# MACHINE LEARNING

1. Which of the following in sk-learn library is used for hyper parameter tuning? A) GridSearchCV() B) RandomizedCV() C) K-fold Cross Validation D) All of the above

Ans = A) GridSearchCV() B) RandomizedCV()

2. In which of the below ensemble techniques trees are trained in parallel? A) Random forest B) Adaboost C) Gradient Boosting D) All of the above

Ans = A) Random forest

3. In machine learning, if in the below line of code: sklearn.svm.SVC (C=1.0, kernel='rbf', degree=3) we increasing the C hyper parameter, what will happen? A) The regularization will increase B) The regularization will decrease C) No effect on regularization D) kernel will be changed to linear

Ans = B) The regularization will decrease

4. Check the below line of code and answer the following questions: sklearn.tree.DecisionTreeClassifier(*criterion='gini',splitter='best',max_depth=None, min_samples_split=2) Which of the

following is true regarding max_depth hyper parameter? A) It regularizes the decision tree by limiting the maximum depth up to which a tree can be grown. B) It denotes the number of children a node can have. C) both A & B D) None of the above

Ans = A) It regularizes the decision tree by limiting the maximum depth up to which a tree can be grown.

5. Which of the following is true regarding Random Forests? A) It's an ensemble of weak learners. B) The component trees are trained in series C) In case of classification problem, the prediction is made by taking mode of the class labels predicted by the component trees. D)None of the above

Ans = C) In case of classification problem, the prediction is made by taking mode of the class labels predicted by the component trees

6. What can be the disadvantage if the learning rate is very high in gradient descent? A) Gradient Descent algorithm can diverge from the optimal solution. B) Gradient Descent algorithm can keep oscillating around the optimal solution and may not settle. C) Both of them D) None of them

Ans = A) Gradient Descent algorithm can diverge from the optimal solution.

7. As the model complexity increases, what will happen? A) Bias will increase, Variance decrease B) Bias will decrease, Variance increase C)both bias and variance increase D) Both bias and variance decrease.

Ans = A) Bias will increase, Variance decrease

8. Suppose I have a linear regression model which is performing as follows: Train accuracy=0.95 and Test accuracy=0.75 Which of the following is true regarding the model? A) model is underfitting B) model is overfitting C) model is performing good D) None of the above

Ans = B) model is overfitting

## Q9 to Q15 are subjective answer type questions, Answer them briefly.

9. Suppose we have a dataset which have two classes A and B. The percentage of class A is 40% and percentage of class B is 60%. Calculate the Gini index and entropy of the dataset.

Ans = Decision trees are often used while implementing machine learning algorithms. The hierarchical structure of a decision tree leads us to the final outcome by traversing through the nodes of the tree. Each node consists of an attribute or feature which is further split into more nodes as we move down the tree. But how do we decide To decide this, and how to split the tree, we use splitting measures like Gini Index, Information Gain, etc. In this blog, we will learn all about the Gini Index, including the use of Gini Index to split a decision tree.

Gini Index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen.The very essence of decision trees resides in dividing the entire dataset into a tree-like vertical information structure so as to divide the different sections of the information with root nodes at the top. In the decision tree model, each node is an attribute or the feature that contains necessary information (going sequentially downward) for the decision tree model. These are the necessary points to keep in mind while deciding each node of the decision tree model

10. What are the advantages of Random Forests over Decision Tree?

Ans = There are many different ways in which machine learning models make decisions. Decision Trees and Random Forests are two of the most common decision-making processes used in ML. Hence, there is always confusion, comparison, and debate about Random Forest vs Decision Tree. They both have their advantages, disadvantages, and specific use case, based on which we can choose the right one specific to our requirement and project. This article will give you all the information required to make this choice.

Let's start with a thought experiment that will illustrate the difference between a decision tree and a random forest model.Suppose a bank has to approve a small loan amount for a customer, and the bank needs to make a decision quickly. The bank checks the person's credit

history and financial condition and finds that they haven't re-paid the older loan yet. Hence, the bank rejects the application. But here's the catch – the loan amount was very small for the bank's immense coffers, and they could have easily approved it in a very low-risk move. Therefore, the bank lost the chance of making some money.

Now, another loan application comes in a few days down the line, but this time the bank comes up with a different strategy – multiple decision-making processes. Sometimes it checks for credit history first, and sometimes it checks for the customer's financial condition and loan amount first. Then, the bank combines the results from these multiple decision-making processes and decides to give the loan to the customer. Even if this process took more time than the previous one, the bank profited using this method. This is a classic example where collective decision-making outperformed a single decision-making process. Now, here's my question to you – do you know what these two processes represent? Machine Learning is the sub-branch of Artificial Intelligence. These are decision trees and a random forest! We'll explore this idea in detail here, dive into the major differences between these two methods, and answer the key question – which machine learning algorithm should you go with Random Forest.

11. What is the need of scaling all numerical features in a dataset? Name any two techniques used for scaling.

Ans = Machine learning is like making a mixed fruit juice. If we want to get the best-mixed juice, we need to mix all fruit not by their size but based on their right proportion. We just need to remember apple and strawberry are not the same unless we make them similar in some context to compare their attribute. Similarly, in many machine learning algorithms, to bring all features in the same standing, we need to do scaling so that one significant number doesn't impact the model just because of their large magnitude.

Feature scaling in machine learning is one of the most critical steps during the pre-processing of data before creating a machine learning model. Scaling can make a difference between a weak machine learning model and a better one. The most common techniques of feature scaling are Normalization and Standardization. Normalization is used when we want to bound our values between two numbers, typically, between. While Standardization transforms the data to have zero mean and a variance of 1, they make our data . Refer to the below diagram, which shows how data looks after scaling in the X-Y plane. Machine learning algorithm just sees number — if there is a vast difference in the range say few ranging in thousands and few ranging in the tens, and it makes the underlying assumption that higher ranging numbers have superiority of some sort. So these more significant number starts playing a more decisive role while training the model. The machine learning algorithm works on numbers and does not know what that number represents. A weight of 10 grams and a price of 10 dollars represents completely two different things — which is a no brainer for humans, but for a model as a feature, it treats both as same. Suppose we have two features of weight and price, as in the below table. The "Weight" cannot have a meaningful

comparison with the “Price.” So the assumption algorithm makes that since “Weight” > “Price,” thus “Weight,” is more important than “Price.”

12. Write down some advantages which scaling provides in optimization using gradient descent algorithm.

Ans = Gradient Decent is the most common optimization algorithm in *machine learning* and *deep learning*. It is a first-order optimization algorithm. This means it only takes into account the first derivative when performing the updates on the parameters. On each iteration, we update the parameters in the opposite direction of the gradient of the objective function  the parameters where the gradient gives the direction of the steepest ascent. The size of the step we take on each iteration to reach the local minimum is determined by the learning rate α. Therefore, we follow the direction of the slope downhill until we reach a local minimum. In this article, we’ll cover gradient descent algorithm and its variants: *Batch Gradient Descent, Mini-batch Gradient Descent, and Stochastic Gradient Descent*. Let’s first see how gradient descent works on logistic regression before going into the details of its variants. For the sake of simplicity, let’s assume that the logistic regression model has only two parameters: weight *w* and bias *b*.

1) We can use fixed learning rate during training without worrying about learning rate decay.

2) It has straight trajectory towards the minimum and it is guaranteed to converge in theory to the global minimum if the loss function is convex and to a local minimum if the loss function is not convex.

3) It has unbiased estimate of gradients. The more the examples, the lower the standard error.

13. In case of a highly imbalanced dataset for a classification problem, is accuracy a good metric to measure the performance of the model. If not, why?

Ans = Classification accuracy is a metric that summarizes the performance of a classification model as the number of correct predictions divided by the total number of predictions. It is easy to calculate and intuitive to understand, making it the most common metric used for evaluating classifier models. This intuition breaks down when the distribution of examples to classes is severely skewed. Intuitions developed by practitioners on balanced datasets, such as 99 percent representing a skillful model, can be incorrect and dangerously misleading on imbalanced classification predictive modeling problems. In this tutorial, you will discover the failure of classification accuracy for imbalanced classification problems.

Classification predictive modeling involves predicting a class label given examples in a problem domain. The most common metric used to evaluate the performance of a classification predictive model is classification accuracy. Typically, the accuracy of a predictive model is good (above 90% accuracy), therefore it is also very common to summarize the performance of a model in terms of the error rate of

the mode A confusion matrix is a summary of the predictions made by a classification model organized into a table by class. Each row of the table indicates the actual class and each column represents the predicted class. A value in the cell is a count of the number of predictions made for a class that are actually for a given class. The cells on the diagonal represent correct predictions, where a predicted and expected class align.

The confusion matrix provides more insight into not only the accuracy of a predictive model, but also which classes are being predicted correctly, which incorrectly, and what type of errors are being made. The simplest confusion matrix is for a two-class classification problem, with negative (class 0) and positive (class 1) classes.

## 14. What is “f-score" metric? Write its mathematical formula

Ans = The F-score, also called the F1-score, is a measure of a model’s accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into ‘positive’ or ‘negative’. The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model’s precision and recall.

The F-score is commonly used for evaluating information retrieval systems such as search engines, and also for many kinds of machine learning models, in particular in nature language processing.It is possible to adjust the F-score to give more importance to precision over recall, or vice-versa. Common adjusted F-scores are the F0.5-score and the F2-score, as well as the standard F1-score.

Classification accuracy is the total number of correct predictions divided by the total number of predictions made for a dataset. As a performance measure, accuracy is inappropriate for imbalanced classification problems. The main reason is that the overwhelming

number of examples from the majority class (or classes) will overwhelm the number of examples in the minority class, meaning that even unskillful models can achieve accuracy scores of 90 percent, or 99 percent, depending on how severe the class imbalance happens to be. An alternative to using classification accuracy is to use precision and recall metrics. In this tutorial, you will discover how to calculate and develop an intuition for precision and recall for imbalanced classification.

15.What is the difference between fit(), transform() and fit_transform()?

Ans = Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It is characterized by a clean, uniform, and streamlined API. A benefit of this uniformity is that once you understand the basic use and syntax of Scikit-Learn for one type of model, switching to a new model or algorithm is straightforward. While working with a dataset for a model we go through data transformation techniques to retrieve strategic information efficiently and easily. Scikit-learn provides a library of transformers, which may clean, reduce (dimensionality reduction), expand or generate feature representations.

These are represented by classes with fit() ,transform() and fit_transform() methods. These methods are used according to the type of object you wish to use it for. Whether you want to use it for "transformers" or "models". *After reading this article you will have a*

*better understanding of fit(), transform() and the fit_transform() method and where and when to use them.*

So let's imagine you are planning to go to a party and you are given a dress code to follow. Firstly you will examine your wardrobe and based on the dress code, you would plan your outfit. Finally, just before your party you would wear the outfit and go. Here, you could think of planning the outfit based on the dress code as fit() method and wearing the outfit and going to the party as the transform() method. Hope you understood this analogy To handle missing values in the training data, we use the Simple Imputer class. Firstly, we use the fit() method on the training data to calculate the mean of the training data and then use transform() method on the same data. This will convert the null values to mean values (calculated using the fit() method).