**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?**

Ans:- R-squared is generally a better measure of the goodness of fit for a regression model than the residual sum of squares (RSS).

R-squared, denoted as \(R^2\), is a statistical measure that represents the proportion of the variance for the dependent variable that's explained by the independent variables in the model. It is dimensionless and ranges from 0 to 1, where a value closer to 1 indicates a better fit. \(R^2\) is calculated using the formula:

\[ R^2 = 1 - (RSS)/(TSS) \]

where \(RSS\) is the residual sum of squares, and \(TSS\) is the total sum of squares. \(TSS\) represents the total variance in the dependent variable.

The RSS, on the other hand, is the sum of the squared differences between the observed actual outcomes and the outcomes predicted by the regression model. It is calculated as:

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

where\(y_i\) is the actual value and \(\hat{y}_i\) is the predicted value from the model for the \(i\)-th observation.

The reason why \(R^2\) is often preferred over RSS as a measure of goodness of fit is due to its standardized nature:

1. Scalability: \(R^2\) is scale-invariant, meaning it does not change if the scale of the data changes, whereas RSS is affected by the scale of the dependent variable. This makes \(R^2\) a better choice when comparing models fitted on different scales.

2. Interpretability: \(R^2\) has an intuitive interpretation as the proportion of variance explained, which is easier to understand than the sum of squared residuals. An \(R^2\) of 0.75 means that 75% of the variance in the dependent variable is explained by the model, which is a straightforward interpretation.

3. Benchmarking: \(R^2\) provides a clear benchmark. An \(R^2\) of 0 indicates that the model explains none of the variability in the response data around its mean, while an \(R^2\) of 1 indicates that the model explains all the variability.

4. Adjustment for model complexity: Adjusted \(R^2\) takes into account the number of predictors in the model, which helps in assessing whether the addition of a new predictor really improves the model or is just adding complexity without significantly improving the fit.

**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.**

Ans :-

The **sum of squares total** **(SST)**or the **total sum of squares (TSS)** is the sum of squared differences between the observed dependent variables and the overall **mean**. Think of it as the dispersion of the observed variables around the [**mean**](https://365datascience.com/tutorials/statistics-tutorials/measures-central-tendency/)—similar to the [variance](https://365datascience.com/tutorials/statistics-tutorials/coefficient-variation-variance-standard-deviation/) in descriptive statistics.

Mathematically, the difference between variance and SST is that we adjust for the degree of freedom by dividing by n–1 in the variance formula.

SST=n∑i=1(yi−¯y)2𝑆𝑆𝑇=∑𝑖=1𝑛(𝑦𝑖−𝑦¯)2

Where:

yi 𝑦𝑖  – observed dependent variable

¯y 𝑦¯  – mean of the dependent variable

SSR:- The **sum of squares due to regression** **(SSR)** or **explained sum of squares** **(ESS)**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.

The SSR formula is the following:

SSR=n∑i=1(^yi−¯y)2𝑆𝑆𝑅=∑𝑖=1𝑛(𝑦^𝑖−𝑦¯)2

Where:

^yi 𝑦^𝑖  – the predicted value of the dependent variable

¯y 𝑦¯  – mean of the dependent variable

If **SSR** equals **SST**, our **regression** **model** perfectly captures all the observed variability, but that’s rarely the case.

SSE:- The **sum of squares error** **(SSE) or residual sum of squares (RSS, where** residual means remaining or unexplained**)** is the difference between the observed and predicted values.

The SSE calculation uses the following formula:

SSE=n∑i=1ε2i𝑆𝑆𝐸=∑𝑖=1𝑛𝜀𝑖2

Where εi𝜀𝑖 is the difference between the actual value of the dependent variable and the predicted value:

εi= yi−^yi𝜀𝑖= 𝑦𝑖−𝑦^𝑖

Regression analysis aims to [minimize the SSE](https://365datascience.com/tutorials/statistics-tutorials/ols-assumptions/)—the smaller the error, the better the **regression’s** estimation power.

Mathematically, **SST** = **SSR** + **SSE**.

The rationale is the following:

The total variability of the dataset is equal to the variability explained by the **regression line** plus the unexplained variability, known as error.

Given a constant total variability, a lower error means a better **regression** model. Conversely, a higher error means a less robust **regression**. And that’s valid regardless of the notation you use.

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

Ans:-

Regularization in machine learning serves as a method to forestall a model from overfitting. Overfitting transpires when a model not only discerns the inherent pattern within the training data but also incorporates the noise, potentially leading to subpar performance on fresh, unobserved data. The employment of regularization aids in mitigating this issue by augmenting a penalty to the loss function employed for [model training](https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-steps). Here are the key points about regularization:

1. Purpose: The primary goal of regularization is to reduce the model's complexity to make it more generalizable to new data, thus improving its performance on unseen datasets.

2. Methods: There are several types of regularization techniques commonly used:

* + L1 Regularization (Lasso): This adds a penalty equal to the absolute value of the magnitude of coefficients. This can lead to some coefficients being zero, which means the model ignores the corresponding features. It is useful for feature selection.
  + L2 Regularization (Ridge): Adds a penalty equal to the square of the magnitude of coefficients. All coefficients are shrunk by the same factor, and none are eliminated, as in L1.
  + Elastic Net: This combination of L1 and L2 regularization controls the model by adding penalties from both L1 and L2, which can be a useful middle ground.

3. Impact on Loss Function: Regularization modifies the loss function by adding a regularization term.

4. Choice of Regularization Parameter: The choice of λ (also known as the regularization parameter) is crucial. It is typically chosen via cross-validation to balance fitting the training [data](https://www.simplilearn.com/what-is-data-article) well and keeping the model simple enough to perform well on new data.

**4. What is Gini–impurity index?**

Ans:- Gini Impurity measures how well does a node splits the data set between the two outcomes. It aims to reduce the impurity score from the root node of the tree to the leaf node. The lower the score, the better the split is, as we have seen in our example a Gini Index of 0 denotes a pure node(all data points belong to one class).

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

Ans:- Decision trees, by their very nature, are prone to overfitting, especially when they are deep. Overfitting occurs when a model captures noise or fluctuations in the training data that do not represent the underlying data distribution. In the context of decision trees, overfitting can mean creating too many branches based on outliers or anomalies in the training data.

A tree that is too complex might achieve a perfect accuracy score on the training data but perform poorly on new, unseen data. Such a tree has low bias but high variance, and its predictions can be unstable.

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

Ans:- Ensemble learning refers to a machine learning approach where several models are trained to address a common problem, and their predictions are combined to enhance the overall performance. The idea behind ensemble learning is that by combining multiple models, each with its strengths and weaknesses, the ensemble can achieve better results than any single model alone. Ensemble learning can be applied to various machine learning tasks, including classification, regression, and clustering. Some common ensemble learning methods include bagging, boosting, and [stacking](https://www.simplilearn.com/tutorials/data-structure-tutorial/stacks-in-data-structures).

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

### Ans:- Differences Between Bagging and Boosting

| S.NO | Bagging | Boosting |
| --- | --- | --- |
| 1. | The simplest way of combining predictions that  belong to the same type. | A way of combining predictions that  belong to the different types. |
| 2. | Aim to decrease variance, not bias. | Aim to decrease bias, not variance. |
| 3. | Each model receives equal weight. | Models are weighted according to their performance. |
| 4. | Each model is built independently. | New models are influenced  by the performance of previously built models. |
| 5. | Different training data subsets are selected using row sampling with replacement and random sampling methods from the entire training dataset. | Every new subset contains the elements that were misclassified by previous models. |
| 6. | Bagging tries to solve the over-fitting problem. | Boosting tries to reduce bias. |
| 7. | If the classifier is unstable (high variance), then apply bagging. | If the classifier is stable and simple (high bias) the apply boosting. |
| 8. | In this base classifiers are trained parallelly. | In this base classifiers are trained sequentially. |
| 9 | Example: The Random forest model uses Bagging. | Example: The AdaBoost uses Boosting techniques |

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

Ans:- Generally, in machine learning and data science, it is crucial to create a trustful system that will work well with the new, unseen data. Overall, there are a lot of different approaches and methods to achieve this generalization. Out-of-bag error is one of these methods for validating the machine learning model.

### 3.1. Definition

This approach utilizes the usage of bootstrapping in the random forest. Since the bootstrapping samples the data with the possibility of selecting one sample multiple times, it is very likely that we won’t select all the samples from the original data set. Therefore, one smart decision would be to exploit somehow these unselected samples, called out-of-bag samples.

Correspondingly, the error achieved on these samples is called out-of-bag error. What we can do is to use out-of-bag samples for each decision tree to measure its performance. This strategy provides reliable results in comparison to other validation techniques such as train-test split or cross-validation.

### 3.2. Probability of Out-of-bag Sample

Theoretically, with the quite big data set and the number of sampling, it is expected that out-of-bag error will be calculated on 36% of the training set. To prove this, consider that our training set has  samples. Then, the probability of selecting one particular sample from the training set is .

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

Ans:- Cross-validation is a statistical method used to estimate the skill of machine learning models.

It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

In this tutorial, you will discover a gentle introduction to the k-fold cross-validation procedure for estimating the skill of machine learning models.

After completing this tutorial, you will know:

* That k-fold cross validation is a procedure used to estimate the skill of the model on new data.
* There are common tactics that you can use to select the value of k for your dataset.
* There are commonly used variations on cross-validation such as stratified and repeated that are available in scikit-learn.

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

Ans:- In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), **hyperparameter optimization**[[1]](https://en.wikipedia.org/wiki/Hyperparameter_optimization#cite_note-1) or tuning is the problem of choosing a set of optimal [hyperparameters](https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)) for a learning algorithm. A hyperparameter is a [parameter](https://en.wikipedia.org/wiki/Parameter) whose value is used to control the learning process.

Hyperparameter optimization finds a tuple of hyperparameters that yields an optimal model which minimizes a predefined [loss function](https://en.wikipedia.org/wiki/Loss_function) on given independent data.[[2]](https://en.wikipedia.org/wiki/Hyperparameter_optimization#cite_note-abs1502.02127-2) The objective function takes a tuple of hyperparameters and returns the associated loss.[[2]](https://en.wikipedia.org/wiki/Hyperparameter_optimization#cite_note-abs1502.02127-2) [Cross-validation](https://en.wikipedia.org/wiki/Cross-validation_(statistics)) is often used to estimate this generalization performance, and therefore choose the set of values for hyperparameters that maximize it.[[3]](https://en.wikipedia.org/wiki/Hyperparameter_optimization#cite_note-bergstra-3)

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

Ans:- However, a high learning rate also has some drawbacks, such as causing instability, oscillations, divergence, and missing the optimal solution. A high learning rate in gradient descent implies that the network takes significant strides in the weight space, striving to swiftly reach the error function's global minimum.

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

Ans:- Logistic regression is simple and easy to implement, but it also has some drawbacks. One of them is that it assumes a linear relationship between the input features and the output. This means that it cannot capture the complexity and non-linearity of the data.

**13. Differentiate between Adaboost and Gradient Boosting.**

Ans:- Boosting and AdaBoost are both ensemble machine learning techniques that combine multiple weak learners into a strong predictor. The key differences are:



**Boosting** is a general method that iteratively trains models to focus more on instances that previous models misclassified. Many boosting algorithms exist like Gradient Boosting, XGBoost, LightGBM, CatBoost, etc.



**AdaBoost** was the first successful boosting technique developed specifically for binary classification problems. It assigns weights to training instances, focusing subsequent models more on misclassified instances from previous iterations.

In both cases, models are added sequentially until a stopping condition is met. The key advantage is reducing bias and variance to improve predictive performance.

So in summary:

* Boosting is a general approach while AdaBoost is a specific implementation of boosting.

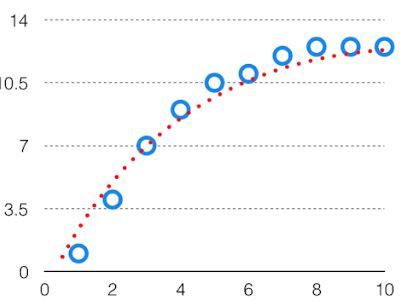
* AdaBoost is constrained to binary classification while boosting algorithms can handle both classification and regression problems.

* Boosting can use any base learner model like decision trees while AdaBoost relies specifically on decision tree classifiers.

The model comparison depends on factors like the problem type, evaluation metrics, data characteristics, and implementation efficiency. For tabular data, LightGBM and CatBoost tend to perform well while XGBoost is preferred for sparse features.

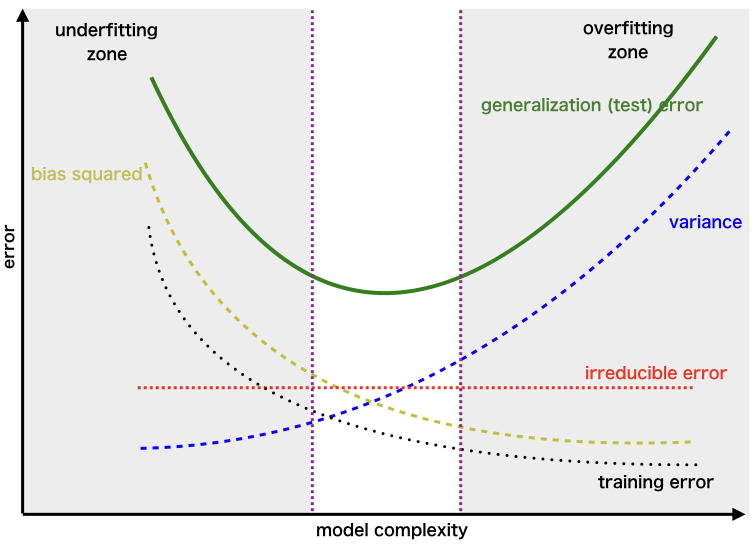
**14. What is bias-variance trade off in machine learning?**

Ans:- 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](https://www.geeksforgeeks.org/bias-vs-variance-in-machine-learning/) 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.**

Ans:- A linear kernel is suitable for separable datasets, while nonlinear kernels such as polynomial, radial basis function (RBF), and sigmoid are better suited for non-separable datasets. Complexity: The complexity of the decision boundary is influenced by the choice of kernel function.