MACHINE LEARNING

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 of goodness of fit model in regression and why?

Both **R-squared** and **Residual Sum of Squares (RSS)** are measures of goodness of fit in regression models. However, they measure different aspects of the model.

R-squared is a statistical measure that represents the proportion of the variance in the dependent variable that is explained by the independent variable(s). It ranges from 0 to 1, with higher values indicating a better fit of the model to the data. R-squared is a useful metric for comparing different models, as it provides a standardized measure of goodness of fit [1].

On the other hand, **Residual Sum of Squares (RSS)** is a measure of the difference between the observed values of the dependent variable and the predicted values of the dependent variable. It is calculated as the sum of the squared residuals, which are the differences between the observed and predicted values of the dependent variable. A lower value of RSS indicates a better fit of the model to the data [2][3].

Therefore, both R-squared and RSS are useful measures of goodness of fit, but they measure different aspects of the model. R-squared is useful for comparing different models, while RSS is useful for evaluating the fit of a single model [4].

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): It is the sum of squared differences between the observed dependent variables and the overall mean. Mathematically, it can be expressed as:

$TSS=i=1\sum n(yi-y^{-})2$

where yi is the observed dependent variable and y^{-} is the mean of the dependent variable.

Explained Sum of Squares (ESS): It is the sum of the differences between the predicted value and the mean of the dependent variable. In other words, it describes how well our line fits the data. Mathematically, it can be expressed as:

$$ESS = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

where yi^ is the predicted value of the dependent variable.

Residual Sum of Squares (RSS): It is the difference between the observed and predicted values. Mathematically, it can be expressed as:

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where yi is the observed dependent variable and yi^ is the predicted value of the dependent variable.

The formula that relates these three metrics is:

$$TSS = ESS + RSS$$

This formula just says that the total variability equals what is explained by a regression model plus what is left unexplained (the residual). The best fit for a regression line is the one that minimizes the RSS, which is called the ordinary least square (OLS) [1]

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

Regularization is a technique used to reduce errors by fitting the function appropriately on the given training set and avoiding overfitting

4.What is Gini–impurity index?

The **Gini-impurity index** is a measure of the **impurity** or **randomness** of a dataset used in **decision tree learning**. It is also known as the **Gini index** or **Gini impurity**. The Gini index is calculated by subtracting the sum of the squared probabilities of each class from one. The resulting value lies between 0 and 1, with 0 indicating that the dataset is completely pure and 1 indicating that the dataset is completely impure [1].

In decision tree learning, the Gini index is used to determine the best attribute to split the data into subsets. The attribute with the lowest Gini index is chosen as the splitting criterion, as it results in the most homogeneous subsets [1].

The Gini index is similar to another measure of impurity called **entropy**. While entropy is based on the concept of information theory, the Gini index is based on the concept of probability theory [1].

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

Yes, unregularized decision-trees are prone to overfitting. Decision trees are prone to overfitting when there is no regularization. Regularization is the process of adding constraints to the model to prevent overfitting. Without regularization, the decision tree can become too complex, resulting in overfitting [1]. When a decision tree is left on its own without regularization, the tree will continue to fit until each data point is a different leaf in the tree. This will not generalize well and will overfit the model [2].

In summary, unregularized decision-trees are prone to overfitting because they can become too complex without any constraints, resulting in poor generalization performance on new unseen data [12].

6.What is an ensemble technique in machine learning?

An **ensemble technique** in machine learning is a method of combining multiple models to improve the overall performance of the learning system[1]. The idea behind ensemble models is to leverage the collective intelligence of multiple models to mitigate errors or biases that may exist in individual models

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

Bagging involves fitting many models on different samples of the dataset and averaging the predictions, whereas boosting involves adding ensemble members sequentially to correct the predictions made by prior models and outputs a weighted average of the predictions.
Bagging is a strategy for minimizing prediction variance that produces additional data for training from a dataset, while boosting is an iterative technique for modifying the weight of an observation in accordance with the previous classification.
Bagging decreases variance, not bias, and solves over-fitting issues in a model, while boosting decreases bias, not variance.
Boosting adjusts the training set's distribution based on the performance of previously created classifiers, while bagging changes it randomly.

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

The out-of-bag error is the average error for each predicted outcome calculated using predictions from the trees that do not contain that data point in their respective bootstrap sample. This way, the Random Forest model is constantly being validated while being trained

9.What is K-fold cross-validation?

K-fold cross-validation is a technique for evaluating predictive models. The dataset is divided into k subsets or folds. The model is trained and evaluated k times, using a different fold as the validation set each time. Performance metrics from each fold are averaged to estimate the model's generalization performance.

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

Hyperparameter tuning is an essential part of controlling the behavior of a machine learning model. If we don't correctly tune our hyperparameters, our estimated model parameters produce suboptimal results, as they don't minimize the loss function. This means our model makes more errors

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

If the step size is too large, however, we may never converge to a local minimum because we overshoot it every time. If we are lucky and the algorithm converges anyway, it still might take more steps than it needed. Large step size converges slowly

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

Logistic regression has traditionally been used to come up with a hyperplane that separates the feature space into classes. But if we suspect that the decision boundary is nonlinear we may get better results by attempting some nonlinear functional forms for the logit function.

13.Differentiate between Adaboost and Gradient Boosting.

AdaBoost and **Gradient Boosting** are both **boosting algorithms** that are used to improve the accuracy of machine learning models. However, they differ in their approach to boosting.

AdaBoost is a boosting algorithm that works by **iteratively training weak learners** on the same dataset. In each iteration, the algorithm assigns higher weights to the misclassified samples from the previous iteration

and lower weights to the correctly classified samples. This way, the algorithm focuses on the samples that are difficult to classify and tries to improve the accuracy of the model by giving more importance to these samples.

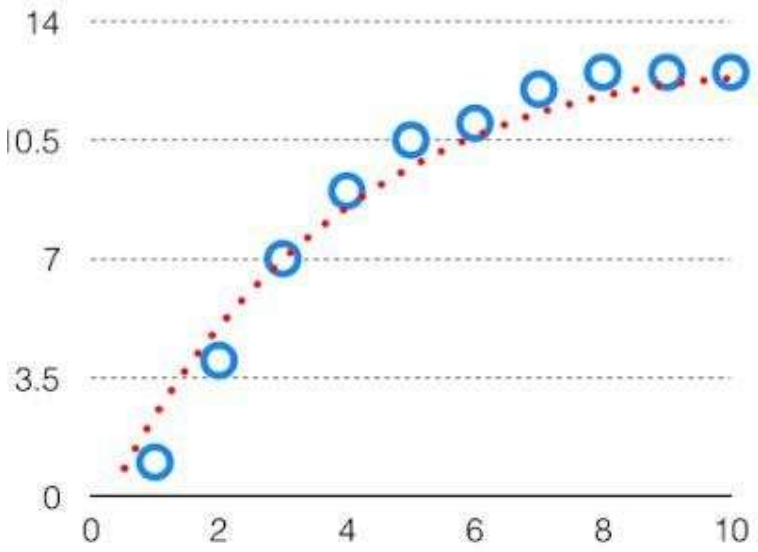
Gradient Boosting, on the other hand, is a boosting algorithm that works by **iteratively training weak learners** on the **residuals** of the previous weak learner. In each iteration, the algorithm tries to fit a new weak learner to the residuals of the previous weak learner. This way, the algorithm focuses on the errors made by the previous weak learner and tries to improve the accuracy of the model by correcting these errors.

In summary, while both AdaBoost and Gradient Boosting are boosting algorithms that work by iteratively training weak learners, they differ in their approach to boosting. AdaBoost assigns higher weights to the misclassified samples from the previous iteration, while Gradient Boosting tries to fit a new weak learner to the residuals of the previous weak learner.

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

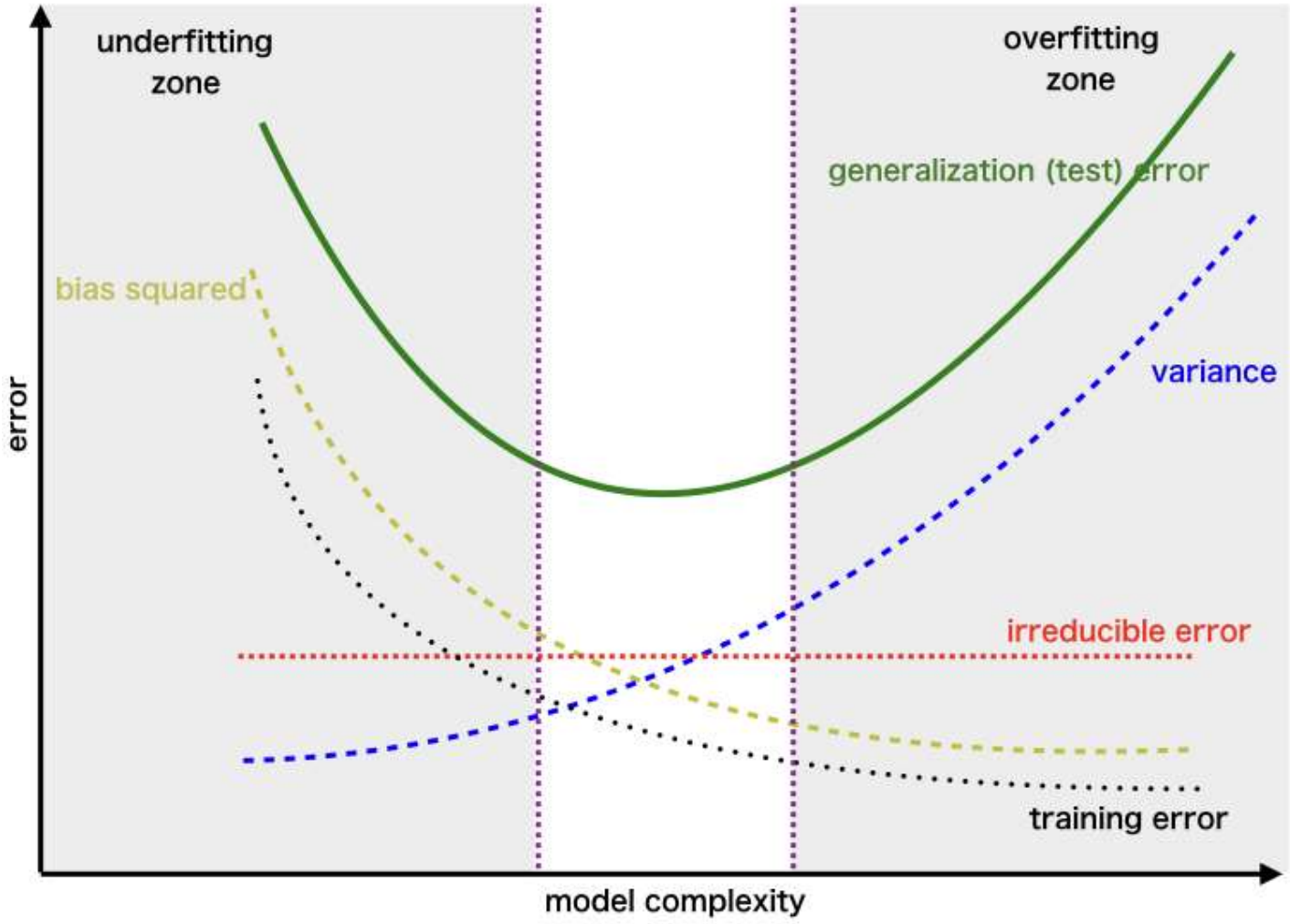
Bias Variance Tradeoff

If the algorithm is too simple (hypothesis with linear equation) then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex (hypothesis with high degree equation) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as a Trade-off or Bias Variance Trade-off. This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time. For the graph, the perfect tradeoff will be like this.



We try to optimize the value of the total error for the model by using the Bias-Variance Tradeoff.

The best fit will be given by the hypothesis on the tradeoff point. The error to complexity graph to show trade-off is given as –



Region for the Least Value of Total Error

This is referred to as the best point chosen for the training of the algorithm which gives low error in training as well as testing data.

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

- **Linear kernel**: This kernel is used when the data is linearly separable, meaning that a straight line can separate the classes. The linear kernel is simply the dot product of the input vectors:

$K(X,Y)=XTY$

- **Polynomial kernel**: This kernel is used when the data is not linearly separable, but can be separated by a polynomial function of some degree. The polynomial kernel is the dot product of the input vectors raised to a power, plus a constant term:

$K(X,Y)=(\gamma \cdot XTY+r)d,\gamma>0$

- **Radial Basis Function (RBF) kernel**: This kernel is also used when the data is not linearly separable, but can be separated by a nonlinear function that depends on the distance between the input vectors. The RBF kernel is the exponential of the negative squared distance between the input vectors:

$K(X,Y)=\exp(-\gamma \cdot \|X-Y\|2),\gamma>0$

FLIP ROBO