**My Name – Vaishali Ramdas Salve**

**Batch No – DS2311**

**Sir No – Shwetank Mishra**

**Date – 5/01/2024**

**Assignment – 5**

**Assignment Name – Machine Learning**

**Q1 to Q15 are subjective answer type question, Answer them briefly .**

1. **R-squared or Residual Sum of Squares (RSS) which one of these two is a better measure of good ness of fit model in regression and why ?**

**Answe**r - R-Squared is generally considered a better measure of the goodness of fit in a

Regression the proportion of the variance in the dependent variables. It ranges

From 0 to 1, with higher values indicating a better fit.

In contrast, RSS measures the total sum of squared differences between the

Observed and predicted values. While RSS provides information about the

Overall model fit, it doesn’t give a standardized measure like R-squared,

Making it harder to assess the relative goodness of fit across different model.

Therefore, R-squared is preferred as it offers a normalized assessment of the

Models explanatory power, making it easier to compare and interpret the

Goodness of fit across various regression model.

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

**Answe**r- In regression analysis, Total Sum of Squares (TSS) represent the total variability in

The dependent variable, regardless of the model. It is the sum of squared

Differences between each observed dependent variable value and the overall

Mean of the dependent variable.

Explained Sum of Squares (ESS) is the portion of the total variability that is

Explained by the regression model. It is the sum of squares differences between t

The overall mean of the dependent variable.

Residual Sum of Squares ( RSS ) quantifies the unexplained variability in the

Unexplained variability in the dependent variable by summing the squares

Differences between the observed value and the predicted values from the

Regression model .

The relationship between these three metrics is captures by the equation:

*TSS = ESS = RSS*

This equation emphasize that total variability (TSS) can be decomposed into the variability explained by the model ( ESS) and the unexplained residual variability

(RSS).

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

**Answer**- Regularization in machine learning is essential to prevent over fitting , a common problem where a model performs well on training data but fails to generalize to new, unseen data. The need for regularization arises due to the complexity of model, especially when dealing with a large number of features.

Regularization constraints or penalties on the model parameters during the training process. This helps in controlling the models complexity and discourages it from fitting the training data too closely. Here are reasons for using regularization:

1. **Preventing over fitting** : Regularization helps prevent over fitting by penalizing overly complex models.
2. **Handling Multicollinearity** : In the presence of highly correlated features, regularization techniques can mitigate multicollinearity issues.
3. **Feature Selection** : Regularization method, particularly L1 Regularization ( lasso ), can induce sparsity in the model, effectively leading to feature selection.
4. **Improving Model Stability** : Regularization contributes to the stability of the model by preventing extreme parameter values.

In summary, regularization is a vital tool in machine learning to strike a balance between fitting the training data well and ensuring the models ability to generalize to new, unseen data.

4 . **What is Gini-impurity index?**

**Answer** – The Gini impurity index is a measure used in decision tree algorithms to evaluate the impurity or uncertainty of a set of data point within a specific node. In the context of decision trees, each node represents a subset of the data, and the Gini impurity helps in determining the best attribute to split the data.

The Gini impurity for a node is calculated by summing the probabilities of each class being chosen times the probability of a misclassification for that class.

IGini = 1 – {k i=1 p2 i

Where pi is the probability of choosing a data point of class i in that node. A lower Gini impurity indicates a pure node with more homogeneity in terms of class labels.

In the context of decision tress, during the tree-building process, the algorithm aims to minimize the Gini impurity at each node by selecting the attribute and spit that result in the purest child nodes.

In summary, the Gini Impurity index is a criterion used in decision tree algorithms to assess the impurity or disorder with in a node, helping the algorithm make optimal decision for splitting the data based on the features available

1. **Are unregularized decision trees prone to over fitting ? It yes, why**

**Answer** – yes ,unregularzed decision trees are prone to over fitting .decision trees are capable of fitting the training data very closely, capturing intricate deta

ils and noise in the data site . while this can lead to high accuracy on the training set it may not generalize well to unseen data.

The primary reason for over fitting In unregularized decision trees lies in their inherent ability to create complex and deep structures. Unrestricted growth of a decision tree can result in nodes that specifically cater to the training data, capturing noise or outliers that may not be representative of the overall pattern in the data.

Over fitting occurs when a decision tree becomes too tailored to the training set, memorizing its underlying rather than learning the underlying patterns. This can lead to poor performance on new, unseen data because the model has essentially memorized the noise in the training set rather than generalizing the underlying relationships.

Regularization techniques, such as pruning or limiting the maximum depth of the tree are commonly employed to mitigate over fitting in decision trees . pruning involves removing branches of the tree that do not contribute significantly to its predictive power, ensuring a more generalized and robust model.

1. **What is an ensemble technique in machine learning?**

**Answer** - An ensemble technique in machine learning involves combining the predictions of multiple individual models to create a stronger and more robust predictive model. The idea is that by leveraging the diverse strengths of different models, the ensemble can often outperform any of its individual components. There are several popular ensemble techniques, including:

1. **Bagging ( Bootstrap Aggregating ):** Bagging involves training multiple instances of the same learning algorithm on different subsets of the training data, each created by random by random sampling with replacement ( bootstrap samples ). The final prediction is usually an average or a voting mechanism models. Random Forests are a notable example of a bagging ensemble.
2. **Boosting :** Boosting focuses on sequentially training weak learners and giving more weight to the misclassified instances in each iteration. It aims to correct the errors of the previous models.
3. **Stacking : S**tacking involves training multiple diverse models and combining their predictions using another model called a meta-learner. The base models outputs serve as input features for the meta-learner, which makes the final prediction.

Ensemble techniques are powerful tools in machine learning because they can improve model performance, reduce over fitting ,and enhance the models ability to generalize to new, unseen data.

1. **What is the difference between Bagging and Boosting technique?**

**Answer** – Bagging ( Bootstrap Aggregating ) and Boosting are both ensemble techniques in machine learning, but they differ in their approach to combining multiple models:

1. **Bagging (Bootstrap Aggregating ):**

**. Approach:** Bagging involves creating multiple subsets of the training data by randomly sampling with replacement. Each subset is used to train a separate instance of the same learning algorithm.

**. Training :** The individual model in bagging are trained independently and in parallel. They have no knowledge of each others training process**.**

**. Weighted Averaging / Voting :** The final prediction is typically obtained by averaging or voting the prediction of the individual models.

1. **Boosting :**

**. Weighted Training :** Boosting focuses on sequentially training weak learners, giving more weight to the iteration misclassified instances in each iteration each subsequent model corrects the errors of the previous ones.

**. Combining :** The final prediction is a weighted sum of the individual models predictions, with more weight given to models that perform well on the training data.

1. **What Out-Of-Bag error in random forests ?**

**Answer** – The out-of-bag (OOB) error in Random Forests is a measure used to estimate the models predictive performance without the need for a separate validation set.

1. **Bootstrap Sampling :** In the process of constructing each tree in a random forest, a bootstrap sample is drawn from the original training dataset. This means that some data point are not included in the bootstrap sample for each tree.
2. **Out-Of-Instances :** The out-of-bag instances are the data point that were not selected in the bootstrap sample for a particular tree. Since each tree in the Random Forest is built using a different bootstrap sample, there are unique out-of-bag instances for each tree.
3. **Estimating Error :** The out-of-bag error is calculated by evaluating the predictions of each tree on its corresponding out-of-bag instances. For each data point, it is only assessed by the trees that did not include it in their bootstrap sample during training.
4. **Aggregating OOB Errors:** The OOB error is then aggregated over all trees in the Random Forest to provide an over all estimate of the models predictive performance.
5. **What is K-Fold Cross- Validation**?

**Answer** – K-fold cross-validation is a widely used technique in machine learning for assessing the performance and generalization ability of a model. It involves partitioning the dataset into K subsets and then training and evaluating the model K times. In each iteration , one of the K subsets is used as the validation set, while the remaining K-1 subsets are used for training.

1. **Data partitioning** : the dataset is divided into K roughly equal- sized folds.
2. **Training and Validation** : The model is trained k times. In each iteration , one of the k folds is used as the validation set, and the model is trained on the remaining k-1 folds.
3. **Performance Evaluation**: The models performance is evaluated on the validation set for each iteration, resulting in k performance metrics.
4. **Aggregation:** The k performance metrics are averaged to obtain a single performance estimate for the model.
5. **Reduced Variance :** It helps in reducing the variance of the performance estimate by averaging over k different subsets of the data, providing a more reliable assessment of the models generalization performance.
6. **Effective Use of Data :** It maximize the use of validation, ensuring that each data training and validation, ensuring that each data point is used for validation exactly once.
7. **Robustness :** It help in assessing the models performance across different subsets of the data, making the evaluation more robust and less dependent on the specific partitioning of the dataset.
8. **What is hyper parameter tuning in machine learning and why it is done** **?**

**Answer** – Why Hype parameter Tuning is Done :

1. **Optimizing Model Performance :** The choice of hyper parameters significantly influences a models performance. Fine tuning these hyper parameters can lead to improved accuracy, precision, recall, or other relevant metrics, making the model more effective.
2. **Avoiding Over fitting or Under fitting :** Appropriate hyper parameter values help strike the right balance between model complexity and generalization. Tuning can prevent over fitting or under fitting .
3. **Enhancing Model Robustness :** Hyper parameter tuning contribution to the robustness of a real world across different dataset or real world scenarios. A well-tuned model is more likely to generalize well and perform consistently in diverse situations.
4. **Resource Efficiency :** Optimizing hyper parameter can lead to model that require fewer resources while maintaining or even improving performance .
5. **Adapting to Data Characteristics :** Different dataset may require different hyper parameter settings. Tuning allows the model to adapt to the specific characteristics and nuances of the data it is being trained on .
6. What issues can occur if we have a large learning rate in Gradient Descent?

Answer – A large learning rate in Gradient Descent can lead to overshooting the minimum of the loss function, causing the algorithm to oscillate or diverge rather than converge to the optimal solution. This can result in instability , slower convergence, or even failure to converge altogether. It may also cause the model to miss the minimum and prevent effective learning of the underlying patterns in the data.

1. **Can we use Logistics Regression for Classification of the Non-Linear data? If Not, Why?**

**Answer** – Logistics Regression is inherently designed for linear classification problems, making it less suitable for handling nonlinear data. Its decision boundary is a linear function, which limits its ability to capture complex nonlinear relationship present in certain dataset. In such case, more machines, Decision trees, or neural network are often preferred as they can learn and represent nonlinear patterns effectively.

1. **Differentiate between Adaboost and Gradient Boosting?**

**Answer** – Both Adaboost and Gradient Boosting are ensemble learning techniques, but they differ in their approach to building a strong predictive model.

1. Focuses on classification problems.
2. Assigns weights to the data points, emphasizing misclassified instances.
3. Sequentially trains weak learners with each iteration, adjusting weights to iteration.
4. Combines weak learners to form a strong model where more weight is given to those with lower training error.

Gradient Boosting :

1. Versatile and can be applied to both regression and classification problems.
2. Builds an ensemble of weak learners sequentiallt.
3. Each weak learner corrects the error of its predecessor by fitting to the residuals .
4. Employs a gradient descent optimization process to minimize the overall loss function.
5. **What is bias-variance trade off in machine learning?**

**Answer** – The bias -variance trade off in machine learning refers to the delicate balance between a models ability to capture the underlying patterns in the training data and its sensitivity to variations in the data’

. High Bias : A model with high bias oversimplifies the training data, often resulting in under fitting and poor performance on both the training and unseen data.

. High Variance : On the other hand, a model with high variance is overly complex, fitting the training data too closely and performing well on training data but poorly on new, data due but to an inability to generalize.

1. **Give short description each of Linear , RBF, Polynomial kernel used in SVM.**

**Answer –**

1. **Linear Kernel:**

**. Description:** The linear kernel is the simplest kernel for support vector machines ( SVM) . it represent a linear relationship between input features and is suitable for linearly separable dataset.

**. Use Case :** Appropriate when the data can be effectively separated by a straight

Line in the feature space.

1. **Radial Basis Function ( RBF ) Kernel :**

. Description: The RBF kernel, also known as the Gaussian kernel, measures the

Similarity between data point in a high – dimensional space.

**. Use Case :** Well- suited boundary is nonlinear or when dealing with data that

Might not be linearly separable.

1. **Polynomial Kernel :**

**. Description :** The polynomial kernel computes the similarity between data points

Based on polynomial function . it introduction nonlinearity and can

Capture more complex patterns in the data.

**. Ues Case :** Useful when the decision boundary is expected to be polynomial in

Nature, providing flexibility for handling moderately complex

Relationships in the data.