ASSIGNMENT - 5

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 (R^2) and Residual Sum of Squares (RSS) are both commonly used measures to assess the goodness of fit of a regression model, but they capture different aspects of model performance, and the choice between them depends on the context and what you want to evaluate.

1. **R-squared (R^2)**:

- R-squared is a measure of the proportion of the variance in the dependent variable (the response variable) that is explained by the independent variables (predictors) in your regression model.

- R-squared ranges from 0 to 1, with higher values indicating a better fit. A value of 1 means that the model perfectly explains the variance in the dependent variable, while a value of 0 means that the model provides no explanatory power.

- R-squared is a relative measure, and it does not provide information about the absolute goodness of fit or the quality of the model's predictions.

R-squared can be useful for comparing different models or assessing how much of the variation in the dependent variable can be attributed to the predictors. However, it has limitations. For example, it can be artificially inflated by adding more predictors to a model, even if those predictors do not have a meaningful relationship with the dependent variable.

2. **Residual Sum of Squares (RSS)**:

- RSS measures the total squared difference between the observed values of the dependent variable and the predicted values from the regression model. It quantifies the overall error or "residuals" in the model's predictions.

- A smaller RSS indicates a better fit because it means that the model's predictions are closer to the actual observed values.

RSS is an absolute measure of the goodness of fit. It tells you how well the model fits the data in terms of minimizing prediction errors. Unlike R-squared, RSS does not provide information about the proportion of variance explained but gives you a direct measure of the model's prediction accuracy.

Which Measure to Use:

- R-squared is often used when you want to understand the proportion of variance explained by the model, especially in the context of comparing different models. It can provide insights into how well the predictors collectively contribute to explaining the variation in the dependent variable.

- RSS is useful when you want to evaluate the absolute goodness of fit, focusing on the magnitude of the prediction errors. If minimizing prediction errors is a primary concern, RSS is a more appropriate choice.

In practice, both measures can be valuable. R-squared provides a high-level summary of the explanatory power of your model, while RSS gives you a detailed view of the model's prediction accuracy. The choice between them should align with your specific goals and the questions you want to answer about your regression model.

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 is a statistical measure of variability. It indicates the dispersion of data points around the mean and how much the dependent variable deviates from the predicted values in regression analysis.

We decompose variability into the **sum of squares total** (SST), the **sum of squares regression** (SSR), and the **sum of squares error** (SSE). The decomposition of variability helps us understand the sources of variation in our data, assess a model's goodness of fit, and understand the relationship between variables.

SST-

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**—similar to the variance in descriptive statistics. But SST measures the total variability of a dataset, commonly used in regression analysis and ANOVA.

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.

$$SST=\sum_{i=1}^{n}(y_i-\bar{y})^2$$

Where:

y_i – observed dependent variable

$\bar{y}$ – mean of the dependent variable

ESS-

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:

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

Where:

^yi– the predicted value of the dependent variable

¯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.

RSS-

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=1ε2i

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

εi= yi−^yi

Regression analysis aims to minimize the SSE the smaller the error, the better the **regression's** estimation power

Relation between TSS,ESS and RSS -

TSS = ESS + RSS, where TSS is Total Sum of Squares, ESS is Explained Sum of Squares and RSS is Residual Sum of Suqares

3.What is the need of regularization in machine learning?
ANS . While training a machine learning model, the model can easily be overfitted or under fitted. To avoid this, we use regularization in machine learning to properly fit a model onto our test set. Regularization techniques help reduce the chance of overfitting and help us get an optimal model.

4. What is Gini–impurity index?
ANS.

Gini Index

- The Gini Index is the additional approach to dividing a decision tree.

- Purity and impurity in a junction are the primary focus of the Entropy and Information Gain framework.
- The Gini Index, also known as Impurity, calculates the likelihood that somehow a randomly picked instance would be erroneously cataloged.

5. Are unregularized decision-trees prone to overfitting? If yes, why?
ANS.
Decision Trees are prone to over-fitting. A decision tree will always overfit the training data if we allow it to grow to its max depth. Overfitting in decision trees occurs when the tree becomes too complex and captures the noise in the training data, rather than the underlying pattern. This can lead to poor generalization performance on new unseen data

Pruning-

In decision trees, pruning refers to the process of removing branches that do not provide much information gain or that are not necessary for the tree to make accurate predictions. The goal of pruning is to simplify the tree and make it more generalizable to new unseen data.

Pre-pruning **-**We usually apply this technique before the construction of a decision tree. This is the process of early stopping the tree-growing process before it reaches its maximum depth. The tree is stopped from growing by setting a stopping condition, such as a maximum depth or a minimum number of samples in a leaf node. The hyperparameters that can be tuned for early stopping and preventing overfitting are:

max_depth, min_samples_leaf, and min_samples_split

Post-Pruning: **-**It is also known as backward pruning. The Post-pruning technique allows the decision tree model to grow to its full depth, then removes the tree branches that do not provide a significant increase in accuracy on the validation set to prevent the model from overfitting. The process of post-pruning typically involves removing a node and its children from the tree, and then evaluating the impact on the accuracy of the tree using a validation set. If the accuracy does not decrease significantly, the node and its children are removed permanently. This process is repeated recursively for each node in the tree, starting from the bottom-most leaf nodes and moving upwards**.**

6. What is an ensemble technique in machine learning?

ANS. Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote.

Advantage : *Improvement in predictive accuracy.*
Disadvantage : *It is difficult to understand an ensemble of classifiers.*

Main Challenge for Developing Ensemble Models?
The main challenge is not to obtain highly accurate base models, but rather to obtain base models which make different kinds of errors. For example, if ensembles are used for classification, high accuracies can be accomplished if different base models misclassify different training examples, even if the base classifier accuracy is low.

Methods for Independently Constructing Ensembles –

- *Majority Vote*
- *Bagging and Random Forest*
- *Randomness Injection*
- *Feature-Selection Ensembles*
- *Error-Correcting Output Coding*

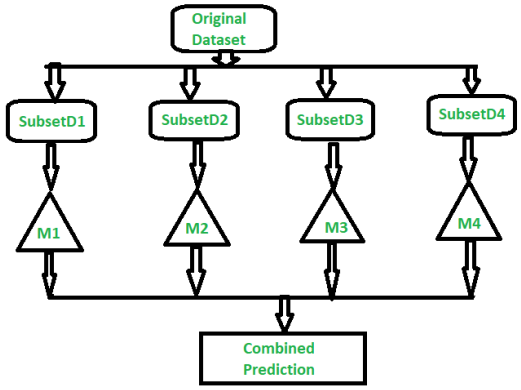
Methods for Coordinated Construction of Ensembles –

- *Boosting*
- *Stacking*

Reliable Classification: *Meta-Classifier Approach*
Co-Training and Self-Training

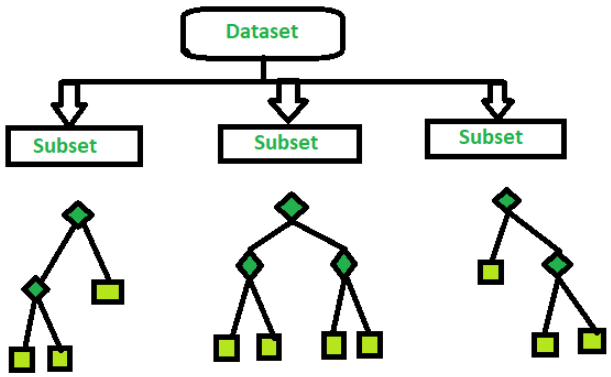
Types of Ensemble Classifier –
Bagging:
Bagging (Bootstrap Aggregation) is used to reduce the variance of a decision tree. Suppose a set D of d tuples, at each iteration *i*, a training set D_i of d tuples is sampled with replacement from D (i.e., bootstrap). Then a classifier model M_i is learned for each training set D < i. Each classifier M_i returns its class prediction. The bagged classifier M* counts the votes and assigns the class with the most votes to X (unknown sample).
Implementation steps of Bagging –

1. Multiple subsets are created from the original data set with equal tuples, selecting observations with replacement.
2. A base model is created on each of these subsets.
3. Each model is learned in parallel from each training set and independent of each other.
4. The final predictions are determined by combining the predictions from all the models.



Random Forest:
Random Forest is an extension over bagging. Each classifier in the ensemble is a decision tree classifier and is generated using a random selection of attributes at each node to determine the split. During classification, each tree votes and the most popular class is returned.
Implementation steps of Random Forest –

1. Multiple subsets are created from the original data set, selecting observations with replacement.
2. A subset of features is selected randomly and whichever feature gives the best split is used to split the node iteratively.
3. The tree is grown to the largest.
4. Repeat the above steps and prediction is given based on the aggregation of predictions from n number of tree



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

Difference Between Bagging and Boosting: Bagging vs Boosting

	Bagging	**Boosting**
Basic Concept	Combines multiple models trained on different subsets of data.	Train models sequentially, focusing on the error made by the previous model.
Objective	To reduce variance by averaging out individual model error.	Reduces both bias and variance by correcting misclassifications of the previous model.
Data Sampling	Use Bootstrap to create subsets of the data.	Re-weights the data based on the error from the previous model, making the next models focus on misclassified instances.
Model Weight	Each model serves equal weight in the final decision.	Models are weighted based on accuracy, i.e., better-accuracy models will have a higher weight.

Error Handling	Each model has an equal error rate.	It gives more weight to instances with higher error, making subsequent model focus on them.
Overfitting	Less prone to overfitting due to average mechanism.	Generally not prone to overfitting, but it can be if the number of the model or the iteration is high.
Performance	Improves accuracy by reducing variance.	Achieves higher accuracy by reducing both bias and variance.
Common Algorithms	Random Forest	AdaBoost, XGBoost, Gradient Boosting Mechanism
Use Cases	Best for high variance, and low bias models.	Effective when the model needs to be adaptive to errors, suitable for both bias and variance errors.

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

ANS.

The RandomForestClassifier is trained using *bootstrap aggregation*, where each new tree is fit from a bootstrap sample of the training observations Zi=(Xi,Yi). The *out-of-bag* (OOB) error is the average error for each Zi calculated using predictions from the trees that do not contain Zi in their respective bootstrap sample. This allows the RandomForestClassifier to be fit and validated whilst being trained .

9. What is K-fold cross-validation?

ANS. Cross-Validation-

You may see cross-validation also being referred to as rotation estimation and/or out-of-sample testing. The overall aim of Cross-Validation is to use it as a tool to evaluate machine learning models, by training a number of models on different subsets of the input data.

Cross-validation can be used to detect overfitting in a model which infers that the model is not effectively generalizing patterns and similarities in the new inputted data.

A typical Cross-Validation workflow:-

In order to perform cross-validation, the following steps are typically taken:

1. Split the dataset into training data and test data
2. The parameters will undergo a Cross-Validation test to see which are the best parameters to select.
3. These parameters will then be implemented into the model for retraining

4. Final evaluation will occur and this will depend if the cycle has to go again, depending on the accuracy and the level of generalization that the model performs.

There are different types of Cross-Validation techniques

- Hold-out
- K-folds
- Leave-one-out
- Leave-p-out

we will be particularly focusing on K-folds.

K-fold Cross-Validation

K-fold Cross-Validation is when the dataset is split into a K number of folds and is used to evaluate the model's ability when given new data. K refers to the number of groups the data sample is split into. For example, if you see that the k-value is 5, we can call this a 5-fold cross-validation. Each fold is used as a testing set at one point in the process.

K-fold Cross-Validation Process:

1. Choose your k-value
2. Split the dataset into the number of k folds.
3. Start off with using your k-1 fold as the test dataset and the remaining folds as the training dataset
4. Train the model on the training dataset and validate it on the test dataset
5. Save the validation score
6. Repeat steps 3 – 5, but changing the value of your k test dataset. So we chose k-1 as our test dataset for the first round, we then move onto k-2 as the test dataset for the next round.
7. By the end of it you would have validated the model on every fold that you have.
8. Average the results that were produced in step 5 to summarize the skill of the model.

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

ANS. Hyperparameter Tuning

Hyperparameter tuning is the process of selecting the optimal values for a machine learning model's hyperparameters. Hyperparameters are settings that control the learning process of the model, such as the learning rate, the number of neurons in a neural network, or the kernel size in a support vector machine. The goal of hyperparameter tuning is to find the values that lead to the best performance on a given task.

Hyperparameter Tuning techniques

Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. The two best strategies for Hyperparameter tuning are:

1. Grid Search
2. Random Search
3. Bayesian Optimization

Grid Search

Grid search is a sort of "brute force" hyperparameter tuning method. We create a grid of possible discrete hyperparameter values then fit the model with every possible combination. We record the model performance for each set then select the combination that has produced the best performance

Grid search is an exhaustive algorithm that can find the best combination of hyperparameters. However, the drawback is that it's slow. Fitting the model with every possible combination usually requires a high computation capacity and significant time, which may not be available.

Random Search

As the name suggests, the random search method selects values at random as opposed to the grid search method's use of a predetermined set of numbers. Every iteration, random search attempts a different set of hyperparameters and logs the model's performance. It returns the combination that provided the best outcome after several iterations. This approach reduces unnecessary computation.

RandomizedSearchCV solves the drawbacks of GridSearchCV, as it goes through only a fixed number of hyperparameter settings. It moves within the grid in a random fashion to find the best set of hyperparameters. The advantage is that, in most cases, a random search will produce a comparable result faster than a grid search.

Bayesian Optimization

Grid search and random search are often inefficient because they evaluate many unsuitable hyperparameter combinations without considering the previous iterations' results. Bayesian optimization, on the other hand, treats the search for optimal hyperparameters as an optimization problem. It considers the previous evaluation results when selecting the next hyperparameter combination and applies a probabilistic function to choose the combination that will likely yield the best results. This method discovers a good hyperparameter combination in relatively few iterations.

11. What issues can occur if we have a large learning rate in Gradient Descent?
ANS . Every time we train a deep learning model, or any neural network for that matter, we're using **gradient descent** (with backpropagation). We use it to *minimize a loss* by *updating the parameters/weights* of the model.

The parameter update depends on two values: a gradient and a **learning rate**. The learning rate gives you control of how big (or small) the updates are going to be. A **bigger learning rate** means **bigger updates** and, hopefully, a model that **learns faster**.

But there is a catch, as always… if the learning rate is **too big**, the model will **not learn anything**. This leads us to *two fundamental questions*:

- **How big is "too big"?**
- **Is there anything I can do to use a bigger learning rate?**

Unfortunately, there is no clear-cut answer for the first question. It will always depend on many factors.

But **there is** an answer for the **second** one: **feature scaling**! How does that work? to show you, in detail, how gradient descent, the learning rate, and the feature scaling are connected.

In this post we will:

- define a **model** and generate a **synthetic dataset**
- randomly **initialize** the parameters
- explore the **loss surface** and **visualize the gradients**
- understand the **effects** of using different **learning rates**

- understand the **effects** of **feature scaling**

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

Logistic regression is known and used as a linear classifier. It is used to come up with a hyper*plane* in feature space to separate observations that belong to a class from all the other observations that do *not* belong to that class. The decision boundary is thus *linear*.
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. Solving for the model parameters can be more challenging but the optimization modules in scipy can help.

13. Differentiate between Adaboost and Gradient Boosting.
ANS.

Features	**Gradient boosting**	**Adaboost**
Model	It identifies complex observations by huge residuals calculated in prior iterations	The shift is made by up-weighting the observations that are miscalculated prior
Trees	The trees with week learners are constructed using a greedy algorithm based on split points and purity scores. The trees are grown deeper with eight to thirty-two terminal nodes. The week learners should stay a week in terms of nodes, layers, leaf nodes, and splits	The trees are called decision stumps.

Classifier	The classifiers are weighted precisely and their prediction capacity is constrained to learning rate and increasing accuracy	Every classifier has different weight assumptions to its final prediction that depend on the performance.
Prediction	It develops a tree with help of previous classifier residuals by capturing variances in data. The final prediction depends on the maximum vote of the week learners and is weighted by its accuracy.	It gives values to classifiers by observing determined variance with data. Here all the week learners possess equal weight and it is usually fixed as the rate for learning which is too minimum in magnitude.
Short-comings	Here, the gradients themselves identify the shortcomings.	Maximum weighted data points are used to identify the shortcomings.
Loss value	Gradient boosting cut down the error components to provide clear explanations and its concepts are easier to adapt and understand	The exponential loss provides maximum weights for the samples which are fitted in worse conditions.

Applications	This method trains the learners and depends on reducing the loss functions of that week learner by training the residues of the model	Its focus on training the prior miscalculated observations and it alters the distribution of the dataset to enhance the weight on sample values which are hard for classification

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

ANS. Bias:-

The bias is known as the difference between the prediction of the values by the Machine Learning model and the correct value. Being high in biasing gives a large error in training as well as testing data. It recommended that an algorithm should always be low-biased to avoid the problem of underfitting. By high bias, the data predicted is in a straight line format, thus not fitting accurately in the data in the data set. Such fitting is known as the **Underfitting of Data**. This happens when the hypothesis is too simple or linear in nature.

Variance

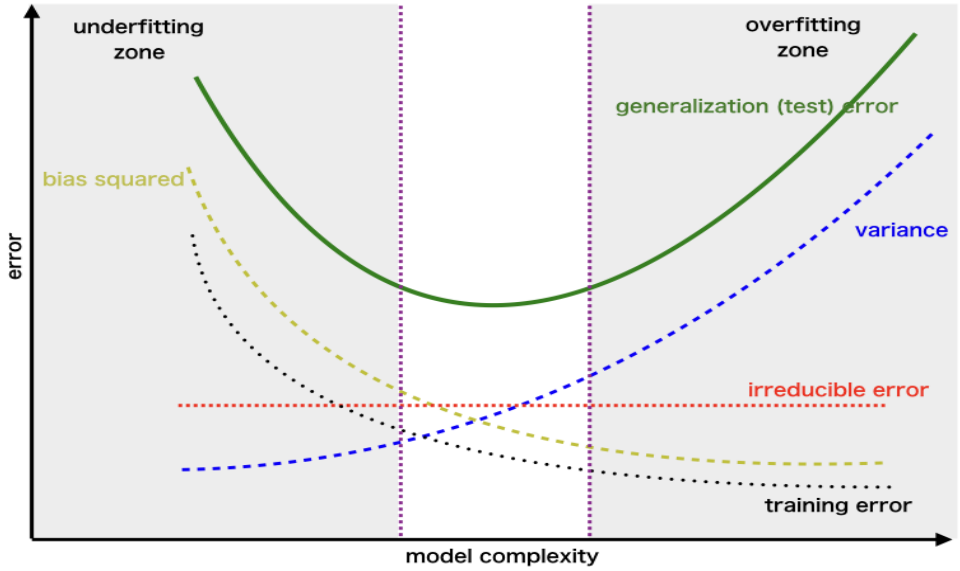
The variability of model prediction for a given data point which tells us the spread of our data is called the variance of the model. The model with high variance has a very complex fit to the training data and thus is not able to fit accurately on the data which it hasn't seen before. As a result, such models perform very well on training data but have high error rates on test data. When a model is high on variance, it is then said to as **Overfitting of Data**. Overfitting is fitting the training set accurately via complex curve and high order hypothesis but is not the solution as the error with unseen data is high. While training a data model variance should be kept low.

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



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

ANS.

Kernel Function is a method used to take data as input and transform it into the required form of processing data. "Kernel" is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data. So, Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces.

- **Gaussian Kernel:** It is used to perform transformation when there is no prior knowledge about data.
- **Gaussian Kernel Radial Basis Function (RBF):** Same as above kernel function, adding radial basis method to improve the transformation.
- **Sigmoid Kernel:** this function is equivalent to a two-layer, perceptron model of the neural network, which is used as an activation function for artificial neurons.
- **Polynomial Kernel:** It represents the similarity of vectors in the training set of data in a feature space over polynomials of the original variables used in the kernel.
- Linear Kernel: used when data is linearly separable.