NAME: Dipesh Ramesh Limaje

INTERNSHIP BATCH : 33

TOPIC: MACHINE LEARNING

SME : Mr. Shwetank Mishra

WORKSHEET NO : 7

ANS 1] D) All of the above

ANS 2] A) Random forest

ANS 3] B) The regularization will decrease

ANS 4] A) It regularizes the decision tree by limiting the maximum depth up to which a tree can be grown.

ANS 5] A) It's an ensemble of weak learners.

ANS 6] A) Gradient Descent algorithm can diverge from the optimal solution.

ANS 7] B) Bias will decrease, Variance increase

ANS 8] B) model is overfitting

Q9 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 know the probability of each class occurring in the dataset. From the given information, we know that:

Percentage of class A = 40%

Percentage of class B = 60%

Therefore, the probability of class A occurring in the dataset is 0.4 and the probability of class B occurring in the dataset is 0.6.

To calculate the Gini index, we can use the following formula:

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

To calculate the entropy, we can use the following formula:

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 Gini index of the dataset is 0.48 and the entropy of the dataset is 1.11.

Q10 What are the advantages of Random Forests over Decision Tree?

ANS

Random Forests are an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting. Here are some advantages of Random Forests over Decision Trees:

- Better accuracy: Random Forests can achieve higher accuracy than a single decision tree by combining the predictions of multiple decision trees. The aggregation of the trees helps to reduce the variance of the model and increase the generalization performance.
- Reduced overfitting: A single decision tree is prone to overfitting because it tries to fit the training data perfectly. Random Forests overcome this problem by creating many decision trees with different subsets of the data, thus reducing the effect of individual trees that might overfit the data.
- Robust to outliers: Decision Trees are sensitive to outliers, but Random Forests are robust to them because they aggregate the predictions of multiple trees.
- Feature importance: Random Forests can provide information about the importance of the features used in the model, which can help in feature selection and data analysis.
- Handling missing values: Random Forests can handle missing values in the data by estimating the missing values from the available data. This is not possible with Decision Trees.
- Parallel processing: Random Forests can be easily parallelized, which means that the training time can be reduced significantly by distributing the computation across multiple cores or machines.

Random Forests offer several advantages over Decision Trees, including higher accuracy, reduced overfitting, robustness to outliers, feature importance, handling of missing values, and parallel processing.

Q11What is the need of scaling all numerical features in a dataset? Name any two techniques used for scaling.

ANS

Scaling is the process of transforming the numerical features in a dataset to have a similar scale, which is important for many machine learning algorithms. Here are some reasons why scaling is needed:

Avoiding bias: Many machine learning algorithms are sensitive to the scale of the features, and if the features are not scaled properly, some features may dominate others in the model, leading to bias.
Improving convergence: Some optimization algorithms, such as gradient descent, converge faster when the features are on the same scale.

Two common techniques used for scaling numerical features are:

Standardization: This technique scales the features to have zero mean and unit variance. Each feature is transformed by subtracting its mean and dividing by its standard deviation. Standardization works well for features that have a roughly normal distribution.
Min-Max scaling: This technique scales the features to a fixed range, typically between 0 and 1. Each feature is transformed by subtracting its minimum value and dividing by the range of values. Min-Max scaling works well for features that have a bounded range and do not have a normal distribution.

scaling is important for many machine learning algorithms to avoid bias and improve convergence, and two common techniques for scaling numerical features are standardization and min-max scaling.

Q12 Write down some advantages which scaling provides in optimization using gradient descent algorithm.

ANS

Scaling the features in a dataset can provide several advantages in optimization using gradient descent algorithm:

- Faster convergence: Gradient descent algorithm converges faster when the features are on a similar scale. If the features are on different scales, some features may have a larger influence on the update of the weights than others, which can slow down the convergence of the algorithm. Scaling the features to a similar scale can help to reduce this problem and speed up the convergence of the algorithm.
- Avoiding oscillations: Gradient descent algorithm can oscillate or overshoot the optimal solution if the features are not scaled properly. Scaling the features can help to reduce the oscillations and ensure that the algorithm converges to the optimal solution.
- Avoiding getting stuck in local minima: Gradient descent algorithm can get stuck in local minima if the features are not scaled properly. Scaling the features can help to avoid getting stuck in local minima and ensure that the algorithm finds the global minimum.
- Better visualization: Scaling the features can make it easier to visualize the data and the decision boundary of the model. If the features are on different scales, it can be difficult to interpret the coefficients and the decision boundary of the model.

scaling the features in a dataset can provide several advantages in optimization using gradient descent algorithm, including faster convergence, avoiding oscillations, avoiding getting stuck in local minima, and better visualization.

Q13 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 may not be a good metric to measure the performance of the model. The reason is that accuracy is calculated by dividing the number of correct predictions by the total number of predictions. In an imbalanced dataset, the number of examples in the majority class is much larger than the number of examples in the minority class, and a model that always predicts the majority class will have a high accuracy even if it performs poorly on the minority class.

For example, let's say we have a dataset with 99% of the examples in the majority class and 1% of the examples in the minority class. A model that always predicts the majority class will have an accuracy of 99%, but it will fail to detect the minority class, which is usually the class of interest.

Therefore, in an imbalanced dataset, it is important to use metrics that are sensitive to the performance of the model on the minority class, such as precision, recall, F1 score, and area under the ROC curve. These metrics take into account both true positives and false positives/negatives, and are not affected by the class imbalance in the same way as accuracy

In summary, accuracy may not be a good metric to measure the performance of the model in a highly imbalanced dataset, and metrics that are sensitive to the performance on the minority class should be used instead.

Q14 What is “f-score" metric? Write its mathematical formula.

ANS

F-score is a metric that combines precision and recall to provide a single measure of performance for binary classification problems. It is also known as F1 score, as it is the harmonic mean of precision and recall.

The formula for F-score is:

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

where precision is the number of true positives divided by the number of true positives plus false positives, and recall is the number of true positives divided by the number of true positives plus false negatives.

F-score ranges between 0 and 1, where a value of 1 indicates perfect precision and recall, and a value of 0 indicates that the model predicts all examples as negative or positive.

F-score is a useful metric when both precision and recall are important, and it is often used in problems where the classes are imbalanced, and there is a need to balance the trade-off between precision and recall.

Q15 What is the difference between fit(), transform() and fit_transform()?

ANS

In machine learning, fit, transform, and fit_transform are methods that are commonly used for preprocessing the data and fitting machine learning models.

fit(): The fit method is used to fit a preprocessing step or a machine learning model to the training data. In other words, it learns the parameters of the model or the transformation based on the training data. For example, in the case of a scaler, the fit method learns the mean and standard deviation of the training data.

transform(): The transform method is used to apply the transformation or model to the data. Once the parameters are learned by the fit method, the transform method is used to apply the transformation to the training or test data. For example, in the case of a scaler, the transform method scales the data based on the mean and standard deviation learned by the fit method.

fit_transform(): The fit_transform method is used to combine the fit and transform methods in a single step. It learns the parameters of the transformation on the training data and applies the transformation to the same data. In other words, it is equivalent to calling the fit method followed by the transform method.