Machine learning

Kavisha Arora

Question 1

R-squared (R^2) is considered a better measure of goodness of fit for a regression model compared to the Residual Sum of Squares (RSS). R-squared is a statistical measure that represents the proportion of variance in the dependent variable that is explained by the independent variables in the regression model. On the other hand, RSS is a measure of the total variance of the residuals or the difference between the observed and predicted values of the dependent variable.

R-squared is a standardized measure that ranges between 0 and 1, with higher values indicating a better fit. In contrast, RSS has no fixed range and can vary depending on the units of the dependent variable.

R-squared can be easily interpreted as the proportion of variance in the dependent variable that is explained by the independent variables, making it a useful metric for comparing the fit of different models. In contrast, RSS has no clear interpretation in terms of the proportion of variance explained..

Question 2

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 are referred to as TSS, ESS, and RSS, respectively.

TSS- Total sum of squares: It is the sum of the squared differences between the actual values of the dependent variable (y) and its mean (). Regardless of the influence of the independent variables, it represent ts the total variation in the dependent variable. 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

In machine learning, regularization is a method for reducing overfitting and increasing a model's generalizability. When a model is overfitted, it fits the training data too well and picks up noise and random fluctuations in the data. This can cause the model to perform poorly on new, untested data.

Regularization helps to address this problem by adding a penalty term to the loss function that the model is trying to minimize. The penalty term imposes a constraint on the model's complexity, limiting the flexibility of the model and reducing the risk 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

A decision tree algorithm's impurity or heterogeneity can be measured using the Gini impurity index. When evaluating the quality of a split in a decision tree, which divides a dataset into subsets based on the values of a feature, it is frequently utilized in classification problems.

The probability that a random example from a subset will be misclassified based on the distribution of classes in that subset is measured by the Gini impurity index. It is between 0 and 1, with 0 representing a pure set in which all examples belong to the same class and 1 representing impure set-in which classes are evenly distributed.

In machine learning frameworks and libraries, the Gini impurity index is frequently used as a metric for evaluating decision trees. Non-experts can easily understand and interpret this straightforward and intuitive measurement.

Question 5

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

When the tree is deep and complex, decision trees are a type of model that is particularly vulnerable to overfitting. This is because decision trees have a high variance, which means they can fit the noise in the training data. This makes it hard for them to generalize to new data that they haven't seen before.

Because they lack a mechanism for controlling the tree's size or complexity, unregularized decision trees are particularly vulnerable to overfitting. A decision tree that has not been regularized will continue to split until each leaf node only contains one example, resulting in a tree that perfectly matches the training data but is unlikely to be able to generalize to new data. Regularized decision trees, which add a penalty term to the cost function that the model is attempting to minimize, can be utilized to prevent overfitting. This penalty term prevents the tree from getting too deep or complex, resulting in a model that is easier to generalize. Pruning, which removes branches from a decision tree that is already in place, can also be used to cut down on the complexity of the tree and avoid overfitting.

Question 6

In machine learning, an ensemble technique is a way to boost a system's overall performance by combining the predictions of multiple individual models. The idea behind ensemble methods is that by combining several models, each model's strengths can be increased while its weaknesses can be reduced.

Bagging: Bagging is the process of combining the predictions of multiple models by using a straightforward average or majority voting scheme and training them on randomly selected subsets of the training data. By introducing randomness into the training process and lowering the variance of the final model, bagging can assist in reducing overfitting.

Boosting: Boosting involves training multiple models sequentially, with each model attempting to correct the previous model's mistakes. A weighted sum or a more complex algorithm is used to combine the individual models' predictions. Boosting can help the final model be more accurate and less biased.

Ensemble techniques are widely used in machine learning and have been shown to be effective in a wide range of applications. They can be used to improve the performance of both classification and regression models, and are particularly effective when the individual models have different strengths and weaknesses.

Question 7

The main difference between bagging and boosting is in how they combine the predictions of multiple models. Bagging involves training independent models on random subsets of the training data and combining their predictions while boosting involves training models in sequence, where each subsequent model tries to correct the errors of the previous model. Both bagging and boosting are ensemble methods that can be effective in improving the performance of machine learning systems.

Bagging can help to reduce overfitting by introducing randomness into the training process and reducing the variance of the final model. Each model in the ensemble is trained independently, and there is no dependence between them. Boosting can help to reduce bias and improve the accuracy of the final model. Each model in the ensemble is trained based on the errors of the previous model, so there is a dependence between them

Question 8

In random forests, the prediction error of the model on unseen data is measured by the out-of-bag (OOB) error, which is only calculated from the training set.

Each tree in a random forest model is trained using a bootstrap sample from the initial training set, which means that some examples are not included in the training set for each tree. The out-of-bag examples for a given tree are the examples that are not used in the training set. The average prediction error of each tree on its corresponding out-of-bag examples is then used to calculate the out-of-bag error. The out-of-bag error provides an impartial estimation of the model's performance on unseen data because each tree only sees a portion of the training data.

Because it estimates the model's generalization error without requiring a separate validation set, the out-of-bag error is a useful performance measure. The random forest model's various hyperparameters, such as the number of trees in the forest and the maximum depth of each tree, can also be evaluated using this method.

Question 9

In machine learning, a common method for evaluating a model's performance on a dataset is K-fold cross-validation. It entails dividing the dataset into K folds of equal size, with K-1 folds serving as the training set and the remaining fold serving as the validation set. This cycle is rehashed K times, with each overlap being utilized as the approval set precisely once. K-overlay cross-approval is utilized to gauge the presentation of a model on new information, and is especially valuable when the dataset is little, or when there is a gamble of overfitting. By utilizing all of the data that are accessible for both training and validation, it enables a more precise estimation of the model's performance.

Stratified K-fold cross-validation and nested K-fold cross-validation are two common variations of K-fold cross-validation. In nested K-fold cross-validation, the folds are created to ensure that each class is represented equally in the training and validation sets. In nested K-fold cross-validation, hyperparameter tuning is performed multiple times.

Question 10

In machine learning, the process of choosing the best values for a machine learning algorithm's hyperparameters is called "hyperparameter tuning." Hyperparameters are model parameters that are set before the training process and are not learned from the data. The learning rate in a neural network, the number of trees in a random forest, or the regularization parameter in a linear regression model are all examples of hyperparameters. The goal of hyperparameter tuning is to make the machine learning model work better with new, untested data. The objective is to locate the hyperparameter values with the highest possible performance for the model. Grid search and random search are two examples of hyperparameter tuning techniques. Grid search entails searching a wide range of values for each hyperparameter in depth before selecting the optimal combination of values. The process of selecting the best value combination through random sampling from the hyperparameter space is known as random search.

Question 11

In gradient descent, a high learning rate can result in a number of problems, including:

Divergence: A large learning rate can make the inclination plummet calculation wander as opposed to joining to the base of the expense capability. This occurs as a result of the algorithm's large steps, which cause it to oscillate around the cost function's minimum and overshoot it.

Instability: The algorithm may also become unstable as a result of a high learning rate, making it sensitive to minute changes in the input data or the initialization of the weights. This can make the calculation hard to prepare and bring about a lackluster showing of the test information.

Slow progress: The algorithm may also converge more slowly to the cost function's minimum if the learning rate is too high. This is because, in comparison to a slower learning rate, the algorithm takes large steps that may not be optimal and takes longer to reach the minimum.

Question 12

Since logistic regression is a linear classification algorithm, it can only be applied to data that can be separated linearly. To put it another way, only data in the feature space that can be separated by a straight line or a hyperplane are suitable for logistic regression.

Logistic regression cannot accurately classify data that cannot be separated linearly. Decision trees, support vector machines (SVMs), and neural networks are just a few examples of non-linear classification algorithms that might work better in these situations.

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

Bias-variance trade-off is a key concept in machine learning that describes the relationship between the bias and variance of a model and its ability to generalize to new data. Bias refers to the difference between the average prediction of a model and the true value of the target variable. A high bias model is one that is too simple and has limited capacity to capture the complexity of the data. This can lead to underfitting, where the model performs poorly on both the training and test data. Variance, on the other hand, refers to the amount by which the predictions of a model would change if it was trained on a different set of training data. A high variance model is one that is too complex and has overfit the training data, resulting in poor generalization performance on the test data. The bias-variance trade-off arises because increasing the complexity of a model can reduce its bias, but increase its variance. Conversely, reducing the complexity of a model can increase its bias, but reduce its variance. The goal is to find a balance between bias and variance that results in a model with low generalization error on new data.

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