Batch No.: DS2401
Machine Learning Assignment-5: Answers

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?

Both R-squared and Residual Sum of Squares are important and needs to be evaluated. But R-squared is better than RSS because it is more understandable and we can easily compare among all models. Since the value of R-squared lies between 0 and 1, hence by looking at the value we can understand the amount of percentage the model is working. Whereas RSS is useful for assessing the overall goodness of fit of the model comparing 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.

TSS (Total Sum of Squares): The total sum of squares (TSS) is the sum of squared differences between the overall mean of the dependent variables from the observed dependent variables.

ESS (Explained Sum of Squares): The ESS (Explained Sum of Squares) is the sum of the squared differences between the mean of the dependent variables from predicted values.

RSS (Residual Sum of Squares): The RSS (Residual Sum of Squares) is the sum of the squared differences between predicted values from observed dependent variables.

Second the equation relating to these three metrices is:

TSS (Total Sum of Squares) = ESS (Explained Sum of Squares) + RSS (Residual Sum of Squares)

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

During the process of Machine Learning, model can be overfitted or underfitted. To avoid the issue of overfitting and underfitting we need to use regularization and help us get an optimal model.

4. What is Gini–impurity index?

Gini impurity index is a measure used in decision tree to assess the impurity or disorder evaluate the impurity or disorder of a dataset. It helps decision trees choose the best way to split the data into smaller groups when making predictions. The highest Gini impurity is 0.5. This indicates that a node has an equal distribution of classes, meaning that it is completely impure.

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

Unregularized decision-trees are very much prone to overfitting. It means they perform very well on the training data they were built on but performs poorly on unseen data.

Reasons for overfitting are as follows:

1. **Greedy nature**: Decision trees grow recursively by choosing the best split at each node to minimize impurity.
2. **Large trees**: Unregulated trees can grow arbitrarily large.
3. **Data dependence**: Without regulations, the tree structure becomes highly dependent on the specifics of the training data. Small changes in the data can lead to drastically different trees, impacting performance on unseen data.

6. What is an ensemble technique in machine learning?

In machine learning, ensemble technique is a method that combines predictions from multiple individual machine learning models to produce a more robust and accurate prediction than any single model alone. The idea is that by combining the predictions of multiple models, the strengths of each individual model can be leveraged while reducing their weaknesses, thereby improving overall performance.

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

Bagging and Boosting are both popular ensemble techniques in machine learning, but they differ in their approach and goals.

Bagging	Boosting
Trains multiple independent models on different subsets of the training data by sampling with replacement. Each subset is used to train a separate base model independently.	Trains models sequentially. Each model focuses on correcting the errors made by the previous model. This means later models are "boosted" to learn from the weaknesses of earlier ones.
Combines the individual predictions by averaging (for regression) or voting (for classification).	Combines the individual predictions using a weighted sum, where the weights depend on the performance of each model.
Simpler to implement, improves model stability and reduces overfitting.	Can achieve significant performance improvements, effective for complex problems, addresses both variance and bias.
May not effectively address bias. Also, may not achieve significant performance gains compared to individual models	Boosting can be more prone to overfitting if not properly tuned or if the base models are too complex.

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

Out-of-bag (OOB) error is a way to measure the prediction error of a random forest. It is calculated using the samples in the training data that were not included in the bootstrap sample used to train each individual decision tree in the forest. In other words, the out-of-bag (OOB) error in Random Forests is like having a built-in test set for each tree.

9. What is K-fold cross-validation?

K-fold cross-validation is a technique used to assess the performance of a machine learning model. We split the dataset into K no. of subsets (also known as "folds") of approximate equal size. The model is then trained and evaluated on K-1 subset leaving one subset for testing of the model. We iterate k-times with different subsets with each time leaving 1 subset for testing.

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

Hyperparameter tuning is the process of finding the best set of hyperparameters for a machine learning model. Hyperparameters are parameters that are set before the learning process begins and control the aspects of the learning process itself. Unlike model parameters, which are learned from the training data, hyperparameters are external to the model and must be specified.

Hyperparameter tuning is done for several important reasons:

1. Optimizing Model Performance
2. Preventing Overfitting:
3. Improving Generalization
4. Enhancing Model Interpretability
5. Increasing Efficiency

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

Having a large learning rate in gradient descent can lead to several issues as follows:

1. Having a large learning rate, the updates to the model parameters can be too large, causing the optimization algorithm may overshoot the minimum.
2. A learning rate that is too large can slow down the convergence of the optimization algorithm which can make the algorithm computationally expensive.

3. Large learning rates can result in unstable behavior of the optimization process.
4. Large learning rates make it challenging to fine-tune the optimization process. It can be difficult to find a learning rate that balances between convergence speed and stability.

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

Logistic regression is known for its simplicity and interpretability in linear classification, it can't directly handle non-linear data for the below mentioned reasons:

1. Linear Decision Boundary: Logistic Regression assumes a linear decision boundary means it can only separate classes using a straight line in two dimensions. If the relationship between the features and the target variable is non-linear it requires more complex boundaries like curves or hyperplanes that a straight line can't represent.
2. Logistic Regression cannot apprehend the non-linear relationships between features that may exist in the data. It's not capable of capturing interactions between features that are not linear.
3. When Logistic Regression is applied to nonlinear data, it may lead to underfitting where the model is too simple to capture the non-linear patterns in the data. This results in poor predictive performance and low accuracy.

13. Differentiate between Adaboost and Gradient Boosting.

Adaboost and Gradient Boosting are both ensemble learning techniques used for improving the performance of machine learning models, particularly decision trees. While they share the same basic principle, they differ in their approach:

Adaboost	**Gradient Boosting**
Higher weights given to data points misclassified by previous learners.	Each learner is trained to correct the errors made by the previous ones. It uses the gradient of the loss function to guide the training process.
Each weak learner is trained on the entire dataset with weights assigned to the training instances based on their difficulty	Each weak learner is trained on the errors made by the previous learners. It fits each new learner to the residuals of the previous model
Each weak learner is given a weight based on its performance. The final prediction is a weighted sum of the predictions made by all the weak learners.	Each weak learner is given a weight based on its contribution to minimizing the loss function. The final prediction is the sum of the predictions made by all the weak learners.
AdaBoost does not explicitly use a learning rate parameter	Gradient Boosting uses a learning rate parameter to control the contribution of each weak learner to the final model

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

The bias-variance trade-off is a concept in machine learning that describes the relationship between a model's bias, variance and overall error.

Bias is the difference between the prediction of values by machine learning model and the correct value. High bias leads to underfitting meaning the model performs poorly on both training and unseen data.

Variance is the overfitting of a model. It measures how much the predictions of the model vary across different training sets. High variance models tend to be overly complex and may overfit the training data.

The bias-variance trade-off means that as we decrease the bias of the model by increasing its complexity we increase its variance and vice versa. Finding the right balance between bias and variance is essential for building models that performs well on unseen data and avoid overfitting or underfitting.

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

Linear Kernel: It calculates the dot product between two input vectors. Suitable for linearly separable data where classes can be separated by a straight line. It works well when the relationship between features and labels is approximately linear.

RBF (Radial Basis Function): It transforms data into a high-dimensional space where it might become linearly separable. Captures a wide range of decision boundaries, including non-linear ones. Suitable for complex, non-linear data distributions.

Polynomial: The polynomial kernel is another non-linear kernel used in SVM. It uses a polynomial function to map the data into a higher-dimensional space. It is done by taking the dot product of the data points in the original space and the polynomial function in the new space.