Machine learning

Mayank Singh Deo

Question 1

When comparing a regression model's goodness of fit, R-squared (R2) is thought to be superior to the residual sum of squares (RSS). The percentage of variance in the dependent variable that is explained by the independent variables in the regression model is expressed statistically as R-squared. However, RSS is a measurement of the total variance of the residuals, or the difference between the values of the dependent variable that were predicted and those that were observed. Higher values denote a better fit, and the R-squared standard measure has a range from 0 to 1. RSS, on the other hand, is flexible and is dependent on the units of the dependent variable. R-squared is a useful metric for evaluating the fit of various models since it can be simply understood as the percentage of variance in the dependent variable that is explained by the independent variables. In contrast, RSS's interpretation of the proportion of variance explained is ambiguous. When comparing a regression model's goodness of fit, R-squared (R2) is thought to be superior to the residual sum of squares (RSS). The percentage of variance in the dependent variable that is explained by the independent variables in the regression model is expressed statistically as R-squared. However, RSS is a measurement of the total variance of the residuals, or the difference between the values of the dependent variable that were predicted and those that were observed.

Question 2

TSS, ESS, and RSS, respectively, are the three measures of the total variation in the dependent variable (y) and the variation that is explained by the independent variables (x) in regression analysis.

Total Sum of Squares (TSS) It is the total of the squared discrepancies between the dependent variable's actual values (y) and its mean (). It represents the overall variation in the dependent variable, regardless of the impact of the independent variables. The TSS formula is:

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

ESS-Explained sum of squares: It is the sum of the squared differences between the mean of the dependent variable and the predicted values of the dependent variable, which is called the squared difference. It depicts the independent variable-explained variation in the dependent variable. The ESS formula is:

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

RSS- Residual Square Sum (RSS): It is the sum of the squared differences between the predicted and actual values of the dependent variable. It symbolizes the variation in the dependent variable that the independent variables cannot account for. RSS's formula is as follows:

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

Three metrices related with each other :

TSS=ESS+RSS

Question 3

Regularization is a technique used in machine learning to decrease overfitting and improve the generalizability of a model. A model that has been overfitted will detect noise and random oscillations in the data because it fits the training data too well. As a result, the model might not perform well with fresh, untested data.

By attaching a penalty term to the loss function that the model is attempting to reduce, regularisation helps to solve this issue. The penalty term places restrictions on the model's flexibility and complexity, lowering the likelihood of overfitting.

Two types of regression techniques used in machine learning: Lasso and Ridge

1. Lasso: adds a penalty term to the loss function that is inversely proportional to the model coefficients' absolute values. Sparse solutions with some coefficients set to zero and feature selection are the result of this.
2. Ridge: adds a penalty term to the loss function inversely proportional to the model coefficients' square. This outcome in smooth arrangements, where every one of the coefficients is decreased, however, none are set to nothing.

Question 4

The Gini impurity index can be used to assess the impurity or heterogeneity of a decision tree algorithm. It is widely used in classification issues to assess the quality of a split in a decision tree, which separates a dataset into subsets depending on the values of a feature.

The Gini impurity index calculates the likelihood that a random example from a subset would be incorrectly classified based on the distribution of classes in that subset. It ranges from 0 to 1 with 1 denoting an impure set in which classes are evenly distributed and 0 denoting a pure set in which all examples belong to the same class.

The Gini impurity index is a popular metric for rating decision trees in machine learning frameworks and libraries. This simple and basic measurement is simple to grasp and interpret for non-experts.

Question 5

Yes, decision trees that are not regularized are prone to overfitting.

Decision trees are a sort of model that are particularly susceptible to overfitting when the tree is deep and complex. This is so that decision trees can accommodate the noise in the training data due to their large variation. They find it challenging to extrapolate from existing data to new data as a result. Unregularized decision trees are particularly prone to overfitting because they lack a mechanism for regulating the size or complexity of the tree. Without regularisation, a decision tree would split until each leaf node has just one example, producing a tree that exactly matches the training set but is unlikely to generalise to new data. Overfitting can be avoided by using regularised decision trees, which add a penalty term to the cost function that the model is seeking to reduce. This penalty term keeps the tree from growing too large or intricate, making the model simpler to generalise. Pruning, which involves removing branches from an existing decision tree, can also be used to reduce the complexity of the tree and prevent overfitting.

Question 6

An ensemble technique in machine learning is a means to improve a system's performance overall by combining the predictions of various independent models. The premise behind ensemble approaches is that by integrating various models, the strengths of each model can be improved while minimising its weaknesses.

Bagging: The method of bagging involves training multiple models on randomly chosen subsets of the training data and combining the predictions of those models using a simple average or majority voting scheme. Bagging can help to lessen overfitting by adding randomness to the training process and lowering the variance of the final model. Boosting: Boosting involves successively training several models, each of which tries to fix the flaws of the prior model. Predictions from the various models are combined using a weighted sum or a more intricate technique. Boosting can make the final model less biassed and more accurate.

In machine learning, ensemble approaches are frequently employed and have been shown successful in a variety of applications. Both classification and regression models can benefit from their use, and they work especially well when the individual models have different advantages and disadvantages.

Question 7

The way that bagging and boosting combine the predictions from various models is the primary distinction between them. While boosting entails training models sequentially with each model attempting to fix the mistakes of the preceding model, bagging entails training different models on independent subsets of the training data and combining their predictions. Ensemble techniques like bagging and boosting have the potential to increase the performance of machine learning systems.

By adding randomization to the training process and lowering the variance of the final model, bagging can help to reduce overfitting. There is no reliance between the models in the ensemble because each one is trained independently. Boosting can aid in lowering bias and enhancing the final model's accuracy. There is a reliance between them since each model in the ensemble is trained using the mistakes of the prior model.

Question 8

In random forests, the out-of-bag (OOB) error, which is exclusively calculated from the training set, is used to quantify the prediction error of the model on unobserved data.
The initial training set is used to train a bootstrap sample for each tree in a random forest model, therefore some examples are left out of the training set for some trees. The examples for a certain tree that are not part of the training set are known as the out-of-bag examples. The out-of-bag error is then calculated using the average prediction error of each tree on the matching out-of-bag examples. Because each tree only sees a piece of the training data, the out-of-bag error offers a fair assessment of the model's performance on unobserved data. The out-of-bag error is a helpful performance metric because it calculates the model's generalisation error without requiring a separate validation set. This approach can also be used to analyse the various hyperparameters of the random forest model, such as the number of trees in the forest and the maximum depth of each tree.

Question 9

K-fold cross-validation is a popular technique in machine learning for assessing a model's performance on a dataset. The dataset is divided into K equal-sized folds, with K-1 folds acting as the training set and the last fold acting as the validation set. Each overlap is used as the approval set exactly once throughout each iteration of this cycle, which is repeated K times. K-overlay cross-approval is used to evaluate how well a model fits new data and is particularly useful when there is a risk of overfitting or when the dataset is small. It offers a more accurate evaluation of the model's performance by utilising all of the data that are available for both training and validation. Two popular variants of Kfold cross-validation are stratified K-fold cross-validation and nested K-fold cross-validation. The folds are made in nested K-fold cross-validation to guarantee that each class is equally represented in the training and validation sets. Several iterations of hyperparameter tweaking are carried out in layered K-fold cross-validation.

Question 10

"Hyperparameter tuning" is the process of selecting the ideal values for a machine learning algorithm's hyperparameters. Hyperparameters are model parameters that are predetermined and not determined by data; they are established before the training process. Examples of hyperparameters include the learning rate in a neural network, the number of trees in a random forest, or the regularisation parameter in a linear regression model. Hyperparameter tuning aims to improve the machine learning model's performance with fresh, untried data. Finding the hyperparameter settings that give the model the best performance is the goal. There are several methods for adjusting hyperparameters, including grid search and random search. Grid search requires thoroughly examining a broad range of values for each hyperparameter before choosing the ideal set of values. Random search is the method of choosing the optimal value combination by randomly sampling from the hyperparameter space.

Question 11

A high learning rate in gradient descent can lead to a variety of issues, such as:

Divergence: A high rate of learning can cause the expense capability to be disconnected from the basis of the inclination plummet computation. This is due to the algorithm's huge steps, which make it oscillate and exceed the cost function's minimum.

High learning rates can make the algorithm unstable by making it extremely sensitive to even the smallest adjustments to the input data or the initialization of the weights. Due to this, the computation may be challenging to prepare and the exam information may appear subpar.

Slow progress: If the learning rate is too high, the algorithm may also converge to the cost function's minimum more slowly. This is so that the algorithm can perform larger, perhaps less-than-optimal steps and take longer to get to the minimum learning rate.

Question 12

Logistic regression can only be used with data that can be split linearly because it is a linear classification procedure. To put it another way, only feature space data that can be divided by a hyperplane or a straight line are appropriate for logistic regression.

Data that cannot be separated linearly cannot be successfully classified using logistic regression. Non-linear classification techniques such as decision trees, support vector machines (SVMs), and neural networks are a few examples that may perform better in certain circumstances.

Question 13

For classification and regression tasks, two well-known boosting algorithms in machine learning are Adaboost and Gradient Boosting. Although both algorithms use an ensemble approach to boost weak learner performance, they differ significantly in the following key areas:

1. Training approach: Adaboost and Gradient Boosting use different approaches for training the weak learners in the ensemble. Adaboost uses a weighted training approach where it assigns higher weights to the misclassified samples, while Gradient Boosting uses a gradient descent approach where it minimizes the loss function by iteratively adding weak learners.
2. sampling: Each weak learner in Adaboost is trained on the entire dataset, whereas Gradient Boosting selects a subset of the training samples at random for each iteration using a sampling method.
3. The complexity of a model: Gradient Boosting can employ more complex weak learners, such as decision trees with multiple splits, whereas Adaboost typically employs simple weak learners like decision stumps (a decision tree with a single split).
4. Performance: Adaboost may be more prone to overfitting, whereas Gradient Boosting has been found to be more resistant to noisy data and outliers. However, both algorithms have been shown to perform well in practice.

Question 14

A fundamental idea in machine learning, the bias-variance trade-off describes the connection between a model's bias and variance and its capacity to generalise to new data. The term "bias" describes the discrepancy between a model's average prediction and the actual value of the target variable. A model with a strong bias is one that cannot adequately capture the complexity of the data because it is too simplistic. Underfitting, where the model performs badly on both the training and test data, might result from this. Contrarily, variance describes how much a model's predictions would vary if it were trained using a different set of training data. A high variance model is one that has overfitted the training data and is overly complex, which leads to poor generalisation performance on the test data. The bias-variance trade-off results from the fact that adding complexity to a model can decrease bias while increasing variance. On the other hand, a model's bias can increase while its variance decreases when its complexity is reduced. Finding a bias-variance balance that produces a model with little generalisation error on fresh data is the objective.

Question 15

Linear Kernel:

In SVM, the simplest kernel function is the linear kernel. It assumes that the data can be separated using a straight line and is linearly separable. When there are a lot of features for a small number of samples, the linear kernel works well.

Radial basis function (RBF) Kernel:

SVM frequently makes use of the RBF kernel function. It can be used to model non-linearly separable complex decision boundaries in the data. The RBF kernel is a radial basis function that uses feature space distance to calculate the similarity between two data points. It is defined by a parameter known as gamma, which regulates the width of the similarity-modelling Gaussian distribution.

Polynomial Kernel:

In SVM, another kernel function is the polynomial kernel. It can be used to model data-based boundaries for nonlinear decisions. A degree parameter controls the degree of the polynomial used to model the similarity between two data points, defining the polynomial kernel. If the degree is too high, a higher degree polynomial can capture more complex decision boundaries but also lead to overfitting.

SQL

Set 2

1. SELECT * FROM Movie
2. SELECT title
   FROM movies
   ORDER BY runtime DESC
   LIMIT 1
3. SELECT title
   FROM movies
   ORDER BY revenue DESC
   LIMIT 1
4. SELECT title
   FROM movies
   ORDER BY revenue/budget DESC
   LIMIT 1
5. SELECT m.title, p.name, p.gender, c.character_name, c.cast_order
   FROM Movie m
   INNER JOIN Cast c ON m.id = c.movie_id
   INNER JOIN Person p ON c.person_id = p.id
6. SELECT c.name, COUNT(*) as num_movies FROM Movie m
   INNER JOIN Country c ON m.country_id = c.id
   GROUP BY c.name
   ORDER BY num_movies DESC
   LIMIT 1

7. SELECT id, name FROM Genre
8. SELECT l.name, COUNT(*) as num_movies FROM Movie m
   INNER JOIN Language l ON m.language_id = l.id
   GROUP BY l.name
9. SELECT m.title, COUNT(DISTINCT cr.person_id) as num_crew_members, COUNT(DISTINCT ca.person_id) as num_cast_members

   FROM Movie m

   LEFT JOIN Crew cr ON m.id = cr.movie_id

   LEFT JOIN Cast ca ON m.id = ca.movie_id

   GROUP BY m.title

10. SELECT title FROM Movie
    ORDER BY popularity DESC
    LIMIT 10
11. SELECT title, revenue
    FROM Movie
    ORDER BY revenue DESC
    LIMIT 1
12. SELECT title FROM Movie
    WHERE status = 'rumoured'

13. SELECT m.title FROM Movie m
    INNER JOIN Country c ON m.country_id = c.id
    WHERE c.name = 'United States of America'
    ORDER BY m.revenue DESC
    LIMIT 1
14. SELECT mp.movie_id, pc.name
    FROM MovieProductionCompany mp
    INNER JOIN ProductionCompany pc ON mp.production_company_id = pc.id
    ORDER BY mp.movie_id

15. SELECT title FROM Movie
    ORDER BY budget DESC LIMIT
    20

STATISTICS

Set 3

1. D. Expected
2. C. Frequencies
3. C. 6

4. B. Chi squared distribution
5. C. F distribution
6. B. Hypothesis
7. A. Null Hypothesis
8. A. two tailed
9. B. Research Hypothesis
10. A. np