MACHINE LEARNING

Batch DS2402

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?**

   R-squared is considered a better measure of goodness of fit in regression compared to the Residual Sum of Squares .
   - R-squared provides a measure of how well the independent variables explain the variation in the dependent variable.
   - expressed as a percentage, which makes it easier to interpret
   - higher R-squared value indicates a better fit of the model to the data
   - easier to compare across different models

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.**

- Total Sum of Squares (TSS): TSS represents the total variability in the dependent variable Y and is calculated as the sum of the squared differences between each observed dependent variable value and the overall mean of Y.
- Explained Sum of Squares (ESS): ESS represents the variability in the dependent variable (Y). It is also known as the regression sum of squares and is calculated as the sum of the squared differences between the predicted values of Y and the overall mean of Y.
- Residual Sum of Squares (RSS): RSS represents the unexplained variability in the dependent variable (Y). It is calculated as the sum of the squared differences between the observed values of Y (Y_i) and the predicted values of Y ($\hat{Y}_i$).

$$TSS=ESS+RSS$$

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

- Preventing Overfitting: Overfitting occurs when a model learns to fit the training data too closely, capturing noise and random fluctuations in the data rather than underlying patterns. Regularization techniques penalize complex models or large coefficients, discouraging them from fitting the noise in the training data and promoting simpler models that generalize better to unseen data.
- Handling High Dimensionality: In high-dimensional spaces, where the number of features is large compared to the number of samples, models are more prone to overfitting. Regularization helps to control the growth of coefficients associated with each feature, preventing the model from becoming overly complex and improving its ability to generalize.

- Dealing with Collinearity: Collinearity occurs when two or more variables are highly correlated. In such cases, the coefficients estimated by the model can become unstable, leading to overfitting. Regularization techniques, such as Ridge regression, help to stabilize the coefficients by penalizing large parameter values, making the model more robust to collinearity.
- Improving Model Interpretability: Regularization techniques can also improve the interpretability of machine learning models by shrinking or eliminating irrelevant features.
- Balancing Bias and Variance: Regularization helps to strike a balance between bias and variance in machine learning models. By controlling the complexity of the model through regularization parameters.

4. **What is Gini–impurity index?**

   The Gini impurity index is a measure used in decision tree algorithms, particularly in classification tasks, to evaluate the purity of a set of data points, it quantifies how often a randomly chosen element from the dataset would be incorrectly classified if it were randomly labeled according to the distribution of labels in the subset.

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

   Yes, unregularized decision trees are prone to overfitting:

- High Model Complexity: Decision trees have the ability to create very complex decision boundaries by partitioning the feature space into finer and finer regions based on the training data. Without any constraints or regularization, decision trees can continue to grow deeper and more complex.
- Memorization of Training Data: Unregularized decision trees have the capacity to memorize the training data, especially when the tree depth is not limited. This means that the model can essentially "learn by heart" the specific examples in the training set.
- Sensitive to Small Changes: Decision trees are sensitive to small changes in the training data. Without regularization, they can split the data too finely, resulting in overly complex models that are highly sensitive to noise or minor fluctuations in the training set.
- Lack of Generalization: Overfit decision trees may perform extremely well on the training data but fail to generalize to new, unseen data. This is because they have learned to capture the noise and idiosyncrasies of the training set rather than the true underlying patterns that are present in the data.

6.What is an ensemble technique in machine learning?

An ensemble technique in machine learning is a method that combines multiple individual models to create a stronger and more robust predictive model. The idea behind ensemble techniques is to leverage the diversity of individual models to improve predictive performance, generalization, and robustness.

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

Bagging:

- involves training multiple instances of the same learning algorithm (e.g., decision trees) on different subsets of the training data, often sampled with replacement (bootstrap samples). Each base learner is trained independently.
- base learners are trained independently of each other, in parallel. Each base learner is trained on a random subset of the training data
- assigns equal weight to all examples in the training data
- the final prediction is obtained by averaging or voting over the predictions of all base learners, which are typically independent and trained on different subsets of the data

Boosting:

- trains a sequence of weak learners (models that are slightly better than random guessing) sequentially. Each subsequent learner focuses more on the examples that were misclassified by the previous learners
- The training process is iterative, and each base learner is trained to correct the errors made by the previous ones.
- assigns weights to the training examples, with more weight given to examples that were misclassified by the previous learners
- the final prediction is obtained by combining the predictions of all base learners, with more weight given to the predictions of the more accurate learners

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

In Random Forests, the out-of-bag (OOB) error is an estimate of the model's prediction error on unseen data. It is computed using the samples that were not included in the training of each individual decision tree

9. What is K-fold cross-validation?

K-fold cross-validation is a popular technique used in machine learning for assessing the performance of a predictive model

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

Hyperparameter tuning, also known as hyperparameter optimization or model selection, is the process of finding the optimal set of hyperparameters for a machine learning model. Hyperparameters are configuration settings that are not learned from the data but are set before the training process begins. They control aspects of the learning algorithm's behavior and directly affect the performance and complexity of the model.

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

- Overshooting the Minimum: When the learning rate is too large, the algorithm may take excessively large steps in the direction of the gradient, causing it to overshoot the minimum of the loss function. This can result in the algorithm diverging or oscillating around the minimum, rather than converging to it.
- Instability and Divergence: A large learning rate can lead to instability in the optimization process, causing the algorithm to diverge or fail to converge to a solution. This instability is often characterized by erratic behavior of the optimization trajectory, with the loss function increasing instead of decreasing over time.
- Inaccurate Parameter Updates: With a large learning rate, the parameter updates can be too large, causing the algorithm to miss the optimal solution or jump over it entirely. This can result in poor convergence and suboptimal performance of the model.
- Sensitivity to Noise and Variability: Large learning rates can make the optimization process more sensitive to noise and variability in the data, leading to erratic behavior and poor generalization to new, unseen data. This can result in overfitting and reduced performance on the test set.
- Slow Convergence: Paradoxically, although large learning rates may lead to initially faster convergence, they can hinder the overall convergence process by causing the algorithm to oscillate or diverge. As a result, the optimization process may take longer to converge to a satisfactory solution compared to using a smaller, more appropriate learning rate.

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

Logistic Regression is a linear classification algorithm, which means it assumes a linear relationship between the input features and the output (class labels). Therefore, Logistic Regression is inherently limited to modeling linear decision boundaries.

13. Differentiate between Adaboost and Gradient Boosting.

Adaboost

- combines multiple weak learners (e.g., decision trees) sequentially. In each iteration, Adaboost adjusts the weights of incorrectly classified examples to focus more on the difficult examples in subsequent iterations.
- assigns higher weights to incorrectly classified examples in each iteration, focusing more on the difficult examples that are misclassified by the previous weak learners.
- fits each weak learner to the entire dataset in each iteration but adjusts the weights of examples to focus more on the difficult examples.

Gradient Boosting

- sequentially trains weak learners to minimize a loss function by using gradient descent optimization
- fits each new weak learner to the residual errors of the previous ones, effectively reducing the errors made by the ensemble model in each iteration.
- It is inherently sequential because each weak learner depends on the output of the previous ones

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

The bias-variance tradeoff is a fundamental concept in machine learning that describes the tradeoff between the bias and variance of a model and its overall prediction error. It highlights the delicate balance between model complexity and the ability to accurately capture the underlying patterns in the data

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

Linear Kernel:

- The linear kernel is the simplest kernel used in SVMs.
- It represents the dot product between the input features in the original feature space.
- The decision boundary created by the linear kernel is a straight line (or hyperplane in higher dimensions).
- It is suitable for linearly separable datasets or when the decision boundary is expected to be linear.

RBF (Radial Basis Function) Kernel:

- The RBF kernel is a popular non-linear kernel used in SVMs.
- It maps the input features into a higher-dimensional space using a Gaussian (radial basis) function.
- The decision boundary created by the RBF kernel is non-linear and can capture complex patterns in the data.
- It is suitable for datasets with non-linear decision boundaries or when the relationship between the features and the target variable is non-linear.

Polynomial Kernel:

- The polynomial kernel is another non-linear kernel used in SVMs.
- It maps the input features into a higher-dimensional space using polynomial functions.
- The decision boundary created by the polynomial kernel is non-linear and can capture polynomial relationships between the features.
- The degree of the polynomial kernel determines the complexity of the decision boundary; higher degrees can capture more complex patterns but may also lead to overfitting.
- It is suitable for datasets with non-linear decision boundaries, and the choice of the polynomial degree depends on the complexity of the data.