MACHINE LEARNING ANSWERS

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 ofgoodness of fit model in regression and why?**

R-squared (R2) and Residual Sum of Squares (RSS) are both commonly used measures to assess the goodness of fit of a regression model, but they capture different aspects of model performance.

R**-squared (R2):**

1. R-squared is a measure of how well the independent variables explain the variation in the dependent variable.
2. It ranges from 0 to 1, where 1 indicates a perfect fit and 0 indicates no improvement over using the mean of the dependent variable.
3. R-squared is often preferred because it provides a standardized measure of goodness of fit that is easy to interpret. Higher R-squared values indicate better model fit.

Residual Sum of Squares (RSS):

1. RSS is a measure of the discrepancy between the observed values of the dependent variable and the values predicted by the model.
2. It represents the sum of the squared differences between the observed and predicted values of the dependent variable.
3. Lower values of RSS indicate a better fit, as it means that the model's predictions are closer to the actual observed values.

So, which measure to use depends on the context and what aspect of model performance you want to emphasize:

R-squared is generally preferred when you want to assess how well the independent variables explain the variability in the dependent variable. It's useful for comparing different models and determining the proportion of variance explained by the model.

RSS can be useful when you want to directly assess the magnitude of the errors made by the model. It's particularly useful for diagnostic purposes, such as identifying outliers or assessing the adequacy of the model's assumptions.

In summary, R-squared is typically used as the primary measure of goodness of fit because of its interpretability and ease of comparison, but RSS can provide additional insights into the accuracy of the model's predictions.

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

In regression analysis, TSS (Total Sum of Squares) represents the total variability in the dependent variable. ESS (Explained Sum of Squares) measures the variability explained by the regression model. RSS (Residual Sum of Squares) quantifies the unexplained variability, which is the discrepancy between the observed values and the values predicted by the model.

The equation relating these three metrics is:

TSS = ESS + RSS

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

Regularization is one of the most important concepts of machine learning. It is a technique to prevent the model

from overfitting by adding extra information to it.
Sometimes the machine learning model performs well with the training data but does not perform well with the test data. It means the model is not able to predict the output when deals with unseen data by introducing noise in the output, and hence the model is called overfitted. This problem can be deal with the help of a regularization technique.

This technique can be used in such a way that it will allow to maintain all variables or features in the model by reducing the magnitude of the variables. Hence, it maintains accuracy as well as a generalization of the model.

It mainly regularizes or reduces the coefficient of features toward zero. In simple words, "In regularization technique, we reduce the magnitude of the features by keeping the same number of features."

4. What is Gini–impurity index?

Gini impurity measures how often a randomly chosen element of a set would be incorrectly labeled if it were labeled randomly and independently according to the distribution of labels in the set. It reaches its minimum (zero) when all cases in the node fall into a single target category.

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

Yes, unregularized decision trees are prone to overfitting. Decision trees have a natural tendency to create complex structures that perfectly fit the training data, which can lead to capturing noise or outliers in the data. Without regularization techniques such as pruning or limiting the depth of the tree, the model can become overly complex and fail to generalize well to unseen data. Overfitting occurs when the model learns the noise or random fluctuations in the training data instead of the underlying pattern, resulting in poor performance on new data.

6. What is an ensemble technique in machine learning?

Ensemble learning is a machine learning technique that enhances accuracy and resilience in forecasting by merging predictions from multiple models. It aims to mitigate errors or biases that may exist in individual models by leveraging the collective intelligence of the ensemble.

The underlying concept behind ensemble learning is to combine the outputs of diverse models to create a more precise prediction. By considering multiple perspectives and utilizing the strengths of different models, ensemble learning improves the overall performance of the learning system. This approach not only enhances accuracy but also provides resilience against uncertainties in the data. By effectively merging predictions from multiple models, ensemble learning has proven to be a powerful tool in various domains, offering more robust and reliable forecasts.

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

Bagging	Boosting
1. Bagging is a learning approach that aids in enhancing the performance, execution, and precision of machine learning algorithms.	1. Boosting is an approach that iteratively modifies the weight of observation based on the last classification.
2. It is the easiest method of merging predictions that belong to the same type.	2. It is a method of merging predictions that belong to different types.
3.In bagging, each model is assembled independently.	3. In boosting, the new models are impacted by the implementation of earlier built models.
4. It helps in solving the over-fitting issue.	4. It helps in reducing the bias.
5. In the case of bagging, if the classifier is unstable, then we apply bagging.	5. In the case of boosting, If the classifier is stable, then we apply boosting.
6. Here, every model has equal weight.	6. Here, the weight of the models depends on their performance.

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

The out-of-bag (OOB) error in random forests is an estimate of the model's performance on unseen data without the need for a separate validation set.

In a random forest, each decision tree is trained on a bootstrap sample (a random sample with replacement) of the original dataset. This means that some data points are not included in the training set for each individual tree. The OOB error is calculated by evaluating each data point using only the trees for which it was not included in the training sample.

After training the random forest, the OOB error is computed as the average error across all data points that were left out during the training process. This provides an unbiased estimate of the model's performance on new, unseen data.

The OOB error is useful for assessing the generalization ability of the random forest model and can be used for model selection and tuning hyperparameters.

9. What is K-fold cross-validation?

K-fold cross-validation is a technique used to assess the performance and generalization ability of a machine learning model. It involves splitting the dataset into K equal-sized folds, where K is typically chosen based on the size of the dataset and computational resources.

The cross-validation process then proceeds as follows:

1. The dataset is divided into K folds, with one fold held out as the validation set and the remaining K-1 folds used for training.

2. The model is trained on the K-1 folds and evaluated on the validation fold. This process is repeated K times, with each fold used once as the validation set.

3. After K iterations, the performance metrics (such as accuracy, precision, recall, or F1-score) are averaged across all folds to obtain an overall estimate of the model's performance.

K-fold cross-validation provides a more reliable estimate of the model's performance compared to a single train-test split, as it utilizes the entire dataset for both training and validation. It helps to reduce the variability in the performance estimate and ensures that the model's performance is evaluated on different subsets of the data.

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

Hyperparameter tuning in machine learning refers to the process of selecting the optimal hyperparameters for a given model. Hyperparameters are parameters that are set prior to training and control the learning process, such as the learning rate, regularization strength, number of hidden layers in a neural network, or the depth of a decision tree.

Hyperparameter tuning is done to improve the performance of the model and enhance its ability to generalize to new, unseen data. By selecting the best combination of hyperparameters, the model can achieve higher accuracy, better precision, or improved recall on the validation or test set.

Hyperparameter tuning is typically performed using techniques such as grid search, random search, or more advanced optimization algorithms like Bayesian optimization or genetic algorithms. It involves systematically exploring different combinations of hyperparameters and evaluating the model's performance using cross-validation or a separate validation set.

Overall, hyperparameter tuning is essential for maximizing the performance of machine learning models and ensuring they achieve the best possible results on real-world tasks.

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

If you set a large learning rate in gradient descent, several issues can occur:

1. Overshooting the minimum: A large learning rate can cause the algorithm to overshoot the minimum of the loss function. Instead of converging to the minimum, the algorithm may oscillate back and forth, or diverge altogether.

2. Instability and divergence: A large learning rate can lead to instability in the optimization process. The gradients may fluctuate wildly, causing the optimization algorithm to diverge rather than converge to a solution.

3. Inability to converge: With a large learning rate, the optimization algorithm may fail to converge to a satisfactory solution within a reasonable number of iterations. This is because the steps taken are too large, and the algorithm cannot settle into the minimum of the loss function.

4. Poor generalization: Even if the algorithm converges, a large learning rate may result in a solution that does not generalize well to new, unseen data. The model may overfit the training data or fail to capture the underlying patterns in the data.

To mitigate these issues, it's essential to choose an appropriate learning rate for gradient descent. Techniques such as learning rate decay, adaptive learning rate methods (e.g., AdaGrad, RMSprop, Adam), and careful tuning through hyperparameter optimization can help ensure stable and effective optimization.

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

Logistic regression is a linear classifier, meaning it makes predictions based on a linear combination of the input features. As a result, logistic regression is inherently limited in its ability to capture complex, non-linear relationships between features and the target variable.

While logistic regression can perform well on linearly separable data, it may struggle to effectively classify non-linear data. When faced with non-linear relationships, logistic regression may underfit the data, leading to poor performance and low predictive accuracy.

To address non-linear relationships in the data, more flexible models such as decision trees, random forests, support vector machines with non-linear kernels, or neural networks are typically used. These models are capable of capturing complex patterns and relationships in the data, making them better suited for classification tasks involving non-linear data.

13. Differentiate between Adaboost and Gradient Boosting.

Adaboost	Gradient Boosting
1. During each iteration in AdaBoost, the weight of incorrectly classified samples are increased so that the next week learner focuses more on these samples	1. Gradient Boosting updates the weights by computing the negative gradient of the loss function with respect to the predicted output.
2. AdaBoost uses simple decision trees with one split known as the decision stumps of weak learners.	2.Gradient boosting can use a wide range of base learners, such as decision trees and linear models.
3. AdaBoost is more susceptible to noise and outliers in the data, as it assigns high weights to misclassified samples.	3.Gradient boosting is generally more robust, as it updates the weights based on the gradients, which are less sensitive to outliers.

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

The bias-variance tradeoff is a fundamental concept in machine learning that deals with the balance between bias and variance in model performance.

- *Bias* refers to the error introduced by approximating a real-world problem with a simplified model. A high bias model tends to underfit the data, meaning it fails to capture the underlying patterns and makes overly simplistic assumptions.

- *Variance* refers to the model's sensitivity to fluctuations in the training data. A high variance model tends to overfit the data, meaning it learns the noise and random fluctuations in the training set, leading to poor generalization to unseen data.

The tradeoff arises because reducing bias typically increases variance, and vice versa. Finding the optimal balance between bias and variance is crucial for building models that generalize well to new, unseen data. Techniques like regularization, cross-validation, and ensemble methods (like random forests) are often used to manage the bias-variance tradeoff in machine learning models.

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

Linear Kernel:

Decision Boundary:

Form: The linear kernel produces a decision boundary that is a hyperplane in the feature space. This hyperplane separates data points from different classes in a linear fashion.

Assumption: It assumes that the relationship between the features and the target variable is linear.

Use Cases:

Linearly Separable Data: Linear kernels are most effective when the data can be effectively separated by a straight line, plane (in 3D), or hyperplane (in higher dimensions).

Simplicity: They are computationally less expensive and are suitable for simple, linear relationships.

Example:

2D Data: Imagine a dataset with two features, and classes are separated by a straight line

Polynomial Kernel:

Decision Boundary:

Form: The polynomial kernel introduces non-linearity by using polynomial functions of the original features. The decision boundary can have curves and turns.

Degree Parameter: The degree of the polynomial is a parameter that determines the complexity of the decision boundary.

Use Cases:

Polynomial Relationships: Suitable for data with polynomial relationships between features and classes.

Example:

Data with Curves: Consider a dataset where the relationship between features and classes follows a polynomial curve.

Radial Basis Function (RBF) Kernel:

During Decision Boundary:

Form: RBF kernel, also known as Gaussian kernel, creates a decision boundary based on the similarity of data points to a reference point (or points). It can create complex, non-linear decision boundaries.

Bandwidth Parameter: The bandwidth parameter determines the reach or influence of a data point in defining the decision boundary.

Use Cases:

Complex Relationships: Effective for capturing complex and non-linear relationships in the data.

Example:

Data with Clusters: In a dataset with clusters of data points, the RBF kernel can create decision boundaries around these clusters.