Worksheet 7
Machine Learning Assignment Solution

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. D) All of the above.

GridSearchCV() and RandomizedCV() are both methods for hyperparameter tuning in scikit-learn library. They are used to systematically search for the best combination of hyperparameters that optimize the performance of a given model.

K-fold cross-validation is a technique for evaluating the performance of a model and can also be used in combination with hyperparameter tuning methods like GridSearchCV() and 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.

In Random forest, trees are trained in parallel, which means that each tree is trained independently of the others. This makes the training process faster and more efficient.

In Adaboost and Gradient Boosting, trees are trained sequentially. The output of one tree is used as input for the next tree. The training process continues until a predefined stopping criterion is met.

Therefore, the correct answer is 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. A) The regularization will decrease.

In the line of code **`sklearn.svm.SVC (C=1.0, kernel='rbf', degree=3)`**, **C** is the hyperparameter for the SVM (Support Vector Machine) model that controls the regularization. The regularization parameter **C** trades off correct classification of training examples against maximizing the decision function's margin.

If **C** is increased, the regularization will decrease, and the model will become more flexible by allowing more misclassifications of the training examples. On the other hand, decreasing **C** will increase the regularization and make the model more restrictive by penalizing misclassifications more strongly.

Therefore, increasing the **C** hyperparameter will reduce the effect of regularization in the SVM model.

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.

In the line of code **`sklearn.tree.DecisionTreeClassifier(criterion='gini', splitter='best', max_depth=None, min_samples_split=2)`**, **`max_depth`** is a hyperparameter for the decision tree classifier model that determines the maximum depth up to which the tree can grow.

Setting a **`max_depth`** value will regularize the decision tree by limiting the depth of the tree. This prevents the tree from overfitting the training data by learning spurious relationships between the features and the target variable. By limiting the depth of the tree, the model becomes simpler and more interpretable.

Option B is incorrect because **`max_depth`** does not denote the number of children a node can have. The number of children a node can have is determined by the **`splitter`** parameter, which can take the values **`'best'`** or **`'random'`**.

Therefore, the correct answer is 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.

Random Forest is an ensemble learning technique that combines multiple decision trees to improve the accuracy and robustness of the predictions.

Option A is partially correct. Random Forest is an ensemble method, but the component trees are typically decision trees, which are not necessarily weak learners.

Option B is incorrect. The component trees in a random forest are trained independently in parallel, not in series.

Option C is correct. In a classification problem, each tree in the random forest predicts the class label of a given input, and the final prediction is made by taking the mode of the class labels predicted by all the trees.

Therefore, the correct answer is 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.

In Gradient Descent algorithm, the learning rate controls the step size that the algorithm takes in each iteration to reach the optimal solution. If the learning rate is too high, the algorithm may overshoot the optimal solution, and the cost function may start increasing instead of decreasing. This can cause the algorithm to diverge from the optimal solution and fail to converge.

Option B is incorrect because keeping the learning rate very high may lead to fast convergence, but the optimal solution might be missed due to oscillation around it.

Therefore, the correct answer is 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. B) Bias will decrease, Variance increase.

As the model complexity increases, the model becomes more flexible and is able to fit the training data more closely. This typically reduces the bias of the model, i.e., the difference between the expected predictions of the model and the true values.

However, as the model becomes more complex, it may also fit the noise in the training data and start overfitting. This means that the model may become too sensitive to the noise and variations in the training data, and may not generalize well to unseen data. This typically increases the variance of the model, i.e., the variability of the predictions of the model for different training datasets.

Therefore, as the model complexity increases, bias typically decreases, but variance increases.

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.

In machine learning, we want our model to generalize well to unseen data. A model that is overfitting the training data means that it is capturing the noise and idiosyncrasies of the training data and is not able to generalize well to new data. This is typically indicated by high training accuracy and low test accuracy.

In this case, the model has a high training accuracy of 0.95, which suggests that it is fitting the training data very closely. However, the test accuracy is much lower at 0.75, which suggests that the model is not able to generalize well to new data. This is a typical sign of overfitting.

Therefore, the correct answer is B) Model is overfitting.

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

9. Suppose we have a dataset which has 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:

The formula to calculate the Gini index is:

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

Substituting the values, we get:

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

Therefore, the Gini Index of the dataset is 0.48.

Entropy:

The formula to calculate entropy is:

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

Substituting the values, we get:

Entropy = -0.4 * log2(0.4) - 0.6 * log2(0.6) = -0.4 * (-1.32) - 0.6 * (-0.97) = 0.528 + 0.582 = 1.11

Therefore, the entropy of the dataset is 1.11.

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

Ans. Advantages are:

- Random forests consist of multiple single trees each based on a random sample of the training data. They are typically more accurate than single decision trees. The following figure shows the decision boundary becomes more accurate and stable as more trees are added.
- **Trees are unpruned**. While a single decision tree like CART is often pruned, a random forest tree is fully grown and unpruned, and so, naturally, the feature space is split into more and smaller regions.

- **Trees are diverse**. Each random forest tree is learned on a random sample, and at each node, a random set of features are considered for splitting. Both mechanisms create diversity among the trees.
- Two random trees each with one split are illustrated below. For each tree, two regions can be assigned with different labels. By combining the two trees, there are four regions that can be labeled differently.
- **Handling Overfitting**

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 it ensures that each feature contributes equally to the analysis and modeling process.

When features are on different scales, some features may dominate over others and this can lead to biased results. For example, in distance-based models, such as k-Nearest Neighbors (k-NN), features with larger values will dominate over features with smaller values, regardless of their importance.

Scaling the features helps to eliminate this bias and ensure that all features are equally important in the analysis and modeling process. Additionally, some algorithms, such as gradient descent based optimization algorithms, converge faster when features are on the same scale.

Two techniques used for scaling are:

1. **Standardization (Z-score normalization):** It scales the features such that they have zero mean and unit variance. This is done by subtracting the mean from each feature and dividing by its standard deviation. This results in features with a mean of 0 and standard deviation of 1.
2. **MinMax Scaling:** It scales the features to a specified range, typically [0,1]. This is done by subtracting the minimum value of the feature and dividing by the range (i.e., the difference between the maximum and minimum values). This results in features with values between 0 and 1.

MACHINE LEARNING ASSIGNMENT - 7

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

Ans. Scaling provides several advantages in optimization using gradient descent algorithm:

3. **Faster Convergence:** When features are on different scales, the gradient for each feature will also be on a different scale. This can result in the algorithm taking longer to converge to the minimum, or in some cases, it may not converge at all. By scaling the features, we can ensure that the gradient for each feature is on the same scale, which can help the algorithm converge faster.
4. **Preventing Overflow or Underflow:** When features are on different scales, the weights associated with each feature will also be on different scales. If the weights associated with some features become too large, it can lead to overflow, and if they become too small, it can lead to underflow. By scaling the features, we can prevent this from happening.
5. **Better Conditioning:** Conditioning refers to the sensitivity of the optimization problem to changes in the parameters (i.e., the weights). When features are on different scales, the optimization problem may be ill-conditioned, which can result in the algorithm taking longer to converge or getting stuck in a local minimum. By scaling the features, we can improve the conditioning of the optimization problem, which can help the algorithm converge faster and find a better solution.
6. **Improving Interpretability:** Scaling the features can also make the weights associated with each feature more interpretable. When features are on different scales, the weights associated with each feature will also be on different scales, making it difficult to compare their importance. By scaling the features, we can ensure that the weights associated with each feature are on the same scale, which can make it easier to interpret their importance.

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. The reason for this is that accuracy does not take into account the imbalance in the dataset and can be misleading.

Consider an example where we have a binary classification problem with 99% of the data belonging to class A and only 1% of the data belonging to class B. If we build a model that simply predicts class A for every instance, the accuracy of this model would be 99%, which appears to be a good performance. However, this model is completely useless as it does not predict any instances of class B, which may be the class of interest.

Therefore, in such cases, it is important to use evaluation metrics that take into account the imbalance in the dataset. Some of the commonly used evaluation metrics for imbalanced datasets are:

7. **Precision:** It is the ratio of true positives to the total number of positive predictions. It gives an idea of how many of the positive predictions made by the model were actually correct.
8. **Recall/Sensitivity:** It is the ratio of true positives to the total number of actual positives. It gives an idea of how many of the positive instances in the dataset were correctly predicted by the model.
9. **F1-score:** It is the harmonic mean of precision and recall. It gives a single metric that balances both precision and recall.
10. **ROC-AUC:** It measures the area under the Receiver Operating Characteristic (ROC) curve, which is a plot of sensitivity (true positive rate) against 1-specificity (false positive rate) for different threshold values. It gives an idea of how well the model is able to distinguish between the positive and negative classes.

Overall, the choice of evaluation metric should be based on the specific requirements of the problem and the relative importance of different types of errors.

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

Ans. The F-score (also called F1-score) is a metric that combines precision and recall into a single measure of model performance. It is commonly used in binary classification problems where we are interested in both the number of true positives and the number of false negatives.

The mathematical formula for the F-score is:

F1-score = 2 * (precision * recall) / (precision + recall)

where precision is the ratio of true positives to the total number of positive predictions, and recall is the ratio of true positives to the total number of actual positives.

The F1-score ranges from 0 to 1, where a score of 1 indicates perfect precision and recall, and a score of 0 indicates that either precision or recall is 0. It is a harmonic mean of precision and recall, which means that it gives equal weight to both measures. The F-score is a useful metric for situations where we want to balance precision and recall, and where we are interested in finding a trade-off between these two measures.

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

Ans. In machine learning, **`fit()`**, **`transform()`**, and **`fit_transform()`** are common methods used in data preprocessing and model training. Here are the differences between them:

- **`fit()`**: This method is used to calculate the parameters required for scaling or transforming the data. When we call **`fit()`** on a dataset, the scaler or transformer learns the mean, variance, or any other required parameter from the data. For example, when we use **`StandardScaler`** to standardize our data, we call the **`fit()`** method on the training data, and it calculates the mean and standard deviation of each feature.
- **`transform()`**: This method is used to apply the transformation to the data. Once we have calculated the parameters using **`fit()`**, we can use **`transform()`** to apply the same transformation to the data. For example, if we have calculated the mean and standard deviation of each feature using **`fit()`**, we can use **`transform()`** to standardize our data.
- **`fit_transform()`**: This method combines the **`fit()`** and **`transform()`** methods into a single step. It is used to both calculate the parameters required for scaling or transforming the data and to apply the transformation to the data. This can be convenient when we want to perform both operations in a single step.

In summary, **`fit()`** is used to calculate the parameters required for scaling or transforming the data, **`transform()`** is used to apply the transformation to the data, and **`fit_transform()`** combines both steps into a single operation.