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?

ANS: R-squared is preferred as a measure of goodness of fit because it directly indicates the proportion of variability in the dependent variable that is explained by the independent variables, making it a more comprehensive and intuitive measure in regression modelling. Please find below sum detail points:

 Interpretability: R-squared provides a clear interpretation of the proportion of the variance in the dependent variable that is explained by the independent variables in the model. It ranges from 0 to 1, where 1 indicates a perfect fit. This makes it straightforward to understand and compare across models.

 Normalized Measure: R-squared is normalized, meaning it is independent of the scale of the data and ranges from 0 to 1. This allows for meaningful comparisons between different models and datasets.

 Direct Relationship with Fit: R-squared increases as the model's predictions better fit the observed data. It quantifies how well the regression model approximates the real data points.

 RSS as a Component: RSS, on the other hand, is a component used in the calculation of R-squared. It represents the sum of squared residuals (differences between observed and predicted values). While RSS provides information about the magnitude of residuals, it doesn't directly indicate how well the model explains the variance in the dependent variable.

 Contextual Use: RSS is more commonly used in statistical tests, such as F-tests for overall significance of the regression model or in calculating standard errors of coefficients. Its role is important in regression analysis but doesn’t serve as a standalone measure of goodness of fit in the same way R-squared does.

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: Actually TSS ESS and RSS means in short:

- **TSS** is the total variability in Y.
- **ESS** is the variability explained by the regression model.
- **RSS** is the unexplained variability in Y after accounting for the regression model.

➢ **Total Sum of Squares (TSS)**:

Mathematically, TSS is calculated as the sum of squared differences between each observed value of Y and the mean of Y:

$$TSS=\sum(Yi-\bar{Y})2$$

- **Explained Sum of Squares (ESS)**:

  It quantifies how well the regression model fits the data by summing the squared differences between the predicted values (Y-hat) and the mean of Y:
  where Yi are the predicted values of Y from the regression model, and $\bar{Y}$ is the mean of Y.

$$ESS=\sum(Yi-\bar{Y})2$$

- **Residual Sum of Squares (RSS)**:

  It is the sum of squared differences between the observed values of Y and the predicted values (residuals) where Yi are the observed values of Y, and Yi are the predicted values of Y from the regression model:

$$RSS=\sum(Yi-Yi)2$$

These metrics are crucial in regression analysis for evaluating how well the model fits the data and how much of the variability in the dependent variable is explained by the independent variables.

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

ANS: Regularization in machine learning is a technique used to prevent overfitting and improve the generalization of machine learning models. Overfitting occurs when a model learns not only the underlying patterns in the training data but also the noise and random fluctuations. This results in a model that performs well on the training data but fails to generalize to unseen data (testing data or real-world data). Regularization helps address this issue by adding a penalty term to the model's objective function, which discourages the model from fitting the training data too closely.

- **Preventing Overfitting**:
- **Improving Model Generalization**:
- **Handling Multicollinearity**:
- **Feature Selection**:
- **Bias-Variance Trade-off:**
- **Handling Large Amounts of Data**

Hence regularization is essential in machine learning to improve model performance, prevent overfitting, enhance generalization, and ensure robustness in handling complex datasets and real-world applications.

4. What is Gini–impurity index?

ANS: Gini Index is a powerful measure of the randomness or the impurity or entropy in the values of a dataset. Gini Index aims to decrease the impurities from the root nodes (at the top of decision tree) to the leaf nodes (vertical branches down the decision tree) of a decision tree model.

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

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

- **High Variance**: Decision trees have a tendency to fit the training data very closely, capturing noise and outliers along with the underlying patterns
- **Complexity**: Unregularized decision trees can grow to be very deep and complex, especially if there are many features or if the data is noisy.
- **Memorization of Noise**: Decision trees can memorize the training data if not constrained.
- **Sensitive to Small Variations**: Unregularized decision trees can be highly sensitive to small variations in the training data.
- **Lack of Generalization**: Overfitting occurs when a model captures noise rather than the underlying relationships in the data.

6. What is an ensemble technique in machine learning?

ANS: Ensemble learning refers to a machine learning approach where several models are trained to address a common problem, and their predictions are combined to enhance the overall performance.

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

ANS: Bagging and boosting are ensemble learning techniques. Bagging (Bootstrap Aggregating) reduces variance by averaging multiple models, while boosting reduces bias by combining weak learners sequentially to form a strong learner.

below are the some difference points:

- **Training Approach**: Bagging trains base models independently and in parallel, whereas boosting trains base models sequentially, where each subsequent model learns from the mistakes of its predecessor.

- **Weight Assignment**: Bagging assigns equal weight to each base model's prediction, while boosting assigns weights dynamically based on the performance of each base model.

- **Purpose**: Bagging aims to reduce variance and improve stability by averaging predictions from diverse models, while boosting focuses on reducing bias and improving accuracy by sequentially adjusting predictions to correct errors.

- **Examples**: Random Forest is a typical example of bagging, while AdaBoost and Gradient Boosting Machines (GBM) are examples of boosting algorithms.

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

ANS: Out-of-bag (OOB) error, also called out-of-bag estimate, is a method of measuring the prediction error of random forests, boosted decision trees, and other machine learning models utilizing bootstrap aggregating (bagging).

9. What is K-fold cross-validation?

ANS: K-fold cross-validation is a popular technique used in machine learning for assessing the performance of a predictive model. It helps in evaluating how the model will generalize to an independent dataset by splitting the original dataset into multiple subsets or folds. Here's how K-fold cross-validation works

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

ANS: Hyperparameter tuning in machine learning refers to the process of selecting the optimal hyperparameters for a given model. Hyperparameters are configuration variables that are set before the learning process begins and are not directly learned from the data. They control aspects of the learning algorithm and have a significant impact on the performance of the model being trained.

Examples of Hyperparameters:

- **Learning Rate** in gradient descent optimization algorithms.
- **Number of Hidden Units** in neural networks.
- **Depth of Trees** in decision tree algorithms.
- **Regularization Parameters** (e.g., lambda in Lasso or Ridge regression).
- **Kernel Parameters** in Support Vector Machines (SVMs).

In summary, hyperparameter tuning is essential in machine learning to optimize model performance, prevent overfitting, and enhance model robustness. It is a critical step in the model development process to ensure that the chosen algorithm performs effectively on new, unseen data, thereby improving the overall reliability and applicability of machine learning models in real-world scenarios.

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

ANS: Having a large learning rate in Gradient Descent can lead to several issues that affect the training and convergence of the model. Here are the key issues:

1. **Overshooting the Minima**:

   A large learning rate causes the updates to the model parameters (weights) to be too large in each iteration. This can lead to the parameters oscillating around the minimum of the loss function or even diverging away from it. In extreme cases, the parameters might overshoot the minimum and fail to converge.

2. **Instability and Divergence**:

   The rapid changes in parameter values due to large updates can make the optimization process unstable. Instead of gradually converging towards the optimal values, the parameters may fluctuate erratically or diverge entirely, making it impossible to find a good solution.

3. **Slow Convergence**:

   Surprisingly, a very large learning rate can also slow down the convergence of the optimization process. This happens because the updates may consistently overshoot the optimal values, causing the algorithm to take longer to settle around the minimum of the loss function.

4. **Difficulty in Finding the Optimal Solution**:

   With a large learning rate, Gradient Descent may fail to fine-tune the parameters effectively. Instead of making small adjustments to reach the optimal solution, the updates can cause the algorithm to miss or skip over the optimal values, resulting in suboptimal model performance.

5. **Sensitivity to Data Noise**:

   Large learning rates can make the optimization process more sensitive to noise in the training data. Small fluctuations or outliers in the data can lead to disproportionately large updates in the parameter values, which may prevent the model from learning robust patterns and instead focus on noise.

6. **Unstable Loss Curve**:
   - When plotting the loss function over iterations, a large learning rate can lead to a jagged or unstable loss curve. Instead of a smooth decrease in loss, the curve may show sharp fluctuations or irregular patterns, indicating instability in the optimization process.

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

ANS: Logistic Regression is a linear classification algorithm and, by definition, it models the relationship between the dependent binary variable (output) and one or more independent variables (predictors) using a linear function. This linear relationship assumes that the decision boundary that separates the classes is linear in nature.

13. Differentiate between Adaboost and Gradient Boosting.

ANS: Adaboost and Gradient Boosting are both ensemble learning techniques that improve the predictive performance of machine learning models by combining multiple weaker learners (often decision trees) into a stronger composite model. Despite their similarities in using boosting as a principle, they differ in their approaches and how they incorporate the concept of boosting.
Differences:

Optimization Approach: Adaboost uses additive models with a focus on misclassified instances and weighted voting, while Gradient Boosting optimizes an objective function using gradient descent principles to minimize residual errors.

Sequential Training: Adaboost sequentially adjusts weights and focuses on difficult instances, whereas Gradient Boosting sequentially fits trees to residual errors to improve the overall model performance.

Loss Function: Adaboost doesn't directly optimize a loss function but rather adjusts instance weights, whereas Gradient Boosting directly minimizes a loss function through gradient descent optimization.

Application: Adaboost is often used with simple base learners like decision stumps, whereas Gradient Boosting is flexible and can be applied with various weak learners, including decision trees of varying depths.

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

ANS: The bias-variance trade-off is a fundamental concept in machine learning that helps in understanding the sources of error in predictive models and guides the selection of appropriate algorithms and model complexity. It refers to the delicate balance between bias and variance in supervised learning models and how they affect the overall predictive performance.

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

ANS: Support Vector Machines (SVMs) are powerful supervised learning models used for classification and regression tasks. They are particularly effective in high-dimensional spaces and when the data has clear margins of separation. SVMs can utilize different types of kernels to transform the input space into a higher-dimensional space where the data may be linearly separable. Here's a brief description of commonly used kernels in SVM: