Machine Learning

1. Ordinary Least Squares (OLS) regression assesses model fit using three statistics: R-squared, overall F-test, and Root Mean Square Error (RMSE). Sum of Squares Total (SST) and Sum of Squares Error are two sums of squares on which all three are based (SSE). SST and SSE measure how far the data are from the mean and the model-predicted values, respectively. The relationship between the regression model and the mean model can be better understood using different combinations of these two values. The improvement in prediction from the regression model over the mean model accounts for the difference between SST and SSE. R-squared is obtained by dividing that difference by SST. It is the ratio of the regression model's prediction improvement over the mean model. It shows how well the model fits the data. R-squared has the advantageous quality that its scale is clear. It has a value between 0 and 1. Zero means that the proposed model does not outperform the mean model in terms of prediction. One denotes an accurate forecast. R-squared rises proportionately when the regression model is improved. R-squared has the drawback of only getting bigger as more factors are included in the regression model. When predictors are not truly enhancing the model's fit, this increase is fake. To address this, the degrees of freedom in the model are taken into account by a related statistic called Adjusted R-squared.

2. The Total Sum of Squares (TSS) indicates how much the dependent variable varies.
   Total SS = $\sum(\boldsymbol{Y_i - mean\ of\ Y})^2$.

   How much of the variation in the dependent variable was explained by the model is shown by the Explained Sum of Squares(ESS).

   Explained SS = $\sum(\boldsymbol{\hat{Y} - mean\ of\ Y})^2$

   The residual sum of squares (RSS) calculates the variance in a regression model's residuals, often known as the error term. The better the model fits the data, the lower the residual sum of squares; the worse the model fits the data, the higher the residual sum of squares.

   Residual SS=$\sum_{i=1}^{n}(y^i - f(x_i))^2$

   Where:
   yi = the ith value of the variable to be predicted
   f(xi) = predicted value of yi
   n = upper limit of summation
3. The term "regularisation" describes methods for calibrating machine learning models to reduce the adjusted loss function and avoid overfitting or underfitting. Overfitting occurs when a model performs exceptionally well on training data but poorly on test

data (fresh data). Underfitting is the term used to describe a model's inability to generalise successfully to new data after poorly learning the patterns in the training data. We can properly fit our machine learning model on a particular test set using regularisation, which lowers the mistakes in the test set.

4. To identify how the attributes of a dataset should split the nodes to construct the tree, decision trees are built using the Gini Impurity measurement. A dataset's Gini Impurity, which is actually a number between 0-0.5, represents how likely it is that new, random data will be incorrectly classified if it is assigned a random class label based on the dataset's class distribution.
5. Overfitting is a problem with decision trees, especially when the tree is quite deep. This is because the level of specificity we consider results in a smaller sample of events that are consistent with the prior hypotheses. Conclusions drawn from this small sample may not be valid. Setting a maximum depth is one strategy for addressing this problem. As always, this will reduce our risk of overfitting at the expense of bias-related inaccuracy. The ideal situation would be to reduce both bias-related and variance-related errors. This issue is well-mitigated by random forests. Simply put, a random forest is a group of decision trees with their findings combined into a single output. They are so potent models because they can control overfitting without significantly increasing bias-related error.
6. A machine learning technique called ensemble techniques combines multiple base models to create a single, ideal predictive model. Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would.
7. By creating additional data for training from a dataset by combining repeats and combinations to create multiple sets of the original data, the bagging strategy lowers prediction variance. Boosting is an iterative technique for modifying the weight of an observation in accordance with the previous classification. It tries to give an observation more weight if it was incorrectly classified. In general, boosting produces accurate predictive models.
8. Generally, building a reliable system that will function effectively with new, untested data is essential in machine learning and data science. To achieve this generalisation, there are numerous alternative strategies and techniques available. One of these techniques for verifying the machine learning model is out-of-bag error. This strategy employs bootstrapping in the random forest. It is highly possible that we won't choose every sample from the original data set because bootstrapping samples the data with the option of selecting one sample more than once. Therefore, one wise choice would be to utilise these out-of-bag samples, which are unselected samples. Out-of-bag error is the term used to describe the mistake that was achieved on these samples.
9. A resampling technique called cross-validation is used to assess machine learning models on a small data sample. The process contains a single parameter, k, that designates how many groups should be created from a given data sample. As a result, the process is frequently referred to as k-fold cross-validation. When a particular number for k is selected, it may be substituted for k in the model's reference, such as

when k=10 is used to refer to cross-validation by a 10-fold factor. In applied machine learning, cross-validation is mostly used to gauge how well a machine learning model performs on untrained data. That is, to use a small sample to assess how the model will generally perform when used to generate predictions on data that was not utilised during the model's training. It is a well-liked technique since it is easy to comprehend and typically yields a less biased or overly optimistic assessment of the model skill than other techniques, including a straightforward train/test split.

10. Finding a set of ideal hyperparameter values for a learning algorithm and using this tuned algorithm on any data set is hyperparameter tuning. The model's performance is maximised by using that set of hyperparameters, which minimises a predetermined loss function and resulting in better results with fewer errors.
11. How quickly the model adapts to the challenge is determined by the learning rate. Given the smaller changes to the weights made with each update, smaller learning rates necessitate more training epochs, whereas bigger learning rates produce quick changes and necessitate fewer training epochs. While a learning rate that is too modest can make the process stall, one that is too great can lead to the model converging too soon to an unsatisfactory result. Carefully choosing the learning rate is a challenge when developing deep learning neural networks. It might be the model's most crucial hyperparameter.
12. A linear decision surface of logistic regression allows it to be utilised as a linear classifier. In order to distinguish between observations that are members of one class and all other observations that are not members of that class, it is utilised to create a hyperplane in feature space. This results in a linear decision boundary.
13. Differentiate between Adaboost and Gradient Boosting

    Loss-Function-Different loss functions are employed in the boosting procedure. When using Adaptive Boosting or AdaBoost, it minimises the exponential loss function, which could make the algorithm more susceptible to outliers. You can use any differentiable loss function with gradient boosting. Compared to AdaBoost, the Gradient Boosting algorithm is more resistant to outliers.

    Flexibility-The first boosting algorithm with a specific loss function was called AdaBoost. Gradient Boosting, on the other hand, is a general technique that helps in the search for approximations to the additive modelling issue. Gradient Boosting is hence more adaptable than AdaBoost.
14. The bias-variance trade-off is a characteristic of a model in statistics and machine learning that allows the variance of the parameter estimated across samples to be decreased by increasing the bias in the estimated parameters. The conflict that arises from attempting to simultaneously reduce these two kinds of error, which prohibit supervised learning algorithms from generalising outside their training set, is known as the bias-variance dilemma or bias-variance problem. The bias error results from incorrect learning algorithm assumptions. An algorithm with a high bias may fail to recognise important relationships between features and goal outputs (underfitting).

    The variance results from the training set's sensitivity to slight variations. An algorithm that simulates the random noise in the training data may provide high variation (overfitting). The bias-variance decomposition is a technique for assessing the

expected generalisation error of a learning algorithm with respect to a certain problem as the product of three terms: bias, variance, and an amount known as the irreducible error, which is caused by noise in the problem itself.

15. **Linear Kernel**-It is the most fundamental kind of kernel and is often of a one-dimensional kind. When there are many features, it turns out to be the best feature. For text classification issues, the linear kernel is typically favoured because the majority of these classification issues can be divided linearly. The speed of linear kernel functions is superior to other functions.

    RBF Kernel-It ranks among the most popular and often used kernel functions in SVM. It's frequently used with non-linear data. When there is no prior understanding of the data, it helps to make the right separation.

    Polynomial Kernel-It is a broader illustration of the linear kernel. Because it is less precise and efficient than other kernel functions, it is not as widely used.

Statistics

1. D
2. C
3. C
4. B
5. C
6. B
7. A
8. A
9. B
10. A

SQL

1. SELECT * FROM movie

2. SELECT title
   FROM movie
   WHERE runtime= (SELECT MAX (runtime) FROM movie)

3. SELECT title
   FROM movie
   WHERE revenue= (SELECT MAX (revenue) FROM movie)

4. SELECT title
   FROM movie

WHERE budget= (SELECT MAX(budget) FROM movie)

5.

6. SELECT company_name, year(order_date),count(orders.order_id) as total_order
   FROM customers
   INNER JOIN orders
   ON customers.order_id = orders.order_id
   GROUP BY company_name, year(order_date)
   order by total_order desc limit 1

7. SELECT genre_id,genre_name
   FROM genre

8. SELECT language.language_name, COUNT(movie_languages.language_id)
   FROM language
   INNER JOIN movie_languages
   ON language.language_id=movie_languages.language_id
   GROUP BY (movie_languages.language_id)

9. SELECT movie.title,COUNT(movie_cast.movie_id),COUNT(movie_crew.movie_id)
   FROM movie
   JOIN (movie_cast JOIN movie_crew ON movie_cast.movie_id=movie_crew.movie_id)
   ON movie.movie_id=movie_cast.movie_id
   GROUP BY movie_id

10. SELECT title
    FROM movie
    ORDER BY(popularity)DESC
    LIMIT 10;

11. SELECT *FROM (SELECT title, revenue, DENSE_RANK () OVER (ORDER BY revenue DESC)3
    FROM movie)

12. SELECT title
    FROM movie
    WHERE movie_status==” rumoured”

13. SELECT movie.title
    FROM movie
    JOIN (production_country JOIN country ON
    production_country.country_id=country.country_id)
    ON movie.movie_id=production_country.movie_id
    WHERE country.country_name==”United States of America” AND movie.revenue= (SELECT
    MAX (revenue) FROM movie)

14. SELECT movie_company.movie_id,production_company.company_name
    FROM movie_company
    INNER JOIN production_company
    ON movie_company.company_id=production_company.company_id

15. SELECT title,budget
    FROM movie
    ORDER BY (budget)DESC
    LIMIT 20;