CODE BASICS

MACHINE LEARNING TAKEAWAYS

Chapter:

Ensemble Learning

What is Ensemble Learning?

1. Ensemble learning is a strategy in machine learning where we join predictions from several models. This approach helps us get a more precise and resilient forecast.

2. It can be achieved using any of the techniques below,

   Basic Techniques
   - Majority Vote
   - Average
   - Weighted Average

   Advanced Techniques:
   - Bagging
   - Boosting
   - Stacking

Majority Voting, Average and Weighted Average

1. The voting classifier typically employs the "hard" method, which is based on majority rule voting. However, for classifiers that are well-calibrated, the "soft" method is suggested. This method determines class labels according to the argmax of the predicted probabilities.

2. The Voting Regressor method will be used for regression, just as the Voting Classifier demonstrated for classification in the video.

Bagging

1 Bagging in ensemble learning is a technique where multiple models are trained on random subsets of the training data and their predictions are aggregated to improve accuracy and reduce variance.

2 Each model in ensemble learning is called a Base learner.

3 Bagging is also known as bootstrap aggregating.

4 Benefit of bagging:

- Robust against Outliers
- Reduction in Variance
- A good way to handle high dimensionality.
- Improved Accuracy.

Bagging: Random Forest

1. Random forest stands as one of the two key algorithms in Machine Learning.

2. A random forest is a learning method that constructs multiple decision trees on random data subsets and features during the training phase, delivering the class that represents the mode of the classes or the average prediction of the individual trees.

3. The use of random forest provides a unique perspective on the data, enhancing the robustness of your model.

Random Forest: Raisin Classification

1. Random Forest is versatile, suitable for both classification and regression tasks.

2. All models in the Random Forest share the same decision tree criterion.

3. Depending on the specific use case or scenario, parameters and arguments can be adjusted for optimal model performance.

Boosting: AdaBoost

1. Boosting in machine learning is a sequential process where multiple models, typically weak learners, are trained one after the other, with each model building upon the errors of its predecessor to improve overall accuracy and form a strong predictive model.

2. Different Boosting Techniques:
    - AdaBoost
    - Gradient Boost
    - XGBoost (eXtreme Gradient Boosting)
    - LightGBM
    - CatBoost

3. A Decision Stump is a tree that has one root node and two child nodes

Boosting: AdaBoost

1. AdaBoost is a machine learning algorithm that adjusts training instances' weights based on prior models' errors. It emphasizes wrongly predicted instances, increases their weights, and enhances the ensemble's overall accuracy.

2. Formulas Used:

- Total Error - It is a sum of weight of all the incorrect predictions
- Amount of Say $(\alpha) = \frac{1}{2} \log \left(\frac{1 - error}{error} \right)$
- Correct Prediction -> New weight = old weight * $e^{(-\alpha)}$
- Incorrect Prediction -> New weight = old weight * e^{α}

Gradient Boosting: Regression Walk Through

1 Gradient boosting is an ensemble machine learning technique that iteratively trains decision trees on residual errors, combining them with a learning rate to progressively improve model accuracy.

2 The residuals can be referred to as pseudo-residuals.

$$F_K(x) = F_0(x) + \text{Learning Rate} \cdot \text{PR1} + \text{Learning Rate} \cdot \text{PR2} + \ldots + \text{Learning Rate} \cdot \text{PRK}$$

Gradient Boosting: Regression Math

Step 1: Initialize the model with a constant value

$$F_0(X) = \underset{\gamma}{\text{argmic}} \sum_{i=1}^{n} L(y_1, \gamma)$$

Step 2: for m=1 to M:

(A) Compute rim = $-\left[\frac{\partial L(yi,F(\chi i))}{\partial F(\chi_i)}\right]_{F(\chi) = F_{m-1}}(\chi)$ For i = 1, ..., n

(B)Train a base learner (or weak learner, e.g., tree) using the training set $\{(\chi_i, r_{im})\}$And create terminal regions R_{jm}, for j= 1 to j_m

(C)Compute $\gamma_{jm} = \underset{\gamma}{\text{argmic}} \sum_{xi\varepsilon Rij} L(y_1,F_{m-1}(\chi_1) + \gamma)$

(D)Update Fm (χ) = Fm-1 $(\chi) + \nu * \sum_{j=1} \gamma jmI(\chi \varepsilon R_{jm})$

Strep 3: Output F_m (χ)

Gradient Boosting: Classification

1. Gradient Boosting applies a similar method for both classification and regression, with the only difference being the loss function – Logistic Loss for classification and Mean Squared Error for regression.

2. When contrasted with a decision tree or random forest, gradient boosting has a slight edge in terms of performance and overall prediction accuracy.

XGBoost: Walk Through

1. XGBoost is the Powerful and widely used statistical machine learning algorithm in industry.

2. XGBoost is a high-performance machine learning library that constructs decision trees using exact and approximate algorithms for split finding. It uses post-growth tree pruning to enhance model accuracy and scalability, particularly for large datasets, while minimizing overfitting.

XGBoost: Benefits

1. High Accuracy
   a. Better at modeling complex non-linear relationships due to tree-based learning and boosting approach.
   b. In-built regularization
   c. Automatic Handling of missing values
   d. Feature importance and selection

2. Very Fast
   a. Tree Pruning
   b. Block structure and parallel processing
   c. Cache Awareness
   d. Efficient Handling of Sparse Data