MACHINE LEARNING ASSIGNMENT –5

Q1. 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?

Answer: 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:

$R^2 = 1 - \frac{RSS}{TSS}$

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 $i\text{-th}$ 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.

Q2. What is 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.

Answer:

- TSS (Total Sum of Squares)

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.

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

TSS=n∑i=1(yi−¯y)2$SST=\sum i=1n(yi-y^-)2$

Where:

yi yi – observed dependent variable

¯y y^- – mean of the dependent variable

- ESS (Explained Sum of Squares)

The **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 ESS formula is the following:

ESS=n∑i=1(^yi−¯y)2$SSR=\sum i=1n(y\hat{}i-y^-)2$

Where:

^yi $y\hat{}i$ – the predicted value of the dependent variable

¯y y^- – mean of the dependent variable

If ESS equals SST, our regression model perfectly captures all the observed variability, but that's rarely the case.

- RSS (Residual Sum of Squares)

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

The RSS calculation uses the following formula:

RSS=n∑i=1ε2i$SSE=\sum i=1n\varepsilon i2$

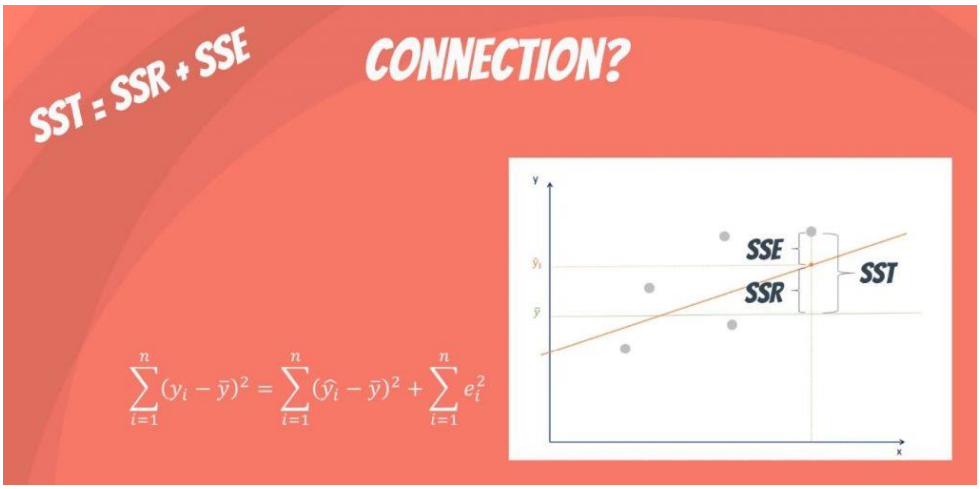
Where εiεi is the difference between the actual value of the dependent variable and the predicted value:

εi= yi−^yi$\varepsilon i= yi-y\hat{}i$

Regression analysis aims to minimize the RSS—the smaller the error, the better the regression's estimation power.

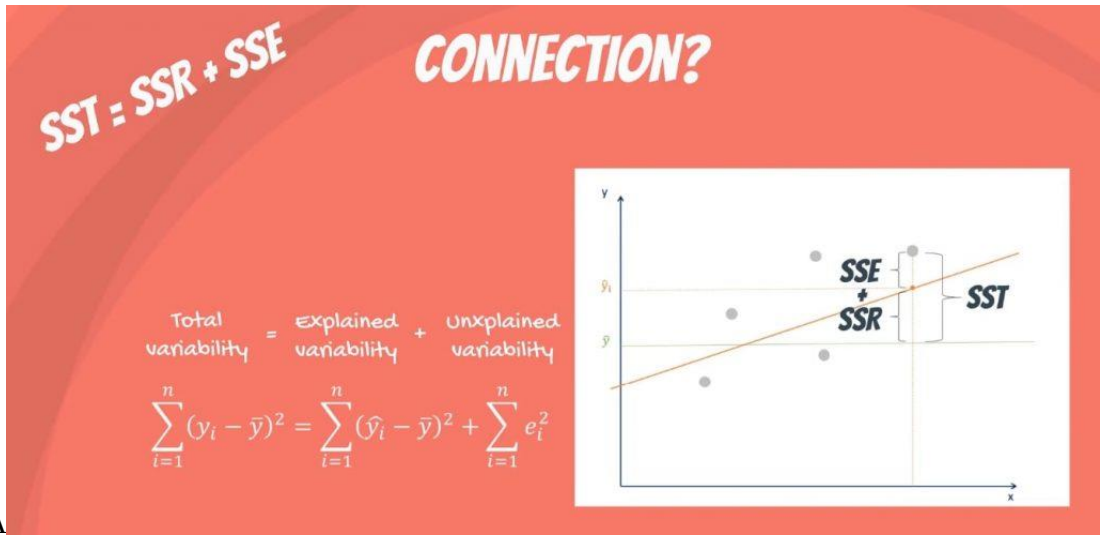
RELATION BETWEEN TSS,RSS,ESS

Mathematically, TSS = RSS + ESS.



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.



QA

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.

Q3. What is the need of regularization in machine learning?

Answer: 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. 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

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 well and keeping the model simple enough to perform well on new data.

Q4. What is Gini–impurity index?

Answer: Gini Index Formula

The Gini Index is a proportion of the impurity or inequality of a circulation, regularly utilized as an impurity measure in decision tree algorithms. With regards to decision trees, the Gini Index is utilized to determine the best feature to split the data on at every node of the tree.

The formula for Gini Index is as per the following:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

where pi is the probability of a thing having a place with a specific class.

For example, we should consider a binary classification issue with two classes An and B. On the off chance that the probability of class An is p and the probability of class B is (1-p), then the Gini Index can be calculated as:

The value of the Gini Index goes from 0.0 to 0.5 for binary classification problems, where 0.0 demonstrates a perfectly pure node (all examples have a place with a similar class) and 0.5 shows a perfectly impure node (tests are equally distributed across the two classes).

Q5. Are unregularized decision-trees prone to overfitting? If yes, why?

Answer: Decision trees are a popular and powerful method for data mining, as they can handle both numerical and categorical data, and can easily interpret the results. However, decision trees can also suffer from overfitting, which means that they learn too much from the training data and fail to generalize well to new data.

Q6. What is an ensemble technique in machine learning?

Answer: Ensemble techniques in machine learning involve combining multiple models to improve performance. One common ensemble technique is bagging, which uses bootstrap sampling to create multiple datasets from the original data and trains a model on each dataset. Another technique is boosting, which trains models sequentially, each focusing on the previous models' mistakes. Random forests are a popular ensemble method that uses decision trees as base learners and combines their predictions to make a final prediction. Ensemble techniques are effective because they reduce overfitting and improve generalization, leading to more robust models.

Simple Ensemble Techniques

Simple ensemble techniques combine predictions from multiple models to produce a final prediction. These techniques are straightforward to implement and can often improve performance compared to individual models.

Max Voting

In this technique, the final prediction is the most frequent prediction among the base models. For example, if three base models predict the classes A, B, and A for a given sample, the final prediction using max voting would be class A, as it appears more frequently.

Averaging

Averaging involves taking the average of predictions from multiple models. This can be particularly useful for regression problems, where the final prediction is the mean of predictions from all models. For classification, averaging can be applied to the predicted probabilities for a more confident prediction.

Weighted Averaging

Weighted averaging is similar, but each model's prediction is given a different weight. The weights can be assigned based on each model's performance on a validation set or tuned using grid or randomized search techniques. This allows models with higher performance to have a greater influence on the final prediction.

Q7. What is the difference between Bagging and Boosting techniques?

Answer:

Bagging (Bootstrap Aggregating)

- Bagging is a technique where multiple subsets of the dataset are created through bootstrapping (sampling with replacement).
- A base model (often a decision tree) is trained on each subset, and the final prediction is the average (for regression) or majority vote (for classification) of the individual predictions.
- Bagging helps reduce variance and overfitting, especially for unstable models.

Boosting

- Boosting is an ensemble technique where base models are trained sequentially, with each subsequent model focusing on the mistakes of the previous ones.
- The final prediction is a weighted sum of the individual models' predictions, with higher weights given to more accurate models.
- Boosting algorithms like Ada Boost, Gradient Boosting, and XG Boost are popular because they improve model performance.

Q8. What is out-of-bag error in random forests?

Answer: Random Forest Classifier 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 Random Forest Classifier to be fit and validated whilst being trained.

Q9. What is K-fold cross-validation?

Answer: In K-fold cross-validation, the data set is divided into a number of K-folds and used to assess the model's ability as new data become available. K represents the number of groups into which the data sample is divided. For example, if you find the k value to be 5, you can call it 5-fold cross-validation.

Q10. What is hyper parameter tuning in machine learning and why it is done?

Answer: Hyper parameter tuning is the process of selecting the optimal values for a machine learning model's hyper parameters. The goal of hyper parameter tuning is to find the values that lead to the best performance on a given task.

Hyper parameters are the parameters that are explicitly defined to control the learning process before applying a machine-learning algorithm to a dataset. These are used to specify the learning capacity and complexity of the model. Some of the hyper parameters are used for the optimization of the models, such as Batch size, learning rate, etc., and some are specific to the models, such as Number of Hidden layers, etc.

Q11. What issues can occur if we have a large learning rate in Gradient Descent?

Answer: Gradient descent is an optimization algorithm used in machine learning to minimize the cost function by iteratively adjusting parameters in the direction of the negative gradient, aiming to find the optimal set of parameters.

If the learning rate is too high, the algorithm may overshoot the minimum, and if it is too low, the algorithm may take too long to converge.

Q12. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?

Answer: No we cannot use Logistic Regression for classification of Non-Linear Data

Logistic regression is a type of linear model that predicts the probability of a binary outcome, such as yes or no, true or false, or 0 or 1. It uses a logistic function, also known as a sigmoid function, to map the input features to a value between 0 and 1. The logistic function has an S-shaped curve that flattens at the extremes. Logistic regression then uses a threshold, usually 0.5, to classify the output as 0 or 1.

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.

Q13. Differentiate between Ada boost and Gradient Boosting.

Answer: The most significant difference is that gradient boosting minimizes a loss function like MSE or log loss while Ada Boost focuses on instances with high error by adjusting their sample weights adaptively.

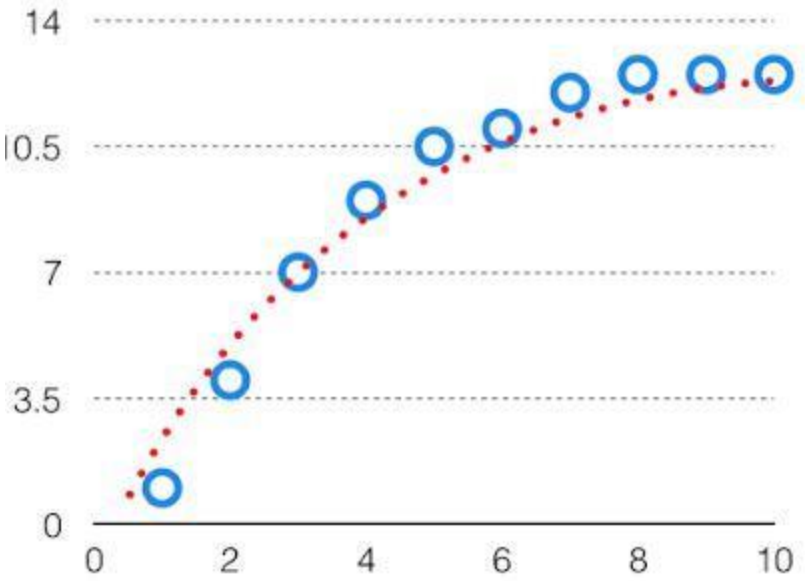
Gradient boosting models apply shrinkage to avoid overfitting which Ada Boost does not do. Gradient boosting also performs subsampling of the training instances while Ada Boost uses all instances to train every weak learner.

Overall gradient boosting is more robust to outliers and noise since it equally considers all training instances when optimizing the loss function. Ada Boost is faster but more impacted by dirty data since it fixates on hard examples.

Q14. What is bias-variance trade off in machine learning?

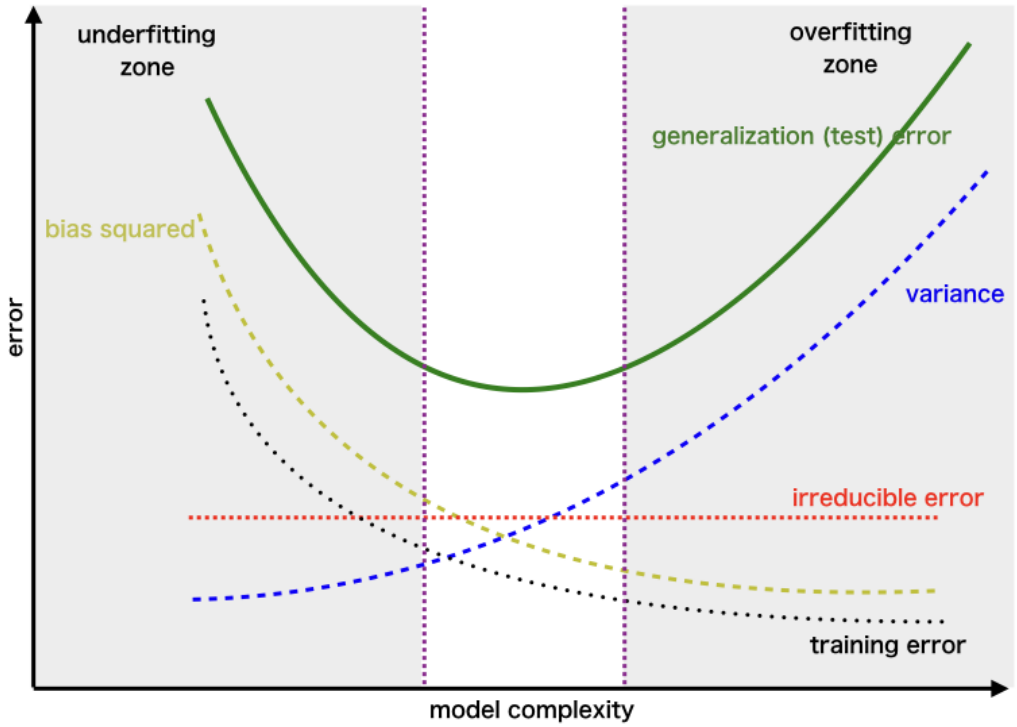
Answer: The bias-variance tradeoff is about finding the right balance between simplicity and complexity in a machine learning model. High bias means the model is too simple and consistently misses the target, while high variance means the model is too complex and shoots all over the place.

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.

Q15. Give short description each of Linear, RBF, Polynomial kernels used in SVM.

Answer: 1. Linear Kernel

- The linear kernel is the simplest and is used when the data is linearly separable.
- It calculates the dot product between the feature vectors.

2. Polynomial Kernel

- The polynomial kernel is effective for non-linear data.
- It computes the similarity between two vectors in terms of the polynomial of the original variables.

3. Radial Basis Function (RBF) Kernel

- The RBF kernel is a common type of Kernel in SVM for handling non-linear decision boundaries.
- It maps the data into an infinite-dimensional space.