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 : D

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 : A

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 : 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

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 : To calculate the Gini index and entropy of the given dataset, we need to know the proportion of each class in the dataset. Let's assume that the dataset has 100 examples.

- Percentage of class A = 40%, so the number of examples in class A = 0.4 * 100 = 40
- Percentage of class B = 60%, so the number of examples in class B = 0.6 * 100 = 60

Now, we can calculate the Gini index and entropy as follows:

Gini index:

- Gini(A) = 1 - (p(A)^2 + p(B)^2) = 1 - (0.4^2 + 0.6^2) = 0.48
- Gini(B) = 1 - (p(A)^2 + p(B)^2) = 1 - (0.6^2 + 0.4^2) = 0.48
- Gini(Dataset) = p(A) * Gini(A) + p(B) * Gini(B) = 0.4 * 0.48 + 0.6 * 0.48 = 0.48

Entropy:

- Entropy(A) = -p(A) * log2(p(A)) - p(B) * log2(p(B)) = -0.4 * log2(0.4) - 0.6 * log2(0.6) = 0.971
- Entropy(B) = -p(A) * log2(p(A)) - p(B) * log2(p(B)) = -0.6 * log2(0.6) - 0.4 * log2(0.4) = 0.971
- Entropy(Dataset) = p(A) * Entropy(A) + p(B) * Entropy(B) = 0.4 * 0.971 + 0.6 * 0.971 = 0.971

Therefore, the Gini index and entropy of the given dataset are 0.48 and 0.971, respectively.

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

Ans : Random Forests have several advantages over Decision Trees:

1. Reduced overfitting: Random Forests reduce overfitting by creating multiple decision trees on different samples of the data and combining their predictions. This ensemble method helps to avoid the overfitting that can occur when a single decision tree is too complex.
2. Better accuracy: Random Forests can provide better accuracy than a single Decision Tree, especially for large datasets with high dimensionality.
3. Robustness to noise and outliers: Random Forests are less sensitive to noise and outliers in the data, as they aggregate the results of many individual trees.
4. Feature importance: Random Forests can provide a measure of feature importance, which can help in feature selection and understanding the underlying patterns in the data.
5. Efficient handling of missing data: Random Forests can handle missing data effectively without the need for imputation.

Overall, Random Forests are a powerful and flexible machine learning algorithm that can provide better accuracy and reduce overfitting compared to a single Decision Tree, making them a popular choice for many classification and regression tasks.

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

Ans : Scaling all numerical features in a dataset is important because some machine learning algorithms are sensitive to the scale of the input features. Features with larger scales can dominate the learning

process and lead to biased results. Additionally, scaling can help to normalize the input features and make them more comparable, which can improve the performance of some algorithms.

Two commonly used techniques for scaling numerical features are:

1. Min-Max scaling (also called normalization): This technique scales the features so that they have a minimum value of 0 and a maximum value of 1. This is done by subtracting the minimum value of each feature from all values and then dividing by the range (i.e., the difference between the maximum and minimum values).
2. Standardization: This technique scales the features so that they have a mean of 0 and a standard deviation of 1. This is done by subtracting the mean of each feature from all values and then dividing by the standard deviation.

Scaling techniques can be easily applied using libraries like Scikit-learn in Python. It is important to note that scaling should only be applied to numerical features, and not to categorical or ordinal features.

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

Ans : Scaling can provide several advantages in optimization using the gradient descent algorithm:

1. Faster convergence: Scaling can help the gradient descent algorithm converge faster to the minimum value of the cost function, as it ensures that the optimization steps are of similar size and in the same direction.
2. Avoiding oscillations: Scaling can help to avoid oscillations and overshooting the minimum value of the cost function. This is because the optimization steps will be more stable and less sensitive to the curvature of the cost function.
3. Better conditioning: Scaling can improve the condition of the optimization problem by ensuring that the gradient values are not too large or too small. This can help to reduce the risk of numerical instabilities and make the optimization more robust.
4. Improved numerical stability: Scaling can also improve the numerical stability of the gradient descent algorithm by reducing the risk of underflow or overflow errors.

Overall, scaling is an important technique for improving the performance and stability of optimization using the gradient descent algorithm, especially for high-dimensional and ill-conditioned problems.

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 : In the case of a highly imbalanced dataset, accuracy is not a good metric to measure the performance of the model. This is because accuracy is biased towards the majority class and does not take into account the class imbalance.

For example, suppose we have a dataset with 99% of the samples belonging to the negative class and only 1% belonging to the positive class. If we build a model that always predicts the negative class, it will have an accuracy of 99%. However, this model will be completely useless in predicting the positive class, which is the class of interest.

To address this issue, other evaluation metrics can be used that take into account the class imbalance. Some commonly used metrics for imbalanced datasets are:

1. Precision: This measures the proportion of true positives among the total predicted positives. It is a useful metric when the cost of false positives is high.
2. Recall (sensitivity): This measures the proportion of true positives among the total actual positives. It is a useful metric when the cost of false negatives is high.
3. F1-score: This is a harmonic mean of precision and recall and provides a balanced measure of the model's performance on both classes.
4. Area under the ROC curve (AUC-ROC): This measures the ability of the model to discriminate between the two classes across different probability thresholds.

In summary, accuracy is not a good metric for evaluating the performance of models on imbalanced datasets, and alternative metrics such as precision, recall, F1-score, and AUC-ROC should be used instead.

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

Ans : The f-score is a metric used in binary classification problems that combines precision and recall into a single measure of the model's performance. It is the harmonic mean of precision and recall, and provides a balanced measure of the model's performance on both positive and negative classes.

The formula for the f-score is:

f-score = 2 * (precision * recall) / (precision + recall)

where precision is the proportion of true positives among the total predicted positives, and recall is the proportion of true positives among the total actual positives.

The f-score ranges between 0 and 1, where a score of 1 indicates perfect precision and recall, and a score of 0 indicates the worst possible performance. The f-score is commonly used in imbalanced datasets, where the class distribution is skewed, as it provides a balanced measure of the model's performance on both classes.

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

Ans : In machine learning, fit(), transform(), and fit_transform() are three common methods used for preprocessing data.

fit() is a method used to learn the parameters of the transformation (such as mean and standard deviation for standardization) from the training data. When we call the fit() method on a training dataset, the transformation parameters are calculated and stored as an internal state of the transformer object.

transform() is a method used to apply the learned transformation to a dataset. Once the transformation parameters have been learned from the training data using the fit() method, we can apply the same transformation to new datasets using the transform() method. The transform() method does not modify the internal state of the transformer object.

fit_transform() is a convenience method that combines the fit() and transform() methods into a single step. It is used to learn the transformation parameters from the training data and apply the same transformation to the same dataset in a single step. This can be more efficient than calling fit() and transform() separately.

To summarize, fit() is used to learn the transformation parameters, transform() is used to apply the transformation to a dataset, and fit_transform() is a convenience method that combines fit() and transform() into a single step.