Self-Training with XGBoost: Regression or Classification task

Introduction

XGBoost is a powerful machine learning library that excels in various types of predictive tasks, including classification and regression. This document provides a guide for self-training using XGBoost with two classic datasets: the Iris dataset for classification and the Boston Housing dataset for regression. These datasets offer a practical introduction to XGBoost's capabilities and how to utilize it effectively.

Select one of the options for implementation in a timeframe of up to 90 minutes.

Datasets Overview

1. Iris Dataset (Classification)

- **Description:** The Iris dataset is a well-known dataset used for demonstrating classification tasks. It consists of 150 samples from three species of Iris flowers (setosa, versicolor, and virginica), with four features measured for each sample: sepal length, sepal width, petal length, and petal width.
- **Target:** The target variable is the species of the Iris flower, which is a categorical variable with three possible values.
- **Use Case:** Classification of Iris flowers into one of the three species based on the given features.

Code for Loading the Iris Dataset:

```
from sklearn.datasets import load_iris
import pandas as pd

# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Create a DataFrame for easy manipulation and visualization
df = pd.DataFrame(X, columns=iris.feature_names)
df['target'] = y

print(df.head())
```

2. Boston Housing Dataset (Regression)

- **Description:** The Boston Housing dataset contains information on housing in the Boston area. It includes 506 instances with 13 features, such as the number of rooms, age of the property, and distance to employment centers, which are used to predict the median value of owner-occupied homes.
- **Target:** The target variable is the median home value, a continuous numerical variable.
- **Use Case:** Predicting the median house prices based on the provided features.

Code for Loading the Boston Housing Dataset:

```python
from sklearn.datasets import load_boston
import pandas as pd

# Load the Boston Housing dataset
boston = load_boston()
X = boston.data
y = boston.target

# Create a DataFrame for easy manipulation and visualization
df = pd.DataFrame(X, columns=boston.feature_names)
df['target'] = y

print(df.head())
```

Setting Up Your Environment

Before diving into the code, it’s important to set up a virtual environment to manage your dependencies. This helps to create an isolated environment for your project, ensuring that the specific versions of the packages you use do not conflict with others on your system.

Using Virtual Environments

1. **Create a Virtual Environment:**
   - Open your terminal (Command Prompt, PowerShell, or Terminal on Mac/Linux).
   - Navigate to your project directory.
   - Create a new virtual environment:
     - `python -m venv xgboost_env`
   - Activate the virtual environment:
     - `xgboost_env\Scripts\activate`
2. **Install Required Packages:**
   - Once the virtual environment is activated, install the necessary packages using pip:
     - `pip install xgboost scikit-learn pandas matplotlib seaborn`

Basic Instructions

- **Activate Virtual Environment:** Each time you start working on your project, activate the virtual environment using the command appropriate for your operating system.
- **Deactivate Virtual Environment:** When you are done, you can deactivate the virtual environment by simply typing:
- `deactivate`

Summary

This guide introduces you to self-training using XGBoost with two datasets:

- **Iris Dataset** for classification, where you predict the species of Iris flowers.
- **Boston Housing Dataset** for regression, where you predict the median value of houses based on various features.

Each dataset is loaded using Scikit-Learn, and we provide the necessary setup instructions to prepare your environment and install required packages. This setup ensures a smooth start to experimenting with XGBoost and exploring its powerful features for both classification and regression tasks.