Preprocessing and Feature Engineering Report

Predicting Next-Day S&P 500 Direction Using Price and Volume Data

Group A

Team Lead: Ross Schanck

Recorder: Auriana Anderson

Spokesperson: Chase Golden

November 24, 2025

Background and Research Question

Our project looks at whether short-term patterns in the S&P 500, mainly recent price changes, volume, and simple technical indicators can help predict if the market will move up or down the next day.

Research Question (Revised):

Can we use historical S&P 500 price and volume data to predict whether the next day's closing price will move up or down?

Hypothesis and prediction:

We expect that short-term market behaviors like recent returns, volatility, and moving averages contain enough repeated structure for machine learning models to slightly outperform random guessing. Out prediction is that simple models like Logistic Regression and Random Forest should achieve around 52-60% accuracy, which would show that these engineered features capture meaningful but subtle signals in the market.

Methods

Overview of Our Plan

Last week, our plan was to finalize preprocessing, confirm how to handle the missing rows created by rolling features, expand our feature engineering, and determine whether unsupervised methods like PCA are suitable. After reviewing our EDA findings, the plan stayed mostly the same, but we added additional features like lagged returns, rolling return averages, and a Sharpe-like metric. We also found the PCA and KMeans clustering were appropriate for our dataset and implemented both.

Preprocessing Methods & Justification

We downloaded S&P 500 data using the yfinance API for the years 2010 – 2025. The dataset included Open, High, Low, Close, Adjusted Close, and Volume. Since yfinance outputs clean and well-structured data we did not need to fix column types or formatting issues. We reset the index so the date was a normal column and sorted the data in chronological order, which is required for time series modeling.

Handling Missing Data

- The only missing values in the dataset were generated by our feature-engineering steps:
- Pct_Change (missing first row)
- Rolling means (MA_3, MA_5, MA_10)

- Rolling volatility (Volatility, rolling_vol_10d)
- Lagged returns (return_lag_1d to return_lag_7d)
- Rolling return mean (ma_return_10d)
- Sharpe-like ratio (sharpe_like_10d)

All missing values were expected because rolling windows and lags require previous observations. These missing rows were removed only after all engineered features were created. This approach avoids artificially imputing values that would distort real market behavior.

Outlier Decisions

We did not remove outliers because extreme movements are a real part of the stock market's dynamics. Removing them would eliminate meaningful market reactions and reduce realism of the dataset.

Train Test Split

We used an 80/20 chronological split, which is required for time-series data to avoid lookahead bias.

- Training: 2010-2021 (3019 rows)
- Testing: 2021-2024 (755 rows)

This ensures the model is only evaluated on future unseen data.

Feature Engineering Methods & Justifications

We engineered several features that capture short-term momentum, volatility, trend behavior, and risk adjusted performance. These are common in financial modeling.

1. Percent Returns and Lagged Features

We calculated daily percent returns (return_1d) and created lag features at 1, 2, 3, 5, and 7 days. These lags help the model identify:

- Immediate reactions (1-2 days)
- Mid-term behavior (3-5 days)
- Short weekly cycles (7 days)

These are widely used in market prediction research.

2. Rolling Price Indicators

We generated moving averages (MA_3, MA_5, MA_10) to smooth noise and show short-term trends.

3. Volatility Measures

We included two volatility-based features:

- Volatility = rolling SD of closing prices
- rolling_vol_10d = rolling SD of returns

These quantify market turbulence and risk, which can influence price direction.

4. Trend Based Return Indicator

ma_return_10d gives the average return over the last 10 days, capturing momentum.

5. Sharpe Like Ratio

We created sharpe_liek_10d = return/ volatility. This measures risk adjusted returns. High values may signal strong upward momentum.

6. Target Variables

We defined the next day's direction using:

- Up - if next day's close is greater than today's close
- Down - otherwise

Later, we converted to binary form (Up = 1, Down = 0) for modeling

Results and Interpretations

Cleaning and Transformation

All NA values were exclusively caused by rolling and lag features. Dropping these rows produced a complete dataset with no remaining missing values. Outliers were kept intentionally due to financial relevance. The binary target encoding was simple and interpretable.

Processing Outcomes

The chronological 80/20 split was verified by checking the minimum and maximum dates in both sets. There was no overlap, confirming no leakage. Because our data is sequential, a random split would have compromised the model integrity.

Unsupervised Feature Engineering

Once Even though our original proposal questioned whether PCA was needed, our notebook results showed it was appropriate.

Principal Component Analysis (PCA)

- Standardized all numeric features

- Retained 95% of variance, resulting in 9 components
- PCA helped reduce redundancy caused by highly correlated price-derived features
- Cumulative variance plot supported the component selection

KMeans Clustering

We tested cluster counts from k=2 to k=7 and selected k=3 based on inertia reduction. Cluster labels were then assigned to both training and testing data. These clusters may later be included as modeling features.

Supervised Feature Engineering Plan

We used Logistic Regression and Random Forest as baseline models. Their results will guide our next feature selection steps.

The next steps planned include:

- Using Logistic Regression coefficients to identify strong predictors
- Using Random Forest feature importance
- Running recursive feature elimination (RFE)
- Testing interaction terms

Discussion and Next Steps

Key Takeaways

Our dataset is now fully cleaned, processed, and prepared for modeling. All missing values were structurally expected from rolling metrics and lags and were resolved appropriately.

The final feature set captures momentum, volatility, trends, and risk adjusted behavior. This directly aligns with our research question about short-term market direction.

PCA and KMeans proved more useful than originally expected. Logistic Regression slightly outperformed Random Forest, but both models provide insight into where to focus future feature selections.

Modeling Plan

Next steps include:

- Training baseline and advanced models - Logistic Regression, Random Forest, Gradient Boosting
- Evaluating accuracy, precision, recall, and F1-score on the test set
- Using model outputs for supervised feature selection
- Experimenting with adding cluster labels as a feature
- Considering class balancing if needed
- Possibly expanding indicators and retesting PCA

GitHub Repository

https://github.com/aurianaanderson/StockMarketTrends/tree/main