MSDS 451: Financial Machine Learning

Feature Engineering

Programming Assignment 1

Blade Hunter Robelly
7-13-2025

Financial Feature Engineering

To model the short-term behavior of NVIDIA Corporation (NVDA) stock, I engineered a compact yet informative set of features using historical market data. The original dataset contained daily Open, High, Low, Close, and Volume data extracted from NVDA's historical records. From the table below, we can identify the features we will use in our model by taking the features from the least AIC listed.

trialNumber	features	aic
i64	str	f64
3963	2 5 6 10 14	8865.398181
88	6 10	8865.660937
1383	2 6 10 14	8865.879891
500	6 8 10	8865.950994
1382	2 6 10 13	8866.259587
8802	2 5 6 8 10 14	8866.599521
3879	2 4 6 10 14	8866.648741
3962	2 5 6 10 13	8866.756605
101	8 10	8867.000303
494	6 7 10	8867.022814

Based on the table, the following five features were selected for their predictive utility:

- *CloseLag3*
  - Lag-three daily closing prices, capturing trend and momentum effects from three trading days ago.
- *HMLLag3*
  - Lag-three high-minus-low price spreads, reflecting daily volatility or price range from three days prior.
- *OMCLag1*
  - Lag-one open minus closing daily prices, capturing short-term intraday reversal patterns from the previous trading day.
- *VolumeLag2*
  - Lag-two daily trading volume values, representing recent volume pressure and potential institutional activity.
- *CloseEMA8*
  - Eight-day exponential moving average of closing prices, providing a smoothed trend indicator over the past week and a half.

Financial Machine Learning

The predictive model implemented in this analysis is an XGBoost classifier. The model was configured to optimize binary classification, predicting the direction of NVDA's next-day return (up = 1, down = 0) using engineered lag and trend-based features.

Model Configuration

The following were the key parameters set in the model:

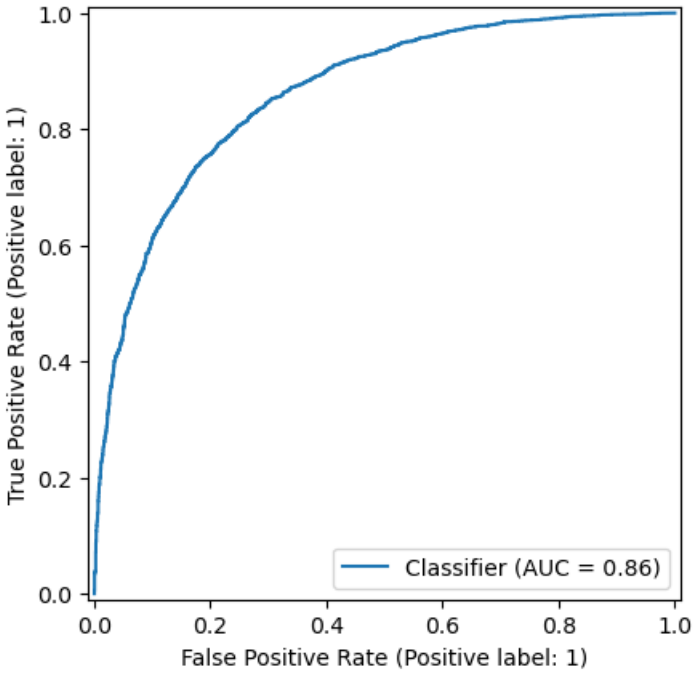
- max_depth = 9
- min_child_weight = 9
- subsample = 0.50
- learning_rate = 0.09
- n_estimators = 273

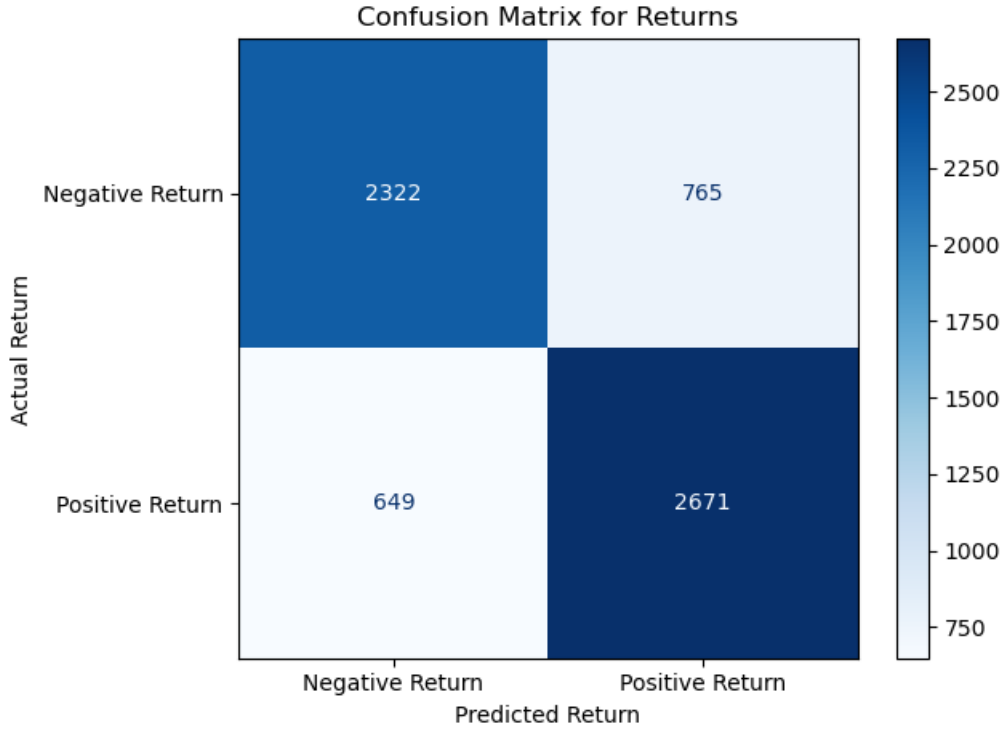
Model Performance

The first two lines of the classification report are reproduced here as well as the accuracy of the model.

	precision	recall	f1-score	support
0	0.78	0.75	0.77	3087
1	0.78	0.80	0.79	3320
micro avg	0.78	0.78	0.78	6407
macro avg	0.78	0.78	0.78	6407
weighted avg	0.78	0.78	0.78	6407

Below, you will find the ROC curve with its AUC, and the confusion matrix for returns.





Looking at these results, the model achieved an overall classification accuracy of ~78%, correctly predicting 4993 out of 6407 instances. Among the 3087 actual negative cases, 2322 were correctly identified as negative, while 765 were incorrectly classified as positive. For the 3320 actual positive cases, 2671 were correctly predicted as positive, with 649

misclassified as negative. The balanced F1-scores, which range between 0.77 and 0.79 for both classes, indicate consistent precision and recall, reflecting robust and reliable performance in identifying both upward and downward price movements. Furthermore, the model's AUC was 0.88, backing its strong ability to discern between classes across varying classification thresholds. Taken together, these results confirm that the model performs effectively in predicting price movement direction.

Conclusions and Next Steps

Although the model shows good performance on the training data, it is important to remember that results on training data tend to be better than what we can expect when the model sees new data. To get a better idea of how well the model will work in practice, it should be tested on a separate hold-out test set.

In the future, it would be useful to expand the model from just predicting whether prices go up or down to a three-category prediction. This would include predicting if prices go up (suggesting buying or going long), stay stable (no action), or go down (suggesting selling or going short). This approach would be more practical for real trading decisions and could help guide automated trading strategies.

Additionally, this project provides a starting point to add more features that might improve predictions. These could include economic indicators, news events, reports, or information from other related markets. Adding these types of data might help the model better understand what drives price changes.

Predicting stock prices is a difficult problem that requires not only knowledge of machine learning but also experience with financial data. Building successful models takes time, practice, and a good understanding of how financial markets work.

References