MACHINE LEARNING

ASSIGNMENT - 5

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?
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.
3. What is the need of regularization in machine learning?
4. What is Gini–impurity index?
5. Are unregularized decision-trees prone to overfitting? If yes, why?
6. What is an ensemble technique in machine learning?
7. What is the difference between Bagging and Boosting techniques?
8. What is out-of-bag error in random forests?
9. What is K-fold cross-validation?
10. What is hyper parameter tuning in machine learning and why it is done?
11. What issues can occur if we have a large learning rate in Gradient Descent?
12. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?
13. Differentiate between Adaboost and Gradient Boosting.
14. What is bias-variance trade off in machine learning?
15. Give short description each of Linear, RBF, Polynomial kernels used in SVM.

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?

Regression is a technique in statistics that determines the relationship between a dependent variable and one or more independent variables. It is generally used for prediction purposes.

R-squared:

The R-squared value is a measure of how well the model explains the variation in the data. It is calculated by dividing the explained variation by the total variation.

The explained variation is the variation that is explained by the model, while the total variation is the variation in the data. A higher R-squared value indicates a better fit for the model. R-squared, also known as the coefficient of determination

R-squared is useful for understanding how much of the variation in the data is explained by the model. A higher R-squared value indicates a better fit for the model. However, R-squared does not consider the complexity of the model, which can lead to underfitting.

Residual Sum of Squares (RSS):

The RSS is a measure of how well the model fits the data. It is calculated by summing the squared differences between the predicted values and the actual values.

RSS is useful for comparing different models and selecting the best one. A lower RSS value indicates a better fit for the model. However, RSS does not consider the number of variables in the model, which can lead to overfitting.

Comparing RSS and R-squared:

While both RSS and R-squared are important measures of model fit, they have different strengths and weaknesses. RSS is a measure of how well the model fits the data, while R-squared is a measure of how well the model explains the variation in the data.

Conclusion:

Both RSS and R-squared are important measures of model fit. RSS is useful for comparing different models and selecting the best one, while R-squared is useful for understanding how much of the variation in the data is explained by the model. It is important to consider both measures when assessing the performance of a model.

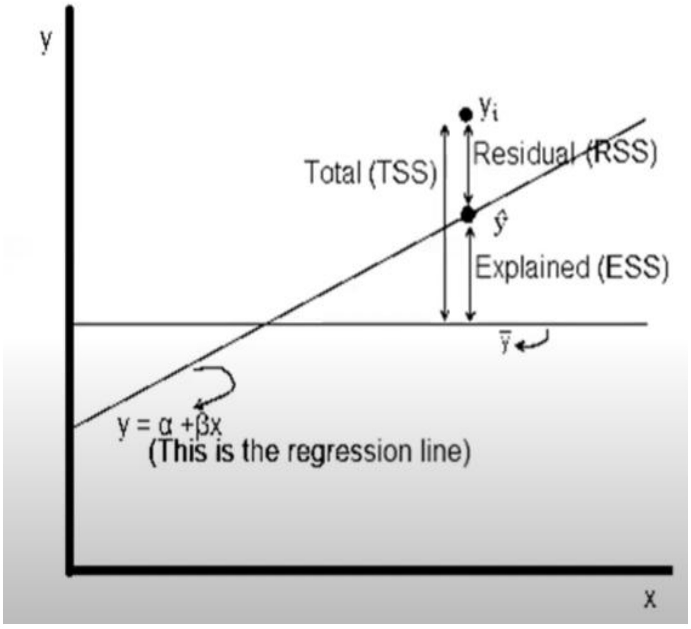
2. What are TSS (Total Sum of Squares), ESS (Explained Sum of Squares) and RSS (Residual Sum of Squares) in regression. Also mention the equation relating these three metrics with each other

The diagram to the right represents TSS, ESS and RSS relation.

$\hat{\boldsymbol{y}}$ is the predicted value of y given x.

$\boldsymbol{y_i}$ is the actual observed value of y.

$\bar{\boldsymbol{y}}$ is the mean of y



Total Sum of Squares:

The TSS is the total variation of the actual Y values from their sample mean.

TSS = $\Sigma(y_i - \bar{y})^2$

TSS can be divided into two categories:

- **ESS (Explained Sum of Squares)**:

  The ESS is the portion of total variation that measures how well the regression equation explains the relationship between x and y.

  ESS = $\Sigma(\hat{Y} - \bar{y})^2$

- **RSS (Residual Sum of Squares):**

  This expression is also known as unexplained variation and is the portion of the total variation that measures

  Errors between the actual values of Y and those estimated by the regression equation.

  RSS = $\Sigma\ (y_i - \hat{y})^2$

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. The commonly used regularization techniques are

- Lasso Regularization
- Ridge Regularization
- Elastic Net Regularization

Role of Regularization:

- **Complexity Control:**

  Regularization helps control model complexity by preventing overfitting to training data, resulting in better generalization to new data.

- **Preventing Overfitting:**

  One way to prevent overfitting is to use regularization, which penalizes large coefficients.

- **Balancing Bias and Variance:**

  Regularization can help balance the trade-off between model bias (underfitting) and model variance (overfitting) in machine learning, which leads to improved performance.

- **Feature Selection:**

  Some regularization methods, such as Lasso, promote solutions that drive some feature coefficients to zero. This automatically selects important features while excluding less important ones.

- **Handling Multicollinearity:**

  When features are highly correlated (multicollinearity), regularization can stabilize the model by reducing coefficient sensitivity to small data changes.

Benefits of Regularization

- Regularization improves model performance by preventing excessive weighting of outliers or irrelevant features.

- Regularization prevents models from becoming overly complex, which is especially important when dealing with limited data or noisy environments.

- Regularization can help handle multicollinearity by reducing the correlated coefficients.

4. What is Gini–impurity index?

Decision Tree is one of the most popular and powerful classification algorithms that we use in machine learning. The concept behind the decision tree is that it helps to select appropriate features for splitting the tree into subparts.

The feature that comes in the root node has impact on labels. The feature that has strong relation with the label would be assigned primarily.

Now the question is, out all the features who is having the highest contribution on the label?

To find the best feature we have 2 criterions:

- Entropy
- Gini impurity

Entropy:

Entropy is eventually used to calculate Information Gain.
The range of values Entropy can have between 0 to 1.
Entropy of 0 denotes a pure node and 1 denotes most impure node.

Gini impurity:

It is similar to Entropy, but here it will calculate the impurity data present in the features.

The lesser the Gini Impurity, the better the split.

A Gini Impurity can be in between 0 to 1, where 0 denotes a pure node and 1 denotes a most impure node.

Conclusion

Gini Impurity 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.

Entropy and Gini criterion measure similar performance metrics. Calculating Gini Impurity is much faster as it is less expensive to compute, whereas Entropy is a more expensive computation. However, the results obtained from Entropy are slightly better.

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

Yes, it is possible that unregularized decision tree can overfit. Overfitting in decision tree models occurs when the tree becomes too complex and captures noise in the training data.

Reasons for overfitting include:

- Complexity:
  - Decision trees become overly complex, fitting training data perfectly but struggling to generalize to new data.
- Memorizing Noise:
  - It can focus too much on specific data points or noise in the training data.
- Overly Specific Rules:
  - Might create rules that are too specific to the training data, leading to poor performance on new data.
- Feature Importance Bias:
  - Certain features may be given too much importance by decision trees, even if they are irrelevant, contributing to overfitting.
- Lack of Early Stopping:
  - Without proper stopping rules, decision trees may grow excessively.

Strategies to Overcome Overfitting in Decision Tree Models

Limiting Tree Depth:

Setting a maximum depth for the decision tree restricts the number of levels or branches it can have.

Minimum Samples per Leaf Node:

Specifying a minimum number of samples required to create a leaf node ensures that each leaf contains a sufficient amount of data to make meaningful predictions.

Pruning Techniques:

This helps simplify the model and prevent it from memorizing noise in the training data.

Conclusion

Decision trees are known for their simplicity in machine learning, yet overfitting causes a common challenge. This occurs when the model learns the training data too well but fails to generalize to new data.

6. What is an ensemble technique in machine learning?

Ensemble learning in machine learning combines multiple individual models to create a stronger, more accurate predictive model. By leveraging the diverse strengths of different models, ensemble learning aims to mitigate errors, enhance performance, and increase the overall robustness of predictions, leading to improved results across various tasks in machine learning and data analysis.

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

Bagging:

Bagging (Bootstrap Aggregating) is an ensemble learning technique designed to improve the accuracy and stability of machine learning algorithms.

Boosting:

Boosting is another ensemble learning technique that focuses on creating a strong model by combining several weak models.

Differences between Bagging and Boosting:

Bagging trains each base model independently and in parallel, using bootstrap sampling to create multiple subsets of the training data. The final prediction is then made by averaging the predictions of all base models. Bagging focuses on reducing variance and overfitting by creating diverse models.

Boosting trains models sequentially, with each subsequent model focusing on correcting the errors made by the previous ones. Boosting adjusts the weights of training instances to prioritize difficult-to-classify instances, thus reducing bias and improving predictive accuracy. The final prediction is made by combining the predictions of all models, typically using a weighted voting or averaging approach.

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

The Random Forest algorithm comes along with the concept of Out-of-Bag.

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.

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.

9. What is K-fold cross-validation?

Cross validation is a technique used in machine learning to evaluate the performance of a model on unseen data. The main purpose of cross validation is to prevent overfitting, which occurs when a model is trained too well on the training data and performs poorly on new, unseen data.

Types of Cross-Validation

- Holdout Validation
- K-Fold Cross Validation
- LOOCV (Leave One Out Cross Validation)

➢ **Hold-out Validation:**

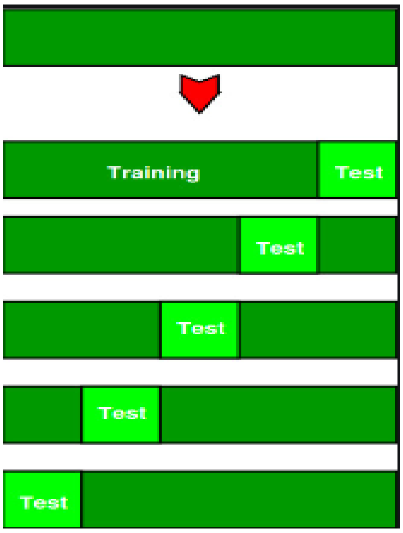
Hold-out is when you split up your dataset into a 'train' and 'test' set. The training set is what the model is trained on, and the test set is used to see how well that model performs on unseen data. A common split when using the hold-out method is using 80% of data for training and the remaining 20% of the data for testing.

➢ **K-Fold Cross Validation:**

In K-Fold Cross Validation, we split the dataset into k number of subsets (known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.

The diagram below shows an example of the training subsets and evaluation subsets generated in k-fold cross-validation.

Advantages	**Disadvantages**
Overcoming Overfitting	Computationally Expensive
Model Selection	Time-Consuming
Hyperparameter tuning	Bias-Variance Trade-off



➢ **LOOCV (Leave One Out Cross Validation):**

In this method, we perform training on the whole dataset but leaves only one data-point of the available dataset and then iterates for each data-point.

An advantage of using this method is that we make use of all data points and hence it is low bias.

The major drawback of this method is that it leads to higher variation in the testing model as we are testing against one data point. If the data point is an outlier it can lead to higher variation. Another drawback is it takes a lot of execution time as it iterates over 'the number of data points'.

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

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 algorithm, leaf_size, n_neighbors in Knn and criterion, max_depth, max_leaf_nodes, min_samples_leaf, min_samples_split in Decision tree.

The goal of hyperparameter tuning is to find the values that lead to the best performance on a given machine learning task.

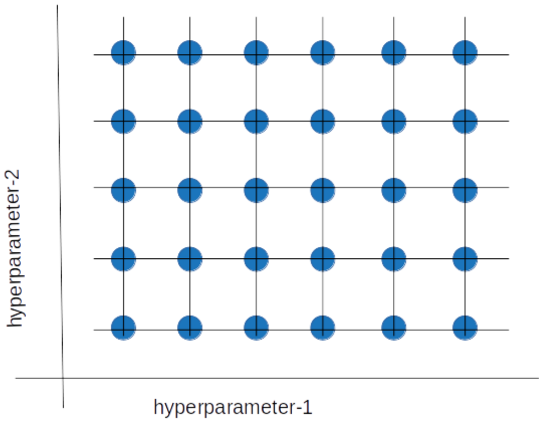
Methods for tuning hyperparameters

- Grid search
- Random search

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 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.



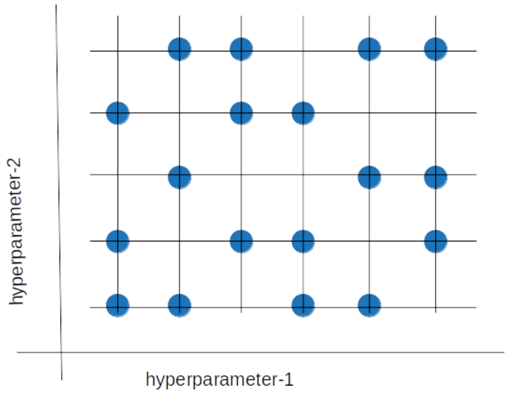
Random search

The random search method chooses values randomly rather than using a predefined set of values like the grid search method.

Random search tries a random combination of hyperparameters in each iteration and records the model performance. After several iterations, it returns the mix that produced the best result.

The benefit is that random search typically requires less time than grid search to return a comparable result. It also ensures we don't end up with a model that's biased.

Its drawback is that the result may not be the best possible hyperparameter combination.



Random search tries a random combination of hyperparameters in each iteration and records the model performance. After several iterations, it returns the mix that produced the best result.

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

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.

While gradient descent is a powerful optimization algorithm, it also faces some challenges affecting its performance.

Some of these challenges include:

The learning is merely signifying how much does the algorithm go near to optimal weights right way in gradient descent. Moreover, having a large learning rate might lead to the following problems:

Overshooting:

This means that the algorithm may actually overshoot the minimum and end up failing to achieve an optimal solution

Divergent:

The algorithm might become unstable and goes for divergence.

Oscillations:

The performance of the algorithm oscillates between epochs during training.

Less accurate weights:

It may give you final, calibrated ones but not necessarily best.

Overfitting:

The algorithm will over fit the training data if model is too complex

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

A logistic regression is a kind of linear classifier used for binary classification tasks. It has a disadvantage which is that the accuracy of it falls down when we have non-linear data. Logistic Regression assumes a linear relationship between input features and the labels.

When the decision is non-linear, Logistic Regression struggles to capture complex patterns.

Non-linear data often involves complex feature interactions, which Logistic Regression can’t explicitly model.

Alternative models like decision trees handle non-linear data more effectively.

13. Differentiate between Adaboost and Gradient Boosting.

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 trees are called decision stumps.
Classifier	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.	Every classifier has different weight assumptions to its final prediction that depend on the performance.
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 a 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?

Bias-Variance Trade-off is a fundamental concept in machine learning that deals with the balance between model bias and variance.

Why Bias-Variance Trade-off is Important

Bias-Variance Trade-off is crucial in machine learning because it directly impacts a model's predictive performance. A model with high bias will consistently produce predictions that are far from the actual values, while a model with high variance will produce widely varying predictions for different training datasets. In both cases, the model's ability to generalize to new, unseen data is compromised.

If algorithms fit too complex then it may be on high variance and low bias.

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.

Technologies Related to Bias-Variance Trade-off

Techniques contribute to understanding and managing the bias-variance trade-off:

Regularization:

A technique used to control model complexity and prevent overfitting by penalizing large parameter values.

Cross-Validation:

A method to evaluate a model's performance on multiple data subsets, helping to estimate bias and variance.

Ensemble Methods:

Combining multiple models to reduce variance and improve overall predictive performance.

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

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it’s best suited for classification.

The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space.

Select the Right Kernel

Choosing the right kernel is crucial for various ML algorithms, especially SVM. To choose the right kernel in SVM, we have to take into consideration the type of problem, the computational complexity, and the characteristics of the data.

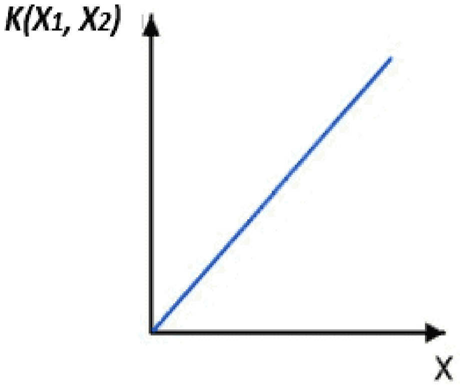
- **Linear Kernel**

We use Linear kernels when the number of features is large compared to the number of samples or when the data are linearly separable.

Linear kernels’ suite for problems high-dimensional data.

The linear kernel can be expressed as:

$$K(x_1, x_2) = x_1^T \cdot x_2 + c$$



where:

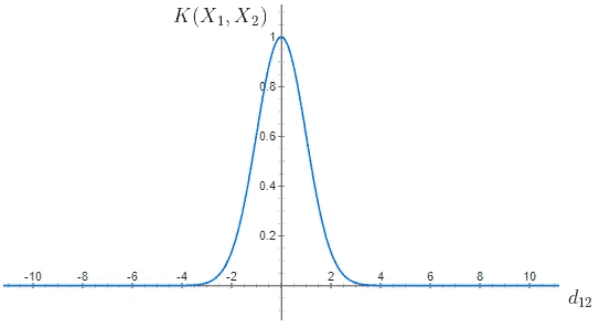
- (x_1^T) denotes the transpose of vector.
- (.) represents vector multiplication.
- (x_2) is another vector.
- (c) is a constant.

- **Radial Basis Function Kernel (RBF)**

The RBF kernel is suitable for non-linear problems and is the default choice for SVM.

It’s powerful when there is no prior knowledge of the data, and we can capture complex relationships between data points.

The RBF kernel can be expressed as: $K(x_1, x_2) = \exp(-y(||x1 - x2)||^2)$

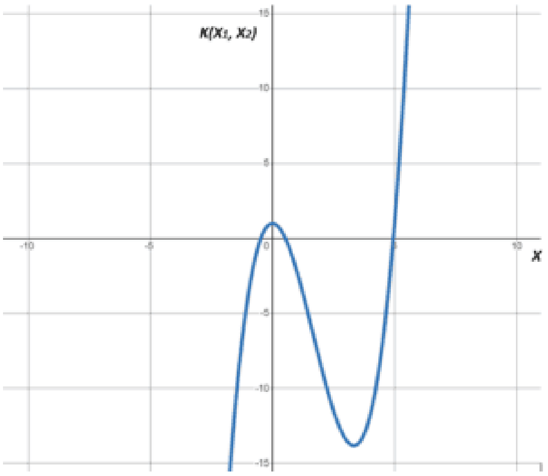


❖ Polynomial Kernel

Polynomial kernels are useful when treating problems that show polynomial behaviour**.**

They are commonly used for computer vision and image recognition tasks. We express the polynomial kernel as:

$$K(x_1, x_2) = (\gamma\, x1. x2 + c)^d$$



Advantages and Disadvantages of Each Kernel Type

Kernel Type	Advantages	Disadvantages
Linear	Computationally efficient – Works well for high-dimensional data.	Limited to linearly separable data – May not capture complex relationships in nonlinear data.
Radial Basis Function (RBF)	Effective for capturing complex nonlinear relationships.	Can be sensitive to overfitting.
Polynomial	It is useful for problems with polynomial behaviour and can also capture nonlinear relationships in the data.	Prone to overfitting in high-degree polynomials, it is sensitive to the choice of degree parameter.