MSDS 451 Financial Engineering Assignment 1

James Chun Kit Kwok

October 2, 2025

Problem Description

This study applies supervised machine learning to predict whether the next day's return for Apple Inc. (AAPL) stock will be positive or negative. The goal is to evaluate whether engineered price-based technical indicators can help improve the discrimination between up and down return days. Predicting short-term market direction is a challenging task because stock prices are noisy and often weakly autocorrelated, yet even a modest edge can provide value for trading and portfolio management strategies.

Financial Feature Engineering

We use machine learning classifiers, specifically gradient boosted decision trees (XGBoost), to predict the direction of Apple Inc. (AAPL) daily returns using engineered price-based features. The study uses daily price data beginning October 2, 2015 and includes lagged closing prices, exponential moving averages (EMA), volatility and volume-based signals, and price difference features such as high minus low (HML) and open minus close (OMC). Our goal is to evaluate whether these technical features can improve prediction of next-day price movement. Feature engineering is essential in financial modeling because price series are noisy and often weakly predictive; selecting and transforming meaningful inputs can improve model performance.

Feature Engineering

The initial dataset includes open, high, low, close, adjusted close, and trading volume for AAPL downloaded from Yahoo Finance. From these raw data, we constructed a set of lagged price and trend indicators. CloseLag features represent one to ten day lags of the daily close to capture autocorrelation and momentum. EMA features for 5, 10, and 20 day exponential moving averages smooth recent price trends. HMLLag features (high minus low) and OMCLag features (open minus close) describe intraday volatility and candlestick behavior. Volume lags are included to capture delayed effects of trading activity. Rows with missing values caused by lagging were dropped. All predictors were standardized using Scikit-Learn's `StandardScaler` to stabilize model training. The target variable was defined

as binary: 1 for a positive next day return and 0 for a negative return. The series was nearly balanced with approximately 53% up days and 47% down days, so no oversampling or class weighting was required.

Modeling and Research Design

We selected **XGBoost**, a high-performance gradient boosting algorithm, as the predictive model. XGBoost builds an ensemble of decision trees sequentially, where each new tree corrects the residual errors of the previous ones. It is well-suited for structured financial data with potentially non-linear feature interactions and varying levels of predictive strength across features. A randomized hyperparameter search with **1000 iterations** was used to optimize model parameters for maximum classification accuracy. The parameter space included tree depth (`max_depth` from 2 to 15), minimum child weight (1 to 20), subsample ratio (0.3 to 1.0), learning rate (0.001 to 0.1), and number of boosting rounds (100 to 3000). We used time series cross-validation with five splits to respect the chronological order of the data and applied a purge gap of 10 days to reduce look-ahead bias between training and testing windows.

Best Model and Parameters

The randomized search produced several strong models and progressively improved cross-validated accuracy. Key tuning results included:

```
New best ACC: 0.4829 {max_depth=8, min_child_weight=15,
subsample=1.0, learning_rate=0.0609, n_estimators=1738}

New best ACC: 0.4930 {max_depth=11, min_child_weight=19,
subsample=0.4000, learning_rate=0.0469, n_estimators=230}

New best ACC: 0.4940 {max_depth=11, min_child_weight=12,
subsample=1.0, learning_rate=0.0905, n_estimators=2417}

New best ACC: 0.4945 {max_depth=14, min_child_weight=8,
subsample=0.3728, learning_rate=0.0832, n_estimators=2781}

New best ACC: 0.4988 {max_depth=12, min_child_weight=19,
subsample=1.0, learning_rate=0.0905, n_estimators=2897}

New best ACC: 0.5027 {max_depth=13, min_child_weight=2,
subsample=0.3244, learning_rate=0.0880, n_estimators=2561}
```

The **best mean accuracy across folds was 0.5027**, indicating the model performed slightly better than random guessing on unseen data. The optimal hyperparameters were:

```
max_depth=13, min_child_weight=2, subsample=0.3244,
learning_rate=0.0880, n_estimators=2561
```

Final Model Evaluation

After selecting the best parameters, the model was refitted on the full dataset and evaluated. Performance on the hold-out set was:

- Accuracy: 97.55%
- AUC: 0.9973

	precision	recall	f1-score	support
0	0.969	0.979	0.974	1164
1	0.981	0.973	0.977	1330
accuracy			0.976	2494
macro avg	0.975	0.976	0.975	2494
weighted avg	0.976	0.976	0.976	2494

Feature Importance

Feature importance was evaluated using both XGBoost gain importance and permutation importance. Gain-based importance identified `Open`, `CloseLag3`, `EMA10`, `CloseLag10`, and `CloseLag5` as the most influential predictors. Permutation importance indicated that recent volume and candlestick-related features contributed some predictive power but many lagged indicators added little incremental value.

Conclusion and Recommendations

The XGBoost model achieved only slightly better than random accuracy during time series cross-validation, showing that next-day AAPL returns are difficult to predict using purely price-based technical indicators. However, when retrained on the full dataset, the model reached very high in-sample accuracy (97.6%) and AUC (0.9973). The large gap between cross-validation and full-sample results suggests overfitting and highlights the importance of careful out-of-sample testing. Future work should include walk-forward validation with a true unseen test period and the inclusion of additional predictors such as macroeconomic signals, options market sentiment, or news-based features. These may provide incremental predictive power beyond technical indicators alone.