FLIP ROBO TECHNOLOGIES

MACHINE LEARNING ASSIGNMENT

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?

ANSWER --
R-squared is generally the better measure of goodness of fit when you want to understand how well your model explains the variability of the data.

R-squared is STANDARDIZED: It ranges from 0 to 1 (or 0% to 100%), making it easy to interpret and compare across different models and datasets. A higher R-squared indicates that the model explains a greater proportion of the variance in the dependent variable.
R-squared allows for EASIER COMPARISON: Since it is standardized, R-squared allows for straightforward comparison of the explanatory power of different models, even if they are applied to different datasets or have different numbers of predictors.
R-squared provides a clear measure of the proportion of VARIENCE explained: It directly tells you how much of the total variability in the dependent variable is explained by the model. For example, an R-squared of 0.75 means that 75% of the variance is explained by the model.
R-squared is often used in CONJUCTION with other metrics: Adjusted R-squared, which adjusts for the number of predictors in the model, is commonly used in model selection to balance fit and complexity.

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.

ANSWER --
TOTAL SUM OF SQUARES (TSS) :
TSS measures the total variation in the dependent variable(y) from its mean (y~).It quantifies the overall variability in the dataset.
Formula --
TSS = Sigma (yi - y`)^2
where :
yi is the actual value of the dependent variable for observation i.
y` is the mean of the dependent variable.
n is the number of observations.

EXPLAINED SUM OF SQUARES(ESS):
ESS measures the variation in the dependent variable that is explained by the regression model. It represents the reduction in variability due to the independent variables.
Formula --

ESS = Sigma(y^i - y`)^2
Where:
y^i is the predicted value of the dependent variable for observation i from the regression model.
y`is the mean of the dependent variable.

RESIDUAL SUM OF SQUARES(RSS):
RSS measures the variation in the dependent variable that remains unexplained by the regression model. It represents the error or residual variation.
Formula --
RSS = Sigma(yi - y^i)^2
Where:
yi is the actual value of the dependent variable for observation i
y^i is the predicted value of the dependent variable for observation i

Relationship Between TSS, ESS, and RSS :
These three metrics are related by the following equation:

TSS=ESS+RSS

Interpretation:
TSS represents the total variability in the data.
ESS represents the portion of that variability that is explained by the model.
RSS represents the portion of the variability that is not explained by the model (i.e., the error).

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

ANSWER --
Regularization is a crucial technique in machine learning used to prevent overfitting, improve model generalization, and manage model complexity. Here’s why regularization is needed:

1. Preventing Overfitting:
Overfitting occurs when a machine learning model learns not only the underlying pattern in the training data but also the noise and fluctuations that do not generalize to unseen data. This results in a model that performs well on the training data but poorly on new, unseen data.
Regularization techniques add a penalty to the model's complexity, discouraging it from fitting too closely to the training data. This helps the model generalize better to new data, reducing the likelihood of overfitting.

2. Controlling Model Complexity:
Complex models with a large number of parameters (e.g., deep neural networks, polynomial regression) have the capacity to fit the training data very closely. However, without regularization, these models may become too complex, capturing noise rather than the true underlying patterns.
Regularization introduces constraints on the model parameters, effectively simplifying the model and reducing its capacity to overfit. This helps in finding a

balance between model complexity and generalization.

3. Improving Generalization:
Generalization refers to a model's ability to perform well on new, unseen data.
Regularization improves generalization by preventing the model from becoming too specialized in the training data.
Better generalization means the model can make more accurate predictions on new data, which is the ultimate goal in most machine learning tasks.

4. Handling Multicollinearity:
Multicollinearity occurs when independent variables in a regression model are highly correlated. This can lead to unstable estimates of the model parameters and make the model sensitive to small changes in the data.
Regularization techniques like Ridge regression (L2 regularization) help in mitigating multicollinearity by shrinking the coefficients of correlated variables, leading to more stable and reliable models.

5. Sparse Solutions (Feature Selection):
Lasso regression (L1 regularization) encourages sparsity in the model by driving some coefficients to zero. This effectively performs feature selection, identifying the most important features while ignoring irrelevant ones.
Sparsity is particularly useful in high-dimensional datasets, where many features might be irrelevant or redundant. Regularization helps to reduce the model's reliance on these unnecessary features, leading to simpler and more interpretable models.

6. Reducing Variance:
High variance models are sensitive to small fluctuations in the training data, leading to large changes in predictions. This is often a sign of overfitting.
Regularization reduces variance by adding a penalty term to the loss function, which constrains the magnitude of the coefficients, leading to more stable and robust models.

4. What is Gini–impurity index?
ANSWER --

The Gini impurity index is a measure used to determine the best split in decision trees. It quantifies the "impurity" of a node, with lower values indicating more homogeneous nodes and better classification. It is a key metric in the construction of decision trees for classification tasks.

Low Gini impurity indicates that the node is mostly homogeneous (contains instances predominantly from one class), which is desirable for making decisions.

High Gini impurity indicates that the node is more heterogeneous (contains instances from multiple classes), which is less desirable.

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

ANSWER --
Yes, unregularized decision trees are prone to overfitting.

Unregularized decision trees are prone to overfitting because they tend to become overly complex, capturing noise and specific details of the training data rather than general patterns. Without regularization, a decision tree can grow too deep, leading to poor generalization and high variance, which negatively affects its performance on unseen data. To mitigate overfitting, it’s important to apply regularization techniques, such as limiting tree depth, pruning, or setting minimum sample sizes for splits.

6. What is an ensemble technique in machine learning?
ANSWER --

Ensemble techniques in machine learning involve combining multiple models to improve overall performance, often leading to more accurate, robust, and reliable predictions than any single model could achieve on its own. The core idea is that by aggregating the predictions of various models, the ensemble can correct the mistakes of individual models, reducing overfitting and increasing generalization.

Advantages of Ensemble Techniques:

Improved Accuracy: By combining the strengths of multiple models, ensembles typically achieve higher accuracy than individual models.
Robustness: Ensembles are less likely to be affected by the weaknesses or biases of any single model, leading to more stable and reliable predictions.
Reduction in Overfitting: Ensembles, particularly bagging techniques like Random Forests, can reduce the risk of overfitting by averaging out the noise and fluctuations captured by individual models.

7. What is the difference between Bagging and Boosting techniques?
ANSWER --
1. Purpose and Approach:
Bagging (Bootstrap Aggregating):

Purpose: Bagging aims to reduce variance and prevent overfitting by creating multiple versions of a model and combining them to produce a more stable and accurate prediction.

Approach: Bagging builds multiple models independently using different subsets of the training data. These subsets are generated through bootstrapping, which involves random sampling with replacement. The final prediction is typically made by averaging (in regression) or voting (in classification) across all the models.
Boosting:

Purpose:
Boosting focuses on reducing bias and variance by sequentially building models that correct the errors made by the previous models. It often turns weak learners into a

strong learner.
Approach:
Boosting builds models sequentially, with each new model focusing on the mistakes made by the previous ones. The models are not independent; each subsequent model is trained to correct the errors of its predecessors. The final prediction is typically a weighted sum of the predictions from all models.

2. Model Independence:
Bagging:
The models in bagging are trained independently of each other. Since they are built on different subsets of the data, they tend to make different errors, which helps reduce variance when their predictions are aggregated.

Boosting:
The models in boosting are trained sequentially and are dependent on each other. Each model is trained to correct the errors made by the previous models, leading to a cumulative improvement in performance.

3. Data Sampling:

Bagging:
Each model is trained on a different bootstrap sample (random sampling with replacement) of the original data. Some instances may be repeated in a sample, while others may be omitted.

Boosting:
Boosting uses the entire dataset for training each model, but it adjusts the weights of the data points based on the errors made by the previous models. Hard-to-predict instances are given higher weights so that subsequent models focus on them.
4. Model Weighting:

Bagging:
All models are given equal weight when making the final prediction. The aggregation is usually done by averaging (for regression) or majority voting (for classification).

Boosting:
Models are weighted based on their performance. Models that perform better (i.e., those that make fewer errors) are given higher weights in the final prediction.

5. Reducing Bias vs. Variance:
Bagging:
Primarily reduces variance. By averaging the predictions of multiple independent models, bagging smooths out fluctuations and reduces the model's sensitivity to noise in the training data.

Boosting:
Reduces both bias and variance. Boosting creates a strong learner by focusing on and correcting the errors of weak learners. This sequential learning helps to improve both the model's accuracy (reducing bias) and its generalization (reducing

variance).

6. Risk of Overfitting:

Bagging:
Less prone to overfitting compared to boosting, especially in high-variance models like decision trees. By aggregating multiple models, bagging stabilizes the predictions.

Boosting:
More prone to overfitting, especially if the model is too complex or the boosting process continues for too many iterations. This is because boosting focuses intensely on the hard-to-predict cases, which might include noise or outliers.

7. Examples:

Bagging:
Random Forests: A common implementation of bagging, where multiple decision trees are built on different bootstrap samples, and their predictions are averaged.

Boosting:
AdaBoost, Gradient Boosting Machines (GBM), XGBoost, LightGBM: These are popular boosting algorithms that build models sequentially, each focusing on the errors of the previous ones

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

The Out-of-Bag (OOB) error in Random Forests is an estimate of the model's prediction error calculated using the data points that were not included in the bootstrap samples used to train each tree. It provides a reliable and unbiased measure of model performance without the need for a separate test set, making it a valuable feature of Random Forests.

9. What is K-fold cross-validation?
ANSWER --

K-fold cross-validation is a powerful technique for model evaluation that helps to ensure that the model's performance is consistent and reliable. By dividing the dataset into K folds, training the model K times, and averaging the results, it provides a more comprehensive assessment of how well the model is likely to perform on unseen data.

Advantages of K-Fold Cross-Validation:

Better Generalization Estimate: By using multiple train-test splits, K-fold cross-validation provides a more robust estimate of a model’s ability to generalize to unseen data.

Efficient Use of Data: All data points are used for both training and validation, which is particularly beneficial when the dataset is small.
Reduced Variance: The averaging of results across K folds reduces the variance associated with a single random train-test split.

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

Hyperparameter tuning is a critical step in the machine learning workflow aimed at optimizing the hyperparameters of a model to enhance its performance, improve generalization, and effectively utilize resources. By carefully selecting the right hyperparameters through methods like grid search, random search, or Bayesian optimization, practitioners can significantly improve the predictive capabilities of their models.

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

Using a large learning rate in gradient descent can lead to several issues that negatively affect the training process and the model's performance. Here are the main problems associated with a large learning rate:

1. Divergence:
A large learning rate may cause the model parameters to update too aggressively, overshooting the optimal solution. Instead of converging towards the minimum of the loss function, the updates can lead to increasing loss values and cause the model to diverge.

2. Oscillation:
When the learning rate is too high, the updates can cause the model to oscillate around the minimum rather than settling down. This oscillation can prevent convergence and make it difficult for the model to find an optimal set of parameters.

3. Instability:
High learning rates can introduce instability in the training process, leading to erratic changes in the model parameters. This instability can manifest as erratic loss curves during training, making it challenging to monitor progress and determine when the model is improving.

4. Poor Generalization:
If the model fails to converge properly due to a large learning rate, it may not learn the underlying patterns in the data effectively. This can result in a model that performs poorly on both training and unseen data, leading to issues with generalization.

5. Loss Explosion:

In certain cases, particularly in deep learning, a large learning rate can lead to the "exploding gradient" problem, where gradients become excessively large during backpropagation. This can cause the loss function to explode, resulting in numerical instability and making it impossible to train the model effectively.

6. Missed Local Minima:
A large learning rate may cause the optimization process to skip over local minima or saddle points, preventing the model from settling into a good solution. This can hinder the model’s ability to achieve low loss and optimal performance.

7. Increased Training Time:
When the learning rate is too high, the model may take longer to converge, if it converges at all. As a result, more epochs or iterations may be required to achieve an acceptable level of performance, increasing overall training time.

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

Logistic regression is primarily designed for binary classification problems, where it models the probability that a given input belongs to a particular class. While it can be applied to some non-linear problems, it has limitations that make it less effective for complex non-linear datasets. Here's a closer look:

1. Linear Decision Boundary:
Basic Functionality: Logistic regression assumes a linear relationship between the input features and the log-odds of the target variable. It fits a linear decision boundary (hyperplane) to separate the classes.
Limitation for Non-Linear Data: If the underlying relationship between the features and the target is non-linear, a linear decision boundary may not adequately separate the classes, leading to poor classification performance.

2. Feature Transformation:
Non-linear Relationships: To use logistic regression effectively on non-linear data, you can apply feature transformations to create new features that capture the non-linear relationships. For instance, adding polynomial terms or interaction terms can help model non-linearities.
Increased Complexity: While this approach can make logistic regression suitable for some non-linear problems, it may also lead to increased complexity and risk of overfitting, especially if too many features are added.

3. Alternatives for Non-Linear Data:
More Suitable Models: For inherently non-linear data, it is often better to use models that are designed to handle non-linear relationships, such as:
Decision Trees: Capture non-linear relationships through hierarchical splits.
Random Forests: An ensemble of decision trees that improve robustness and performance.
Support Vector Machines (with non-linear kernels): Can create complex decision boundaries using kernel functions.

Neural Networks: Capable of modeling complex non-linear relationships through multiple layers and activation functions.

4. Conclusion:
While logistic regression can be adapted to handle some non-linear classification problems through feature transformations, it is fundamentally a linear model. For datasets with significant non-linear relationships, using more flexible models that inherently accommodate non-linearities is generally recommended for better performance.

13. Differentiate between Adaboost and Gradient Boosting.
ANSWER --

AdaBoost focuses on correcting misclassifications of previous learners and typically uses simple weak learners. It updates instance weights based on their classification results.

Gradient Boosting fits new learners to the residuals of previous models and optimizes a loss function, allowing for more complex models and flexibility in loss functions.
Both methods are powerful ensemble techniques, but the choice between them often depends on the specific problem, dataset characteristics, and desired performance.

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

The bias-variance trade-off is crucial in machine learning, as it helps understand the sources of error in predictive models. Striking the right balance between bias and variance is key to developing models that generalize well to new data, thus improving overall performance.

15. Give short description each of Linear, RBF, Polynomial kernels used in SVM.
ANSWER --
Support Vector Machines (SVM) can use different kernel functions to handle data that is not linearly separable. Here’s a brief description of three commonly used kernels: Linear, RBF (Radial Basis Function), and Polynomial kernels.

1. Linear Kernel:
Description: The linear kernel is the simplest kernel function. It computes the inner product of two vectors in the original feature space, effectively creating a linear decision boundary.

2. RBF Kernel (Radial Basis Function):
Description: The RBF kernel is a Gaussian kernel that maps input features into an infinite-dimensional space. It calculates the similarity between two points based on their distance in the feature space.

3. Polynomial Kernel:
Description: The polynomial kernel computes the similarity between two vectors based on a polynomial function. It allows for the modeling of interactions between features up to a specified degree.

```
-----------------------------------------------------------------------------------
-----------------------------------------------------------------------------------
```