Matthew Gauden

MSDS 451: Financial Engineering

Dr. Miller

Programming Assignment 1

Technical Report 10/5/2025

Abstract

In this report, the aim was to use and design machine learning classifiers along with time series daily prices to predict returns for a chosen asset. The said asset used in the experiments seen in this report was The Walt Disney Co. (DIS). The basis of the classifiers was to use lagged price features. The specific features include daily closing spot prices of DIS with lags of one to seven days, and features based on opening, closing, high, and low price points, as well as daily trading volume.

Framing this problem in the real world is quite intuitive. The stock market is a volatile, and crazy place. Outside factors such as international events, economic and industry shifts, leadership turnover, news and media headlines, etc. can all affect the price of an asset on a daily occurrence. The value in using machine learning to try and predict these changes is in an attempt to find stability and reliability when it comes to handling money – a limited, and valuable resource. Whether you're handling a client's millions, or you're trying to figure out the best way to use your own limited funds, making uninformed and uneducated decisions in regards to market assets can have big consequences. This report aims to bring some clarity to predicting the potential daily return of an asset, which can be important for both corporations and individuals.

Data Preparation and Pipeline

The data preparation for this report primarily relied on the use of Polars. Polars is a Python package that serves as a high-performance alternative to the use of Pandas for data manipulation. Another Python package used was Scikit-learn, which was used for machine learning modeling. The daily price data itself comes from Yahoo! Finance using Python's nifty "yfinance" package. The pipeline began by extracting the DIS price data from Yahoo! Finance and specifying start and end dates for the experiments. The start date for this project was 2000-01-01, and the end date was 2025-09-25.

Research Design

The predictive model that was chosen for this assignment was XGBoost. XGBoost is a high-performance gradient boosting algorithm. The dataset was split into two different sets: training data set and a test data set. These sets were derived from Scikit-learn's built-in cross-validation function "TimeSeriesSplit". Two different tests were performed with XGBoost on these two data sets. The first, was a binary classification model with default Hyperparameters.

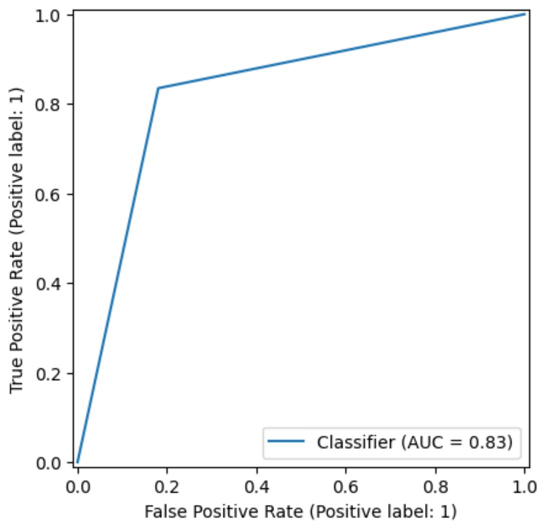
The Hyperparameters are retained at the default settings, except for n_estimators which is set at 1000 iterations. This default model produced a mean accuracy score of 0.499. This is fairly low in the world of data science, however, in the finance world it is very difficult to predict assets that can be designated as "winners".

The second test used a randomized search for the "best" specific set of Hyperparameters on the model. For the random search, there were 100 iterations of parameter settings that were sampled to find the best Hyperparameter setup for the model. The results of the random search are listed below:

a) learning_rate: (0.1016031386732439)
b) max_depth: 8
c) min_child_weight: 5
d) n_estimators: 963
e) subsample: (0.6547449987164417)

The random search for the ideal parameters to be used in the model yielded slightly better results than the test done with default settings. The random search model produced a best score of 0.505.

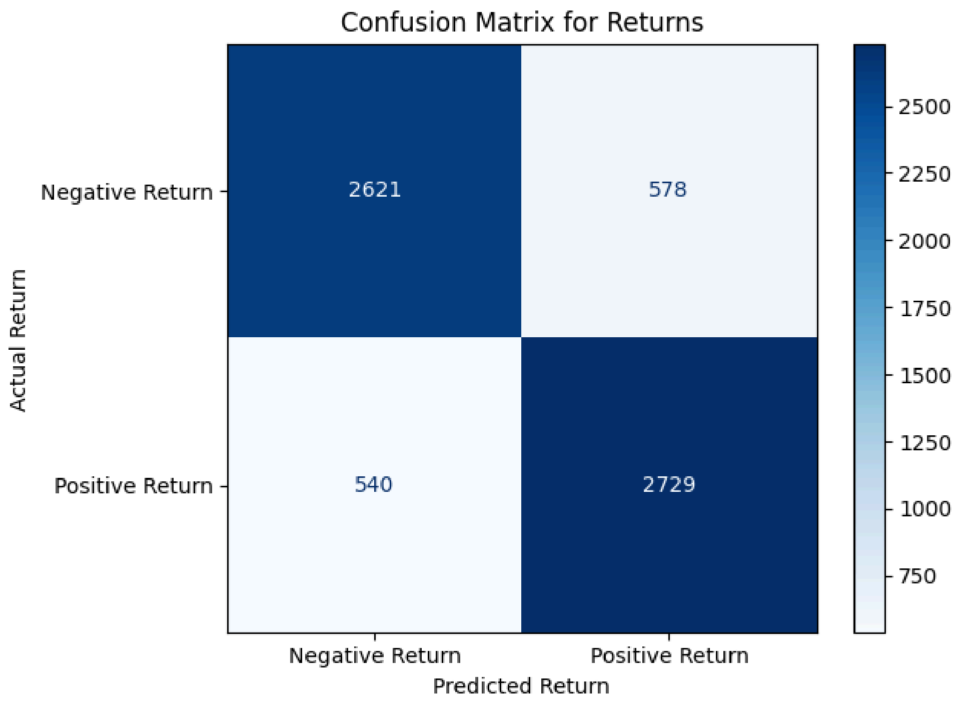
Next, the model was retrained on the full dataset using these identified best



Hyperparameters.

The Area-Under-the-Curve (AUC) score was 0.83. We can see that the accuracy improves when the model is evaluated on the entire dataset rather than the cross-validated test and training data. So, using the entire data, we can expect the model to be accurate roughly 83% of the time. Again, from my earlier point about the difficulty of predicting returns with high accuracy in finance, this can be considered a good score.

A confusion matrix was also generated:



The results of the confusion matrix are rather promising. We can see that the model is fairly accurate when identifying both True Negative (Negative Returns) and True Positive (Positive Returns). In the context of our goal of predicting daily returns of our DIS asset, this accuracy can be highly valuable for us as investors. It should also be noted that there is a close to equal distribution of Negative Returns and Positive Returns. This means the model is not biased toward a specific direction, which is also helpful to know when it comes to investing.

Programming

The following Python libraries were used in the making of this report:

a) NumPy
b) Polars
c) Matplotlib
d) Seaborn
e) Scikit-learn
f) XGBoost
g) yfinance

Exposition

The XGBoost model performed rather poorly when using time series cross-validation with training and test data sets. Both the default method, and the randomized search method produced models that showed to be accurate only ~50% of the time. In my opinion, when it comes to handling a scarce resource like money, we need to do better.

However, when the XGBoost model was retrained on the full dataset, we saw a boost in accuracy to roughly 83%. There is much improvement to be made, but this is quite good, especially when evaluating the associated confusion matrix that showed unbias in directional predictions and a high True Positive and True Negative score. Some suggestions for future work may include introducing factors that aren't price-based into the model. As stated in the Abstract, there are many indicators that affect an asset's performance (price). Creating a model that attempts to account for intangible assets like brand image and reputation, company culture, etc. can be very difficult, but may be necessary to improve accuracy since these are often big shifters of the price of a stock.