for models with high bias (underfitting).

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

Ans : Out-of-bag (OOB) error in random forests is a technique to gauge how well your model might function on hypothetical data. It makes use of a method known as bootstrapping to capitalize on the intrinsic quality of random forest construction.

The OOB error breakdown is as follows:

Starting from scratch: Each tree in a random forest is trained using a portion of the initial data that was produced via sampling with replacement. This implies that certain data points may be selected more than once, while other data points may not be included at all. We refer to these remaining points as out-of-bag (OOB) samples.

OOB Calculation Error: By utilizing only the trees that excluded a certain data point from their training, the OOB error can be defined as the average prediction error for every data point in the training set.

9. What is K-fold cross-validation?

Ans : In machine learning, k-fold cross-validation is a technique used to evaluate a model's performance on omitted data. It entails dividing the data into k folds, or groups, of equal size. Next, the subsequent actions are carried out k times:

The remaining k-1 folds are utilized as the training set (data used to train the model), and one fold serves as the testing set (data held aside for evaluation).

On the training set, the model is trained.

On the testing set, the model's performance is assessed.

You receive k performance metrics as a result of this process, one for each fold that is tested. To obtain a more reliable measure of the model's generalizability on fresh data, these metrics are then averaged.

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

Ans : In machine learning, hyperparameter tuning is the process of finding the optimal settings for a model's configuration variables. These variables, called hyperparameters, are distinct from regular model parameters. Here's a breakdown: Hyperparameters: These are settings defined before training a model. They control the learning process itself, rather than being learned from the data. Examples include the number of trees in a random forest or the learning rate in an optimization algorithm. Model Parameters: These are internal values the model learns from the training data. They influence the model's specific predictions. For instance, coefficients in a linear regression model are parameters. Hyperparameter tuning is crucial because it significantly impacts a model's performance. By adjusting these settings, you can essentially fine-tune the learning process to achieve the best possible results on your specific data. Here's why it's important: Improves Model Performance: Hyperparameters influence how well a model generalizes to unseen data. Tuning them helps prevent issues like underfitting (model is too simple and doesn't capture the data's patterns) and overfitting (model memorizes the training data but performs poorly on new data). Balances Model Complexity: Proper hyperparameter tuning can help achieve a balance between model complexity and accuracy. A complex model might perform well on the training data but fail to generalize. Tuning helps find the right level of complexity for the task. Optimizes Learning Algorithm: Different hyperparameter values can affect how quickly a learning algorithm converges (reaches an optimal state). Tuning helps find settings that lead to efficient training and optimal results. Overall, hyperparameter tuning is an essential step in the machine learning workflow. It allows you to squeeze the best possible performance out of your chosen machine learning model.

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

Ans : There are a couple of issues that can arise if you use a large learning rate in Gradient Descent, which is commonly used to train machine learning models:

Overshooting the Minimum: Imagine you're rolling a marble down a hill to find the lowest point. A large learning rate acts like a big push. You might shoot right past the lowest point (the minimum) and end up rolling back and forth on the other side, never settling on the optimal solution. In Gradient Descent, this translates to the algorithm bouncing around the minimum instead of converging on it.

Instability and Divergence: With a large learning rate, the updates to the model's parameters can be too large in each iteration. This can cause the training process to become erratic and unstable. In the worst case, the loss (error) can keep increasing, and the model's performance will degrade significantly – diverging from ever finding a good solution.

Overfitting: A large learning rate can also lead to overfitting. This happens when the model memorizes the training data too well and becomes overly sensitive to specific details that might not generalize well to unseen data.

In essence, a large learning rate can make Gradient Descent behave more like a bull in a china shop, missing the optimal solution and potentially causing more harm than good.

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

Ans : The main application of logistic regression is in linear classification issues. This means that in feature space, it functions best when the decision boundary dividing the classes can be depicted as a straight line.Nonetheless, logistic regression can be used to handle non-linear data in the following ways:

Feature Transformation: By creating new features based on the existing features, we can make the relationship more linear. This may entail applying further non-linear transformations or generating polynomial terms out of the current features (such as x^2, x*y, etc.).Ignoring (even slight) non-linearity Sometimes, especially if the non-linearity is minor, logistic regression can still produce a good enough model even with a little non-linear relationship. Interpretability is the trade-off in this case; compared to a really non-linear model, the model may be harder to understand. Inaccurate Decision Boundaries: The intricate curves and shapes required to precisely divide the classes will be beyond the linear model's capacity to represent.

Excessive or insufficient Fit: If the model is unable to achieve a satisfactory fit, it may overfit the training set or be underfit and fail to identify the underlying pattern.

In summary, logistic regression works well with linear data, but feature transformation can make it work with non-linear data as well. Consider employing more suitable models, such as decision trees, support vector machines, or neural networks, for genuinely non-linear issues.

13. Differentiate between Adaboost and Gradient Boosting.

Ans : Among the ensemble approaches that are widely used in machine learning are AdaBoost and Gradient Boosting. They function by fusing together several subpar learners (such as decision trees) to produce a single, more effective strong learner. They take different approaches, though:

Improvement's primary focus:

AdaBoost: Concentrates on weighting data. After every iteration, the weights of the training instances are modified. Higher weights are applied to instances that the previous weak student incorrectly classified, making the subsequent learner concentrate more on them.

Remaining mistakes are the main focus of gradient boosting. It examines the mistakes committed by the prior weak learner and attempts to match the mistakes with a new model. In this manner, later learners make up for the errors made by their predecessors.

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

Ans : The bias-variance tradeoff is a fundamental concept in machine learning that deals with the balance between two sources of error in a model's predictions: bias and variance.

Understanding Bias and Variance:

Bias: Bias refers to the systematic error introduced by the model's assumptions. A high bias model makes overly simplified assumptions about the data, leading to underfitting. Underfitting happens when the model fails to capture the underlying patterns in the training data, resulting in poor performance on both the training and unseen data.

Variance: Variance reflects the model's sensitivity to the specific training data it's exposed to. A high variance model learns the specifics of the training data too well, including the noise, and fails to generalize well to new, unseen data. This phenomenon is known as overfitting. Overfitting models perform well on the training data but poorly on unseen data.

The Tradeoff:

The bias-variance tradeoff arises because there's a connection between model complexity and these errors.

Simpler models (low complexity): These models tend to have high bias and low variance. They make strong assumptions that might be too rigid for the data, leading to underfitting.

Complex models (high complexity): These models tend to have low bias and high variance. They can fit the training data very well, potentially memorizing noise along with the actual patterns. This leads to overfitting.

The goal is to find a sweet spot between these two extremes. A model with both low bias and low variance would be ideal – it can capture the important trends in the data without getting overly influenced by the specifics of the training data, and thus perform well on both the training and unseen data.

Finding the Balance:

There are various techniques used to achieve a good bias-variance tradeoff, such as:

Regularization: This technique introduces constraints on the model's complexity, preventing it from overfitting to the training data.Data augmentation: This technique increases the amount and diversity of training data, helping the model learn more generalizable patterns.Model selection: Choosing the right model architecture and hyperparameters can significantly impact the bias-variance tradeoff.By understanding and managing the bias-variance tradeoff, machine learning practitioners can build models that are both accurate and generalizable.

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

Ans : Linear Kernel:

Simplest and fastest kernel.

Well-suited for linearly separable data where a straight line can perfectly divide the classes.

Computes the dot product of the input vectors, essentially performing linear classification in the original feature space.

Advantage: Efficient, interpretable decision boundary (a hyperplane).

Disadvantage: Limited to linearly separable data.

RBF (Radial Basis Function) Kernel:

Most popular and versatile kernel.

Effective for non-linearly separable data by implicitly mapping data points to a higher-dimensional space where they become linearly separable.

Uses a Gaussian function to measure similarity between data points.

Advantage: Handles non-linear relationships well, often good default choice.

Disadvantage: Can be computationally expensive for large datasets, requires tuning a hyperparameter (gamma) that controls the influence of distant data points.

Polynomial Kernel:

Explicitly maps data points to a higher-dimensional polynomial feature space.

Can create complex decision boundaries for non-linear data.

Defined using a degree parameter that determines the complexity of the polynomial features.

Advantage: Can capture a wider range of non-linear relationships compared to the linear kernel.

Disadvantage: Can lead to overfitting if the degree is too high, requires tuning the degree parameter, and may suffer from the "curse of dimensionality" in high-dimensional spaces.

Choosing the Right Kernel:

The optimal kernel choice depends on the specific dataset and problem.

Often, RBF is a good starting point due to its flexibility, but consider the trade-offs between computational cost and overfitting.

Linear kernel is preferred for efficiency and interpretability if the data is already linearly separable.

Polynomial kernel might be considered for specific non-linear patterns, but use caution with the degree parameter.