MACHINE LEARNING

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?

Ans = Residual Sum of Squares :- The RSS measures the amount of error remaining between the regression function and the data set after the model has been run. A smaller RSS figure represents a regression function that is well-fit to the data.

In general terms, the sum of square is a statistical technique used in regression analysis to determine the dispersion of data points. In a regression analysis, the goal is to determine how well a data series can be fitted to a function that might help to explain how the data series was generated. The sum of squares is used as a mathematical way to find the function that best fits (varies least) from the data.

- R-squared :- R-squared is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. Whereas correlation explains the strength of the relationship between an independent and dependent variable, R-squared explains to what extent the variance of one variable explains the variance of the second variable. So, if the R^2 of a model is 0.50, then approximately half of the observed variation can be explained by the model's inputs. R-Squared is a statistical measure of fit that indicates how much variation of a dependent variable is explained by the independent variable(s) in a regression 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.

Ans = TSS :- The total sum of squares is a variation of the values of a dependent variable from the sample mean of the dependent variable. Essentially, the total sum of squares quantifies the total variation in a sample.

ESS :- n statistics the explained sum of squares alternatively known as the models sum of square or sum of squares due to regression (**SSR** – not to be confused with (RSS) or sum of squares of errors), is a quantity used in describing how well a model, often a regression models represents the data being modelled. In particular, the explained sum of squares measures how much variation there is in the modelled values and this is compared to the (TSS), which measures how much variation there is in the observed data, and to the , which measures the variation in the error between the observed data and modelled values.

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

Ans = When we use regression models to train some data. There is a good chance that the models will overfit the given training data set regularization helps sort this overfitting problem by restricting the degress of freedom of a given equation simily reducing the number of degress function by reducing their corresponding weights.

In a linear equation we do not want huge weight coefficient as a chance in weight can a large different for the dependent variable (y) so regularization constrints the weight of such feature to avoid overfitting

To regularize the models a shrinkage penalty is added to be cost function let's see different type of regularization in regression.

1) LASSO
2) RIDGE
3) ELASTICNET (LESS POPULAR)

Ridge regression shrinks the coefficient for those predictors which contribute very less in the models but have huge weights very close to zero.but it never makes them exactly zero. thuis the final models will still contain all those predictors thoughts with less weight this doesn’t help in interpreting different with ridge regression in lasso the L1 penalty does reduce some coefficient exactly to zero when we use a sufficient large turning parameters so in addition to regularzing lasso also perform feature selection.

4.What is Gini–impurity index?

Ans = The Gini impurity measure is one of the methods used in decision tree algorithms to decide the optimal split from a root node, and subsequent splits. Before moving forward you may want to review making decision with trees To put it into context, a decision tree is trying to create sequential questions such that it partitions the data into smaller groups. Once the partition is complete a predictive decision is made at this terminal node (based on a frequency). Suppose we have a list of observations, that indicates if a person decided to stay home from work. We also have two features, namely if they are sick and their temperature. We need to choose which feature, emotion or temperature, to split the data on. A Gini Impurity measure will help us make this decision.

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

Ans = Hands-on implementation of pre-pruning, post-pruning, and ensemble of Decision Trees Decision Trees are a non-parametric supervised machine learning approach for classification and regression tasks. Overfitting is a common problem, a data scientist needs to handle while training decision tree models. Comparing to other machine learning algorithms, decision trees can easily overfit. Overfitting refers to the condition when the model completely fits the training data but fails to generalize the testing unseen data. Overfit condition arises when the model memorizes the noise of the training data and fails to capture important patterns. A perfectly fit decision tree performs well for training data but performs poorly for unseen test data. If the decision tree is allowed to train to its full strength, the model will overfit the training data. There are various techniques to prevent the decision tree model from overfitting. In this article, we will discuss 3 such techniques.

6. What is an ensemble technique in machine learning?

Ans = We regularly comes across many game shows on television and you mush have noticed an optior of ‘audience poll’ most of the times a contestant goes with the option which has the highest vote from the audience and most of the times they win we can generalize this in real life as well where taking opinion from a majority of people is more preferred than the opinion of a single person ensemble technique has a similar underlying idea where we aggregate prediction from a group of than most of the obtained using a single prediction such algorithm are called ensemble me those and such predictors are called Ensembles.

Let’s suppose we have ‘n’ predictors:

Z1, Z2,Z3…………..Zn with a standard deviation of var(z)=^2

If we use single predictors z1,z2,z3…….zn the variance associated with each will be but the expected value will be the average of all the prediction

Let’s consider the average of the prediction:

If we use as the predictor then the expected value still remain the same but see the variance now:

So the expected value remained but variance decrease when we use average off all the predictors.

This is why taking means is presefferred over using single predictors.

Ensemble method take multiple small models and combine predictior to obtain a more powerfull predictor power.

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

Ans = Now that we have thoroughly described the concepts of Bagging and Boosting, we have arrived at the end of the article and can conclude how both are equally important in Data Science and where to be applied in a model depends on the sets of data given, their simulation and the given circumstances. Thus, on the one hand, in a Random Forest model, Bagging is used, and the AdaBoost model implies the Boosting algorithm. A machine learning model’s performance is calculated by comparing its training accuracy with validation accuracy, which is achieved by splitting the data into two sets: the training set and validation set. The training set is used to train the model, and the validation set is used for evaluation.

From the dataset, bagging creates extra data for training. Random sampling and substitution from the original dataset is used to achieve this. In each new training data set, sampling with replacement may repeat certain observations. Every Bagging element has the same chance of emerging in a fresh dataset. Multiple models are trained in parallel using these multi datasets. It is the average of all the forecasts from several ensemble models. When determining classification, the majority vote obtained through the voting process is taken into account. Bagging reduces variation and fine-tunes the prediction to a desired result. Bagging and boosting are ensemble strategies that aim to produce N learners from a single learner. They sample at random and create many training data sets. They arrive at their final decision by averaging N learners' votes or selecting the voting rank of the majority of them. They reduce variance and increase stability while reducing errors.

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

Ans = Random forest is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

Since the random forest model is made up of multiple decision trees, it would be helpful to start by describing the decision tree algorithm briefly. Decision trees start with a basic question, such as, “Should I surf?” From there, you can ask a series of questions to determine an answer, such as, “Is it a long period swell?” or “Is the wind blowing offshore?”. These questions make up the decision nodes in the tree, acting as a means to split the data. Each question helps an individual to arrive at a final decision, which would be denoted by the leaf node. Observations that fit the criteria will follow the “Yes” branch and those that don’t will follow the alternate path. Decision trees seek to find the best split to subset the data, and they are typically trained through the Classification and Regression Tree (CART) algorithm. Metrics, such as Gini impurity, information gain, or mean square error (MSE), can be used to evaluate the quality of the split.

9. What is K-fold cross-validation?

Ans = Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train and test split.

Cross validation is an evaluation method used in machine learning to find out how well your machine learning model can predict the outcome of unseen data. It is a method that is easy to comprehend, works well for a limited data sample and also offers an evaluation that is less biased, making it a popular choice. The data sample is split into ‘k’ number of smaller samples, hence the name: K-fold Cross Validation.

You may also hear terms like fourfold cross validation, or tenfold cross validation, which essentially means that the sample data is being split into four or ten smaller samples respectively.

1)First, shuffle the dataset and split into k number of subsamples. (It is important to try to make the subsamples equal in size and ensure k is less than or equal to the number of elements in the dataset).

2)In the first iteration, the first subset is used as the test data while all the other subsets are considered as the training data.

3)Train the model with the training data and evaluate it using the test subset. Keep the evaluation score or error rate, and get rid of the model.

4)Now, in the next iteration, select a different subset as the test data set, and make everything else (including the test set we used in the previous iteration) part of the training data.

5)Re-train the model with the training data and test it using the new test data set, keep the evaluation score and discard the model.

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

Ans = In machine learning, we need to differentiate between parameters and hyperparameters. A learning algorithm learns or estimates model parameters for the given data set, then continues updating these values as it continues to learn. After learning is complete, these parameters become part of the model. For example, each weight and bias in a neural network is a parameter. Hyperparameters , on the other hand, are specific to the algorithm itself, so we can’t calculate their values from the data. We use hyperparameters to calculate the model parameters. Different hyperparameter values produce different model parameter values for a given data set.

Hyperparameters type :-

- **Number of hidden layers**: It’s a trade-off between keeping our neural network as simple as possible (fast and generalized) and classifying our input data correctly. We can start with values of four to six and check our data’s prediction accuracy when we increase or decrease this hyperparameter.

- **Number of nodes/neurons per layer**: More isn't always better when determining how many neurons to use per layer. Increasing neuron count can help, up to a point. But layers that are too wide may memorize the training dataset, causing the network to be less accurate on new data.

- **Learning rate**: Model parameters are adjusted iteratively — and the learning rate controls the size of the adjustment at each step. The lower the learning rate, the lower the changes to parameter estimates are. This means that it takes a longer time (and more data) to fit the model — but it also means that it is more likely that we actually find the minimum loss.

- **Momentum**: Momentum helps us avoid falling into local minima by resisting rapid changes to parameter values. It encourages parameters to keep changing in the direction they were *already* changing, which helps prevent zig-zagging on every iteration. Aim to start with low momentum values and adjust upward as needed.

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

Ans = The process of repeatedly nudging an input of a function by some multiple of the negative gradient is called gradient descent. It’s a way to converge towards some local minimum of a cost function basically valley in a graph. According to Wikipedia is a first-order iterative optimization algorithm for finding the minimum of a function. To find a local minimum or minimum cost of a function using gradient descent, one takes steps proportional to the negative of the of the function at the current point.

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

Ans = Logistic regression is one such regression algorithm which can be used for performing classification problem it calculates the probability that a given value belongs to a specific class. If the probability is more than 50% it assign the value I that particular class value else if the to the other class therefore we can say that logistic regression acts as a binary classifier.

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time.

From this example, it can be inferred that linear regression is not suitable for classification problem. Linear regression is unbounded, and this brings logistic regression into picture. Their value strictly ranges from 0 to 1

13. Differentiate between Adaboost and Gradient Boosting.

Ans = AdaBoost or Adaptive Boosting is the first boosting ensemble model.The method automatically adjusts its parameters to the data based on the actual performance in the current iteration. Meaning, both the weights for re-weighting the data and the weights for the final aggregation are re-computed iteratively.

In practice, this boosting technique is used with simple classification trees or stumps as base-learners, which resulted in improved performance compared to the classification by one tree or other single base-learner.

Gradient Boost is a robust machine learning algorithm made up of Gradient descent and Boosting. The word 'gradient' implies that you can have two or more derivatives of the same function. Gradient Boosting has three main components: additive model, loss function and a weak learner.

The technique yields a direct interpretation of boosting methods from the perspective of numerical optimisation in a function space and generalises them by allowing optimisation of an arbitrary loss function.

The technique of Boosting uses various loss functions. In case of Adaptive Boosting or AdaBoost, it minimises the exponential loss function that can make the algorithm sensitive to the outliers. With Gradient Boosting, any differentiable loss function can be utilised. Gradient Boosting algorithm is more robust to outliers than AdaBoost.

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

Ans = Whenever we discuss model prediction, it’s important to understand prediction errors (bias and variance). There is a tradeoff between a model’s ability to minimize bias and variance. Gaining a proper understanding of these errors would help us not only to build accurate models but also to avoid the mistake of overfitting and underfitting. So let’s start with the basics and see how they make difference to our machine learning Models. Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data. Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn’t seen before. As a result, such models perform very well on training data but has high error rates on test data. In supervised learning, happens when our model captures the noise along with the underlying pattern in data. It happens when we train our model a lot over noisy dataset. These models have low bias and high variance. These models are very complex like Decision trees which are prone to overfitting.

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

Ans = Linear :- Linear machine learning algorithms assume a linear relationship between the features and the target variable. In this article, we’ll discuss several linear algorithms and their concepts. Here’s a glimpse into what you can expect to learn:

1)Types of linear ML algorithms.

2)Assumptions of linear algorithms.

3)The difference between various linear machine learning algorithms.

4)How to interpret the results of linear algorithms.

5)When to use different linear algorithms.

RBF :- You’re working on a Machine Learning algorithm like Support Vector Machines for non-linear datasets and you can’t seem to figure out the right feature transform or the right kernel to use. Well, fear not because Radial Basis Function (RBF) Kernel is your savior.

SVM:- SVM is an algorithm that has shown great success in the field of classification. It separates the data into different categories by finding the best hyperplane and maximizing the distance between points. To this end, a will be introduced to demonstrate how it works with support vector machines.

Kernel functions are a very powerful tool for exploring high-dimensional spaces. They allow us to do linear discriminants on nonlinear manifolds, which can lead to higher accuracies and robustness than traditional linear models alone.