MACHINE LEARNING

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 generally preferred because it provides a proportion of the variance explained by the model and is standardized, making it easier to compare across different models and datasets.

RSS is useful for understanding the absolute amount of error but lacks context regarding the proportion of variability explained by the model.

Thus, R-squared is typically a better measure of goodness of fit in regression analysis as it offers a relative measure of model performance.

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:

1. Total Sum of Squares (TSS)

TSS measures the total variance in the dependent variable. It quantifies the total amount of variability in the observed data points relative to their mean.

TSS=∑(yi−y¯)2TSS = \sum (y_i - \bar{y})^2TSS=∑(yi−y¯)2 where yiy_iyi is an individual observed value, and y¯\bar{y}y¯ is the mean of the observed values.

2. Explained Sum of Squares (ESS)

ESS represents the part of the total variance that is explained by the regression model. It measures how much of the variability in the dependent variable is accounted for by the model.

ESS=∑(y^i−y¯)2ESS = \sum (\hat{y}_i - \bar{y})^2ESS=∑(y^i−y¯)2 where y^i\hat{y}_iy^i is the predicted value from the regression model, and y¯\bar{y}y¯ is the mean of the observed values.

3. Residual Sum of Squares (RSS)

- RSS measures the portion of the variance that is not explained by the model. It quantifies the discrepancies between the observed values and the predicted values.

- RSS=∑(yi−y^i)2RSS = \sum (y_i - \hat{y}_i)^2RSS=∑(yi−y^i)2 where yiy_iyi is an individual observed value, and y^i\hat{y}_iy^i is the predicted value from the regression model.

The relationship between these metrics is given by the following equation:

TSS=ESS+RSSTSS = ESS + RSSTSS=ESS+RSS

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

ANS:

Regularization in machine learning is a technique used to prevent overfitting and to improve the generalization of a model. Here's why regularization is needed:

1. Prevent Overfitting
2. Improve Model Generalization
3. Simplify Models
4. Handle Multicollinearity
5. Encourage Robustness

Regularization is essential in machine learning to prevent overfitting, improve model generalization, simplify models, handle multicollinearity, and encourage robustness. It achieves this by adding a penalty to the loss function based on the magnitude of the model coefficients, thus encouraging simpler and more generalizable models.

4 . What is Gini–impurity index?

ANS:
The Gini impurity index is a metric used in decision tree algorithms to quantify the purity of a node. It helps in making decisions about the best splits to create more homogeneous nodes and ultimately build an accurate and efficient decision tree.

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

ANS:
Yes, Unregularized decision trees are prone to overfitting because they can grow very complex, capturing noise and specific patterns in the training data that do not generalize to new data. Regularization techniques such as pruning, depth limiting, and setting minimum samples for splits and leaves are used to manage the complexity of decision trees and improve their generalization performance.

6. What is an ensemble technique in machine learning?

ANS:
Ensemble techniques in machine learning involve combining the predictions of multiple models to produce a more accurate and robust final prediction. Common methods include bagging, boosting, stacking, and voting. By leveraging

the strengths of various models and aggregating their predictions, ensemble techniques can significantly improve the performance and generalization of machine learning models.

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

ANS:
Bagging and Boosting are complementary techniques that address different aspects of model performance. Bagging focuses on reducing variance and improving stability, while boosting aims to improve accuracy by addressing both bias and variance through iterative correction of errors.

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

ANS:
The **out-of-bag (OOB) error** is a concept used in the context of Random Forests, a popular ensemble learning method. It provides a way to estimate the performance of a Random Forest model without the need for a separate validation dataset.
The out-of-bag error is an estimate of the generalization error (or test error) of a Random Forest model. It is calculated using the observations that were not included in the bootstrap sample used to train each individual tree in the forest.

9. What is K-fold cross-validation?

ANS:
K-fold cross-validation is a technique for evaluating machine learning models by partitioning the dataset into KKK subsets and performing KKK rounds of training and validation. Each fold serves as the validation set exactly once, and the results are averaged to provide a robust estimate of the model's performance. This method reduces bias and ensures that all data points are used for both training and validation, leading to a more reliable performance assessment

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

ANS:
Hyperparameter tuning is an essential process in machine learning aimed at finding the optimal set of hyperparameters to improve the performance, generalization, and efficiency of a model. It involves various methods such as grid search, random search, Bayesian optimization, and AutoML. Proper hyperparameter tuning can significantly enhance a model's accuracy and robustness, leading to better predictions and more effective machine learning solutions.

Hyperparameter Tuning is done to:

1. **Improve Model Performance:**
    - Optimal hyperparameters can significantly improve the model's performance by finding the best configuration for the given problem.

2. **Enhance Generalization:**
    - Proper tuning helps in finding a balance between bias and variance, ensuring the model generalizes well to new, unseen data and avoids overfitting or underfitting.
3. **Optimize Training Time:**
    - Efficient hyperparameter settings can reduce training time and computational resources. For example, finding a suitable learning rate can lead to faster convergence.
4. **Model Robustness:**
    - Tuning can make the model more robust and stable across different datasets and scenarios by fine-tuning its settings to the specific characteristics of the data.

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

ANS:
A large learning rate in Gradient Descent can lead to several issues, including divergence, instability, poor convergence, oscillation, and exploding gradients. These problems can adversely affect the training process and model performance. To mitigate these issues, consider using learning rate schedules, adaptive learning rate methods, gradient clipping, and manual tuning to find an appropriate learning rate for your model and problem.

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

ANS:
Logistic Regression is best suited for linearly separable data due to its linear decision boundary and the linear relationship it assumes between features and the log odds of the outcome. For non-linear data, logistic regression may not perform well on its own. However, you can handle non-linear relationships by using feature engineering (e.g., polynomial features), leveraging kernel methods in related algorithms, exploring non-linear extensions like GAMs, or opting for other algorithms better suited for non-linear classification.

13. Differentiate between Adaboost and Gradient Boosting.

ANS:

AdaBoost adjusts the weights of training samples based on classification errors from previous models and combines weak learners into a strong learner. It focuses on misclassified instances and may be sensitive to noisy data and outliers.

Gradient Boosting builds models to correct residual errors from previous models, directly optimizing a loss function using gradient descent. It tends to handle noisy data better and includes techniques to control overfitting.

Both AdaBoost and Gradient Boosting are powerful ensemble methods, but they have different approaches and are suited to different types of problems and datasets. Choosing between them depends on the specific characteristics of the data and the goals of the modeling process.

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

ANS:
The **bias-variance trade-off** is a fundamental concept in machine learning that describes the balance between two sources of error that affect the performance of a predictive model: bias and variance. Understanding this trade-off is crucial for building models that generalize well to new, unseen data.

Achieving this balance requires careful selection of model complexity, regularization techniques, and validation methods to ensure optimal performance and generalization.

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

ANS:

Support Vector Machines (SVMs) use kernel functions to handle non-linear relationships by transforming the original feature space into a higher-dimensional space where a linear decision boundary can be applied. Different kernels provide different ways to perform this transformation. Here's a brief description of the three commonly used kernel functions: **Linear**, **RBF (Radial Basis Function)**, and **Polynomial** kernels.

1. Linear Kernel

   - The Linear kernel is the simplest kernel function. It essentially performs no transformation and computes the dot product of the input vectors in the original feature space.

2. RBF (Radial Basis Function) Kernel

   - The RBF kernel, also known as the Gaussian kernel, transforms the input space into an infinite-dimensional space, allowing the SVM to fit the data in a highly flexible way.

3. Polynomial Kernel

   - The Polynomial kernel computes a dot product raised to a certain power, effectively mapping the data into a higher-dimensional space based on polynomial functions.