README FOR REFACTORED CODE

Overview

This project contains a set of Python classes designed for handling, preprocessing, training, evaluating, and predicting data using machine learning models. The project utilizes various libraries such as pandas, numpy, scikit-learn, seaborn, matplotlib, and joblib for these tasks. The classes are structured to provide a streamlined workflow for data analysis and machine learning.

Requirements

Ensure you have the following Python libraries installed:

- pandas

- numpy

- matplotlib

- seaborn

- scikit-learn

- joblib

You can install the required libraries using pip:

```bash

pip install pandas numpy matplotlib seaborn scikit-learn joblib

```

Classes and Functions:

1. `DataHandler`

Purpose: Handles loading of historical and latest data from CSV files.

Methods:

- **__init__(self, historical_data_path, latest_data_path=None):** Initializes with paths to historical and latest data files.
- **load_data(self):** Loads data from the provided CSV files, handling errors gracefully.

2. `DataPreprocessor`

Purpose: Preprocesses the dataset by scaling its features.

Methods:

- **__init__(self, data):** Initializes with the dataset to be preprocessed.
- **preprocess(self):** Scales the features using `StandardScaler`.
- **save_scaler(self, filepath='scaler.pkl'):** Saves the scaler to a file.

3. `ModelTrainer'

Purpose: Trains a machine learning model using the provided data.

Methods:

- **__init__(self, model, X, y):** Initializes with the model and dataset, splitting the data into training and testing sets.
- **train(self):** Trains the model using the training data.
- **save_model(self, filepath='model.pkl'):** Saves the trained model to a file.

4. `ModelEvaluator`

Purpose: Evaluates the performance of a trained model using test data.

Methods:

__init__(self, model, X_test, y_test): Initializes with the model and test data.

evaluate(self): Evaluates the model, returning accuracy, confusion matrix, and classification report.

plot_confusion_matrix(self, confusion): Plots the confusion matrix.

5. `Predictor`

Purpose: Makes predictions using a trained model and plots the results.

Methods:

- **__init__(self, model, scaler, latest_data):** Initializes with the model, scaler, and latest data.
- **make_predictions(self):** Makes predictions on the latest data.
- **plot_predictions(self, predictions):** Plots the predictions on the latest data.