451 Programming Assignment 1: Research Report

Author: Chenyi Zhao
GitHub Repository

Problem Description

This project addresses the challenge of predicting the next day's return in a financial time series. The dataset includes daily historical prices of WTI crude oil. The main objective is to determine whether the return on the next trading day will be positive or not, using only historical price information. To approach this task, I used feature engineering techniques to extract useful patterns from the raw data, selected a set of predictors, and trained a classification model using a cross-validation strategy that respects the temporal structure of the data. The key goal was to explore how well historical trends can be used to predict future returns, even if the signal is weak.

Data Preparation and Pipeline

The raw dataset used in this assignment consisted of daily WTI oil prices, with columns such as Date, Open, High, Low, and Close. I first sorted the data in chronological order to ensure all future steps respected the time series nature. Two columns that appeared in earlier versions of the dataset documentation, Dividends and StockSplits, were not present in the CSV file I used. Therefore, no column-dropping was necessary at this stage.

I proceeded to create new features. Lag variables were constructed from the Close price, shifting values back by 1 to 5 days to form CloseLag1 through CloseLag5. These lagged features are intended to capture short-term trends. Additionally, I computed daily returns and rolling averages to better capture price momentum and smoothing effects. A binary target variable called Target was created to indicate whether the next day's return was positive (1) or not (0).

The final dataset, therefore, consisted of engineered features derived from past prices, along with a target column. I used the polars library for efficient data manipulation, then converted the feature matrix and target series to pandas format for compatibility with scikit-learn and XGBoost.

Research Design

This is a binary classification task. The feature matrix (X) included lagged closing prices and return-based features, while the target (y) was the binary indicator of next-day price movement. Rather than randomly splitting the dataset, I employed a time-series cross-validation strategy using the TimeSeriesSplit class from scikit-learn, which ensures that training always precedes validation temporally. This method avoids data leakage and mimics how models would be deployed in production.

For the model, I used the XGBoost classifier. XGBoost is a gradient-boosted tree algorithm known for its accuracy and ability to handle both linear and nonlinear relationships in data. It also includes built-in mechanisms to handle missing values and provides feature importance scores.

The cross-validation involved five temporal splits. Each fold trained the model on earlier data and validated it on subsequent periods. This allowed me to estimate how well the model might generalize to unseen future data.

Results and Analysis

The model achieved an average AUC (Area Under the ROC Curve) score of approximately 0.5145 across the five cross-validation folds. This value is only marginally better than 0.5, which represents random guessing.

While the result may seem underwhelming at first, it is consistent with expectations. Predicting short-term movements in financial markets is notoriously difficult due to noise, external shocks, and the semi-random walk nature of price series. The modest performance suggests that the features derived from price alone contain limited predictive power, at least in the simple forms used here.

That said, the pipeline worked correctly from data loading through feature extraction and model evaluation. The structure allows for iterative refinement, where more sophisticated features or external data sources could be introduced to improve performance.

Conclusion and Future Directions

In this assignment, I built a complete machine learning pipeline to predict next-day price movements in WTI oil markets. Although the predictive performance was close to random, the process helped solidify my understanding of time series feature engineering, model validation in temporal settings, and using XGBoost for classification.

To potentially improve model accuracy, future steps could include engineering more advanced features, such as technical indicators (RSI, MACD, Bollinger Bands), or incorporating external macroeconomic variables, such as inventory reports, interest rates, or geopolitical news sentiment. More complex models such as recurrent neural networks (RNNs) or transformer-based models may also be appropriate for capturing long-term dependencies in sequential data.

AI Assistance Acknowledgement

Throughout this assignment, I used AI assistants such as ChatGPT to help clarify programming errors, understand how to use libraries like polars and xgboost, and improve code documentation. The AI assistant was particularly helpful in explaining errors, suggesting alternatives, and reviewing markdown formatting for this report. All final decisions, code writing, and analyses were performed manually.