FLIP ROBO

ASSIGNMENT – 7

MACHINE LEARNING

1. Which of the following in sk-learn library is used for hyper parameter tuning?
D) All of the above

2. In which of the below ensemble techniques trees are trained in parallel?
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?
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.

5. Which of the following is true regarding Random Forests?
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.

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

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?
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. To calculate the Gini index and entropy of the dataset, we need to first calculate the probability of each class.

Probability of class A = 0.4

Probability of class B = 0.6

Gini Index:

Gini index measures the impurity of a dataset. It ranges from 0 to 1, where 0 means all the samples belong to one class and 1 means the samples are evenly distributed across all classes.

Gini index = 1 - (P(A)^2 + P(B)^2)

Gini index = 1 - (0.4^2 + 0.6^2)

Gini index = 1 - (0.16 + 0.36)

Gini index = 1 - 0.52

Gini index = 0.48

Entropy:

Entropy measures the impurity of a dataset similar to the Gini index. It ranges from 0 to 1, where 0 means all the samples belong to one class and 1 means the samples are evenly distributed across all classes.

Entropy = -P(A)log2(P(A)) - P(B)log2(P(B))

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

Entropy = -(0.4)(-1.32) - (0.6)(-0.74)

Entropy = 0.528 + 0.444

Entropy = 0.972

Therefore, the Gini index of the dataset is 0.48 and the entropy of the dataset is 0.972.

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

Ans. Random Forests have several advantages over a single decision tree. Some of the advantages are:

Reduces overfitting: Decision trees tend to overfit the data, which means they fit the training data too well and fail to generalize to new data. Random Forests reduce overfitting by aggregating multiple decision trees and taking the average prediction.

Improves accuracy: Random Forests can achieve higher accuracy compared to a single decision tree. This is because the component trees are trained on different subsets of the data, reducing the chance of overfitting and increasing the diversity of the models.

Handles missing data: Random Forests can handle missing data and maintain accuracy. This is because when a tree is constructed using a subset of data, the missing data is handled differently in each tree, and the final prediction is made by taking the average of all the tree predictions.

Robust to outliers: Random Forests are less sensitive to outliers compared to a single decision tree. This is because each tree is constructed using a random subset of the data, and outliers have less influence on the final prediction.

Easy to use: Random Forests are easy to use and require minimal data preprocessing. They can handle categorical and numerical data without the need for feature scaling or one-hot encoding.

Overall, Random Forests are a powerful and popular machine learning algorithm that can handle a variety of data types and achieve high accuracy.

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

Ans. Scaling numerical features is important because machine learning algorithms such as k-nearest neighbors, support vector machines, and gradient descent-based algorithms are sensitive to the scale of the input features. If the features are not scaled, the algorithms may give undue importance to features with a larger scale, leading to poor performance.

Two commonly used techniques for scaling numerical features are:

Standardization: Standardization scales the features to have a mean of 0 and a standard deviation of 1. This is done by subtracting the mean from each feature and dividing by the standard deviation.
Min-Max scaling: Min-Max scaling scales the features to a range between 0 and 1. This is done by subtracting the minimum value from each feature and dividing by the range (i.e., the difference between the maximum and minimum values).
Scaling ensures that the features are on a similar scale and have a similar impact on the model. This can improve the performance of the model and make it more robust to outliers.

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

Ans. The gradient descent algorithm is an optimization technique used to minimize a cost function by iteratively adjusting the parameters of a model or system. Scaling the input features can provide several advantages in optimization using gradient descent:

Improved convergence: Scaling can speed up convergence of the algorithm. If the input features are on different scales, then the algorithm may take a longer time to converge. Scaling can help the algorithm to converge faster and with fewer iterations.

Better conditioning: Scaling can also help in improving the conditioning of the optimization problem. The condition number of a problem indicates how sensitive the solution is to changes in the input. Scaling can reduce the condition number and make the problem more well-conditioned.

Avoiding overflow or underflow: Scaling can also help to avoid numerical overflow or underflow. If the input features have very large or very small values, then the intermediate computations can lead to overflow or underflow. Scaling can help to keep the values within a reasonable range and avoid these issues.

Better visualization: Scaling can also help in visualizing the data. If the input features are on different scales, then it can be difficult to visualize the data in a meaningful way. Scaling can help to normalize the input features and make the visualization more interpretable.

More accurate optimization: Scaling can also lead to more accurate optimization results. If the input features are on different scales, then the optimization may not be able to find the global minimum. Scaling can help to make the optimization more accurate and find the global minimum.

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 case of a highly imbalanced dataset for a classification problem, accuracy is not a good metric to measure the performance of the model. This is because accuracy is biased towards the majority class and may not provide a good measure of how well the model is performing for the minority class(es).

Consider an example where we have a dataset with 95% of the samples belonging to the majority class and only 5% of the samples belonging to the minority class. If we build a classifier that always predicts the majority class, it would have an accuracy of 95%. However, this classifier is not useful for predicting the minority class, which may be the class of interest.

In such cases, metrics like precision, recall, and F1-score are more appropriate measures of the model's performance. Precision measures the proportion of true positives among the predicted positives, recall measures the proportion of true positives among the actual positives, and F1-score is the harmonic mean of precision and recall. These metrics take into account both the true positives and false positives and provide a better measure of the model's performance for the minority class(es).

Additionally, other metrics like ROC-AUC and PR-AUC can also be used to evaluate the performance of a classifier on an imbalanced dataset. These metrics focus on the trade-off between true positive rate and false positive rate, and are useful for comparing different models or tuning the threshold for binary classification.

14. What is “f-score" metric? Write its mathematical formula.
Ans. The F-score, also known as the F1 score, is a metric used to evaluate the performance of a binary classification model. It combines the precision and recall of the model into a single score. Precision measures the proportion of true positives out of all predicted positives, while recall measures the proportion of true positives out of all actual positives.
The mathematical formula for F-score is:
F-score = 2 * (precision * recall) / (precision + recall)
where precision = true positives / (true positives + false positives)
and recall = true positives / (true positives + false negatives)
The F-score ranges from 0 to 1, with 1 being the best possible score. A high F-score indicates that the model has high precision and high recall, which means that it is able to correctly identify both positive and negative cases.

15. What is the difference between fit(), transform() and fit_transform()?
Ans. In machine learning, the terms fit(), transform(), and fit_transform() are commonly used in the context of data preprocessing and model training. Here's a brief explanation of the differences between them:

fit(): This method is used to fit the model to the training data. During this process, the model learns the patterns and relationships in the data and generates a set of parameters that define the model. For example, in a linear regression model, fit() calculates the slope and intercept of the line that best fits the data.

transform(): Once the model has been trained using fit(), the transform() method is used to apply the model to new data. This method takes the new data as input and applies the same transformations that were applied to the training data during the fit() step. The resulting output is the transformed data, which can then be used for prediction or further analysis.

fit_transform(): This method combines the fit() and transform() steps into a single operation. It is commonly used for data preprocessing, where the same transformations need to be applied to both the training and testing data. In this case, fit_transform() fits the model to the training data and applies the same transformations to both the training and testing data.

In summary, fit() is used to train the model, transform() is used to apply the model to new data, and fit_transform() is used to train the model and apply the same transformations to both the training and testing data.