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 and B

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

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

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

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

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: C

7. As the model complexity increases, what will happen?
   A) Bias will increase, Variance decrease B) Bias will decrease, Variance increase

MACHINE LEARNING

C)both bias and variance increase D) Both bias and variance decrease.

ANS: B

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

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.

The percentage of class A is = 40%

The percentage of class B is =60%

Gini index = 1 - (p_A^2 + p_B^2)

In this case, p_A = 0.4 and p_B = 0.6, so the Gini index is: 1-_(p_A^2 + P_B^2)

Gini index = 1 - (0.4^2 + 0.6^2) = 1 - (0.16 + 0.36) = 1 - 0.52 = 0.48

Entropy = -p_A * log2(p_A) - p_B * log2(p_B)

Entropy = -0.4 * log2(0.4) - 0.6 * log2(0.6) = 0.97095

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

Random Forests are an ensemble method that utilizes multiple decision trees, whereas a single decision tree is used in traditional decision tree algorithm. The main advantages of Random Forests over Decision Tree are:

Improved accuracy: By averaging or combining the results of multiple decision trees, Random Forests can often produce more accurate results than a single decision tree.

Reduced overfitting: One of the main problems with decision trees is that they can easily overfit the training data, especially when the tree becomes very deep. Random Forests addresses this problem by averaging the results of many smaller trees, which reduces the chance of overfitting.

Better handling of missing values: Random Forests can handle missing data better than decision trees, as it takes into account the uncertainty of the missing values.

Handling of non-linear relationships: Random Forests can model non-linear relationships between the input variables and the output variable better than decision trees.

Variable importance: Random Forests also provides an easy way to determine the importance of each

MACHINE LEARNING

feature in the dataset, which can be helpful in feature selection.

Better interpretability: Random Forest provide more interpretability when compared to decision tree as they provide variable importance which can be used to understand the feature importance in the model

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

Scaling is a preprocessing step that is performed on numerical features in a dataset to ensure that all features are on the same scale. The need of scaling is mainly to ensure that the algorithm doesn't treat one feature as more important than another simply because the values of one feature are on a much larger scale than the other. This is important because many machine learning algorithms use distance-based metrics and are sensitive to the scale of the features. Scaling also helps in faster convergence of optimization algorithms.

Two common techniques used for scaling are:

Min-Max Scaling: This technique scales the values of the feature such that they lie between 0 and 1. It is done by subtracting the minimum value of the feature from each value and then dividing the result by the range (maximum value - minimum value) of the feature.

Standardization: This technique scales the values of the feature such that they have a mean of 0 and a standard deviation of 1. It is done by subtracting the mean of the feature from each value and then dividing the result by the standard deviation of the feature.

It is important to note that, the scaling technique used can depend on the type of problem, data and algorithm.

If we consider a car dataset with below values:

Here age of car is ranging from 5years to 20years, whereas Distance Travelled is from 10000km to 50000km. When we compare both the ranges, they are at very long distance from each other. The machine learning algorithm thinks that the feature with higher range values is most important while predicting the output and tends to ignore the feature with smaller range values. This approach would give wrong predictions.

To avoid such wrong predictions, the range of all features are scaled so that each feature contributes proportionately and model performance improves drastically.

Another reason for feature scaling is that if the values of a dataset are small then the model learns fast compared the unscaled data. Example, in gradient decent, to minimize the cost function, if the range of values is small then the algorithm converges much faster

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

Scaling can provide several advantages when used in optimization algorithms such as gradient descent:

MACHINE LEARNING

Faster convergence: Scaling can greatly speed up the convergence of gradient descent by reducing the range of the feature values. This means that the optimization algorithm can take larger steps in the direction of the gradient, which can make the optimization process faster.

Better optimization: Scaling can also improve the optimization process by ensuring that all features are on the same scale. This is important because gradient descent is sensitive to the scale of the features, and if some features have much larger values than others, the optimization process can be slowed down or even become stuck in a suboptimal solution.

Better generalization: Scaling can also improve the generalization of the model by reducing the chance of overfitting. Because gradient descent is sensitive to the scale of the features, unscaled features can introduce noise into the optimization process and lead to overfitting.

Better handling of missing data: Scaling can help in better handling of missing data as some algorithms can be sensitive to missing data, scaling can ensure that missing data doesn't affect the optimization process.

Handling of non-linear relationships: Scaling can also help in handling non-linear relationships in the data. As gradient descent is sensitive to the scale of the features, scaling can help in ensuring that all features are on the same scale and hence improve the optimization process.

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?

In case of a highly imbalanced dataset, accuracy is generally not a good metric to measure the performance of the model. This is because accuracy is a ratio of correctly classified samples to the total number of samples, but in a highly imbalanced dataset, even a model that always predicts the majority class will achieve high accuracy.

For example, if the dataset has 90% of class A and 10% of class B, a model that always predicts class A will achieve an accuracy of 90%, even though it is not performing well on class B.

In such cases, other metrics that are less sensitive to class imbalance should be used, such as:

Confusion Matrix: A confusion matrix gives a more detailed view of the model performance by showing the number of true positives, true negatives, false positives, and false negatives.

Precision: Precision is the ratio of true positives to the number of true positives plus false positives. It measures the proportion of positive predictions that are actually correct.

Recall (Sensitivity or True Positive Rate): Recall is the ratio of true positives to the number of true positives plus false negatives. It measures the proportion of actual positive cases that are correctly identified.

F1-Score: F1-score is a harmonic mean of precision and recall. It is a good metric to use when you want to balance precision and recall.

MACHINE LEARNING

Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC): AUC-ROC can be used as an evaluation metric for binary classification problems, it measures the ability of the model to distinguish between positive and negative classes.

In cases of highly imbalanced datasets, it is common to use a combination of metrics, such as precision, recall, and AUC-ROC to evaluate the performance of the model.

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

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'.

F1 score is a machine learning evaluation metric. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset

F-score = (2 * Precision * Recall) / (Precision + Recall)
Precision is the number of true positive predictions divided by the number of true positive plus false positive predictions. It is a measure of the accuracy of the classifier when it predicts the positive class.

Recall is the number of true positive predictions divided by the number of true positive plus false negative predictions. It is a measure of the classifier's ability to find all the positive instances.

The F-score combines both precision and recall into a single metric and it ranges between 0 and 1. The higher the value of F-score, the better is the performance of the classifier. The F-score is commonly used when there is an imbalance in the classes, and precision and recall have different importance.

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

fit() is used to fit the model to the data, and learn the parameters of the model.

transform() is used to apply the transformation learned during the fit() step to new data.

fit_transform() is a convenience method that combines fit() and transform() steps into one, and is used to fit the model to the data and then apply the transformation to the same data.

In short, fit() is used to learn the parameters of the model, transform() is used to apply the learned transformation to new data, and fit_transform() combines both steps into one.

fit()	transform()	fit_transform()
The fit method is used to compute the mean and std dev for a given feature to be used further for scaling	The transform method is used to perform scaling using mean and std dev calculated using the .fit() method	The fit_transform() method does both fits and transform.
In scikit-learn transformers,	The transform() method of sklearn transformers, will	scikit-learn transformers during fit_transform() method is

MACHINE LEARNING

the fit() method is used to fit the transformer to the input data and perform the required computations to the specific transformer we apply	transform the input data into some transformed spaced. The output is usually an array or a sparse matrix with equal number of samples (n_samples) as the input data. The transformation will be performed based on the parameters that were computed during fit	responsible for fitting the transformer and then return the transformed **training** instances
We use **fit()** method only on the training data	We use **transform()** method on train data as well as test data as we need to perform transformation in both cases.	We use fit_**transform()** method on train data as well as test data
While working with model (Models like the Linear Regression model, Decision Tree model, Random Forest model etc.) **fit()** method is used to calculate parameters/weights on the training data	**transform()** method is not used in the model	fit_transform() method is not used in the model