Forecasting the S&P 500 Using Macro-Financial Variables

Solal Danan

January 27, 2025

1 Introduction

Financial markets are inherently complex, influenced by a multitude of factors. The ability to systematically forecast market trends provides a significant advantage in risk management, asset allocation, and trading strategies. In particular, Machine Learning (ML) for alpha generation has become an essential tool in quantitative finance, aiming to uncover predictive signals in financial data. One of the primary challenges in financial time series forecasting lies in the low signal-to-noise ratio inherent in market data. Financial markets operate as complex, dynamic systems, where interactions between macroeconomic indicators, investor sentiment, and external shocks contribute to non-stationary behaviors and high levels of noise. As a result, developing a robust predictive model requires careful feature selection, regularization techniques, and model validation strategies to avoid overfitting and improve generalization.

Motivation

This report presents the results of a machine learning project proposed by AI for Alpha, where the objective was to forecast the one-week ahead log returns of the S&P 500 using a dataset of historical macro-financial variables spanning over 20 years. Given the high dimensionality and potential multicollinearity of the features, feature selection and model interpretability played a crucial role in our approach. To evaluate our models, we used a custom scoring metric that balances Root Mean Squared Error (RMSE), which captures prediction accuracy, and Directional Accuracy (DA), which measures the model's ability to correctly predict the market's direction (up or down). This combined metric allows us to assess both the magnitude of prediction errors and the practical usefulness of the model in trading applications.

Since the problem formulation was open-ended, we first analyze the target variable to justify our choice of predictive formulation. Next, we outline our methodology, detailing the data preprocessing, model selection, and validation process. We then present our experimental results and discuss the impact of key design choices on model performance. Finally, we conclude with insights and potential directions for future work, including improvements to feature selection, model robustness, and alternative approaches for feature engineering.

2 Related Works

Predicting stock returns is a topic that has fascinated several scholars throughout history, and a wealth of approaches has been utilized to forecast or describe the future price path of equities as well as their corresponding returns. Some of the earlier methodologies harnessed the power of econometrics to pinpoint factors that could be identified as determinants of future stock performance. Such early attempts led to the financial field's most infamous models, including the CAPM, the Fama-French three-factor model, and the Carhart four factor model; notably, these models included factors such as the market factor, the value factor, the size factor, and the momentum factor to attempt to explain and forecast future returns. As this econometric research broadened and deepened by adding ever more complex and numerous factors, the technologic capacities as well as the analytics techniques expanded in parallel, and an array of novel machine learning approaches began to be applied in an endeavor to maximize predictive power.

3 Problem Formulation

Target Variable

Stock prices are inherently non-stationary, following a random walk that leads to spurious correlations with macroeconomic variables. Predicting prices directly is not ideal, as it mainly captures trend persistence rather than meaningful relationships with explanatory features. Additionally, strong autocorrelation in prices inflates model performance without providing useful predictive insights. To ensure stationarity and facilitate modeling, we focus on forecasting log returns rather than prices. Log returns exhibit more stable statistical properties, making them suitable for regression models. Compared to arithmetic returns, log returns reduce skewness, improving normality assumptions and ensuring robustness. Moreover, log returns are time-additive, allowing for better interpretability over different time horizons.

Choice of Forecasting Horizon

We predict the S&P 500 log returns at a weekly horizon rather than a daily one. Daily returns are highly volatile, making it difficult to capture meaningful trends. Weekly aggregation smooths short-term noise, improving predictive stability. Furthermore, macroeconomic variables, which often have a delayed impact on the market, align better with a weekly frequency. A longer horizon, such as monthly, was considered but deemed impractical due to the limited number of observations in the dataset.

Evaluation Metrics

To assess the performance of our model, we employ three key metrics: Root Mean Squared Error (RMSE), and Directional Accuracy (DA). Additionally, we define a custom scoring function that combines RMSE and DA as:

$$Score = (1 - \beta) \cdot \text{DA} - \beta \cdot \text{RMSE} \tag{1}$$

where β is a hyperparameter that controls the weight given to RMSE relative to DA. A higher β prioritizes reducing RMSE, whereas a lower β emphasizes directional accuracy.

Given these considerations, we build a model to forecast the weekly log returns of the S&P 500, leveraging macro-financial variables while addressing the challenges posed by financial time series forecasting.

4 Methodology

4.1 Data Preprocessing

To prevent data leakage and maintain the temporal structure of financial time series, we split the dataset chronologically. The training set includes past data up to 2020, while the test set contains more recent observations. We ensure that both sets include significant market downturns to allow the model to generalize across different market regimes. Our approach ensures that both the training and test sets include major financial crises (e.g., 2008, COVID-19) to improve model robustness. The dataset contains missing values due to rolling statistics and data availability constraints. Rather than discarding large portions of data, we adopt a multi-step imputation approach, implemented in a python class. The first rows of the dataset, where rolling features are incomplete, are removed to ensure meaningful feature availability. Columns with excessive missing values (above a predefined threshold) are eliminated. The remaining missing values are imputed using a combination of forward fill (for time-continuous features), backward fill (for stable macroeconomic indicators), and median imputation. This strategy ensures that we retain sufficient data while minimizing biases introduced by imputation. We also tried to mitigate the impact of extreme values by applying winsorization to cap outliers. This prevents extreme fluctuations in macroeconomic variables from disproportionately influencing the model. Since financial variables span different scales, we standardize all features using the `StandardScaler`. This transformation ensures that all features contribute equally to the learning process.
To streamline data processing, we implement a `Preprocessor` class that encapsulates all preprocessing steps, from missing value imputation to normalization. This modular design ensures consistency and reproducibility throughout the pipeline.

4.2 Feature Engineering and Transformation

Given the focus of this challenge on feature selection rather than feature creation, our primary objective was to refine existing variables for better model interpretability. Due to time constraints, we did not engineer new features but instead applied transformations to enhance data usability. Highly skewed variables can introduce biases in regression models, making them harder to interpret and less robust. To mitigate this, we applied log and square root transformations to selected features, reducing skewness and bringing their distributions closer to normality. This transformation ensures that extreme values do not disproportionately influence the model. Including interaction terms or financial indicators derived from raw variables, could further enhance model performance.

4.3 Feature Selection

Given the high dimensionality of the dataset, feature selection is crucial to reduce redundancy, improve model interpretability, and mitigate the risk of overfitting. We employed two complementary approaches: Principal Component Analysis (PCA) and a manual selection process based on permutation importance, correlation analysis, and statistical testing.

4.3.1 Dimensionality Reduction with PCA

Principal Component Analysis (PCA) is a linear transformation technique that projects high-dimensional data into a lower-dimensional space while preserving as much variance as possible. It achieves this by computing the eigenvectors and eigenvalues of the covariance matrix, generating principal components that are uncorrelated by construction.

One key assumption of PCA is that the principal components effectively capture the most meaningful structure in the data. However, PCA assumes linear relationships between features and does not explicitly optimize for predictive performance.

As illustrated in Figure 1, the first 60 principal components account for nearly all of the cumulative explained variance, demonstrating that much of the information in the original feature set is concentrated in a lower-dimensional space.

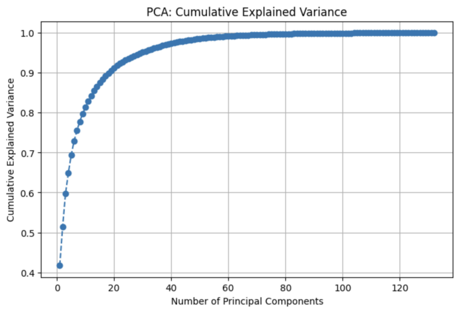

Figure 1: Cumulative explained variance by principal components

4.3.2 Manual Feature Selection: Permutation Importance and Correlation Analysis

The other approach to refine the feature set, we applied a manual selection process based on model-agnostic permutation importance and correlation filtering.

Permutation Importance Permutation importance measures the contribution of each feature by randomly shuffling its values and evaluating the impact on model performance. Features with low or negative importance scores degrade model performance when removed, while those with negative scores may introduce noise. We systematically eliminated features with negative permutation scores to retain only those that positively contributed to predictive power.

Correlation Filtering and Bonferroni Correction Highly correlated features can introduce multicollinearity, inflating variance and reducing interpretability. To address this, we computed pairwise Pearson correlation coefficients and applied a statistical significance test.

To control for multiple hypothesis testing (MHT) and mitigate false discoveries (data snooping), we applied the Bonferroni correction. The Bonferroni method adjusts the significance threshold by dividing the original alpha level by the number of tests performed:

$$\alpha_{\text{adjusted}} = \frac{\alpha}{m} \tag{2}$$

where m is the number of tested feature pairs. This conservative approach reduces the likelihood of false positives. Features with a correlation above a threshold and a statistically significant p-value (after correction) were removed, ensuring that only the most independent features were retained.

4.4 Model Selection

Our modeling approach follows a structured evaluation process, beginning with a baseline model, followed by the exploration of more sophisticated models. We ensured that all feature selection, feature engineering, and model selection steps were evaluated on the training set to prevent lookahead bias. To maintain robustness, we employed a time series split validation strategy, ensuring that past information is used to predict future returns, mimicking a realistic forecasting setup.

4.4.1 Baseline Model: ARIMA

As a first step, we implemented an ARIMA (AutoRegressive Integrated Moving Average) model without incorporating macro-financial variables. The goal was to establish a baseline for comparison and assess the intrinsic predictability of the S&P 500 weekly log returns based solely on past values.

The results from ARIMA confirmed that macro-financial variables carry predictive power. Indeed, all machine learning models incorporating macroeconomic features significantly outperformed the autoregressive model, justifying our focus on feature selection and model refinement.

4.4.2 Exploring Predictive Models

Given the high-dimensional nature of the dataset and the need for interpretability, we considered both linear and tree-based models:

- **Ridge Regression:** A regularized linear model that mitigates multicollinearity by shrinking less relevant features. This model provides a strong baseline for generalization and interpretability.
- **Extra Trees Regressor:** A tree-based ensemble method that demonstrated strong predictive power while retaining some level of interpretability through feature importance analysis.

4.4.3 Feature Selection and Model Tuning

Throughout the model selection process, we systematically evaluated feature engineering, feature selection using only the training set. This ensures that our model selection was not influenced by data snooping or lookahead bias, which could artificially inflate predictive performance. For validation, we employed a time series split strategy, where models were trained on past data and evaluated on progressively more recent data. This method replicates a realistic investment scenario, where future predictions must rely solely on historical information.

The final models were then selected based on their performance in terms of root mean squared error (RMSE) and directional accuracy (DA). In the next section, we detail our tuning process and the final model evaluations.

5 Results

To improve model performance, we conducted hyperparameter tuning using a randomized search approach. This was applied to both the Extra Trees Regressor and Ridge Regression, leveraging a time series split validation strategy to ensure robustness and prevent look-ahead bias.

5.1 Hyperparameter Tuning Strategy

Given the large number of features and potential interactions, we opted for Randomized Search Cross-Validation instead of an exhaustive grid search to efficiently explore a wide range of hyperparameter values. This approach allowed us to strike a balance between computational efficiency and effective hyperparameter optimization.

- **Extra Trees Regressor:** We optimized parameters such as the number of estimators, maximum depth, minimum samples per split, and feature selection strategy.
- **Ridge Regression:** The main hyperparameter, the regularization strength (α), was tuned to balance bias-variance trade-offs and improve generalization.

5.2 Model Performance Comparison

Below, we present a summary table of key models tested, highlighting their Root Mean Squared Error (RMSE), Directional Accuracy (DA), and the final custom score on the unseen test set, which includes the COVID-19 crisis and more recent data:

Model	**RMSE** $\downarrow$	**DA (%)** $\uparrow$
Baseline ARIMA	0.0207	52.57
Extra Trees (Untuned)	0.0193	63.54
Ridge Regression (Untuned)	0.0566	55.75
Tuned Extra Trees + FS	0.0248	64.00
Tuned Ridge Regression + FS	0.0255	65.78

Table 1: Comparison of Model Performance After Feature Selection and Hyperparameter Tuning

6 Discussion

- Baseline ARIMA underperforms: The autoregressive model struggles to capture meaningful patterns beyond past log returns. This confirms that macroeconomic features contain valuable predictive information.
- Extra Trees Regressor provides strong predictive power:** The untuned Extra Trees model already outperformed ARIMA significantly, indicating that a non-linear approach can effectively exploit macro-financial variables.
- Feature selection improves performance: Removing highly correlated features led to better generalization and interpretability. However, some feature selection techniques (such as PCA) did not substantially improve accuracy.

- Extra Trees vs. Ridge Regression:
  - Extra Trees performed well, but exhibited signs of slight overfitting, as its test performance slightly deteriorated compared to training/validation results.
  - Ridge Regression, while performing worse than Extra Trees, showed strong robustness and generalization capabilities on the test set. This highlights the benefit of simpler, regularized model in preventing overfitting.

These results confirm the added value of macroeconomic features for forecasting S&P 500 weekly log returns.

7 Conclusion

In this project, we explored the use of macro-financial variables to predict the weekly log returns of the S&P 500. By implementing robust preprocessing, feature selection, and model tuning techniques, we demonstrated that macroeconomic data significantly enhances predictive power compared to an autoregressive baseline. Extra Trees and Ridge Regression proved to be effective models, each with distinct strengths in handling non-linearity and generalization.

Future Directions. Moving forward, several improvements could be explored:

- **Enhanced Feature Engineering:** Developing new financial indicators and incorporating alternative data sources (e.g., sentiment analysis, alternative macro indicators) could refine predictive accuracy.
- **Adaptive Modeling:** Since model performance varied across different market regimes, particularly during crises, a regime-switching approach or meta-learning framework could be introduced to adapt model selection based on market conditions.
- **Feature Selection:** Reinforcement Learning (RL)-based feature selection could dynamically adapt to changing market conditions by identifying the most relevant features in different regimes. By formulating feature selection as a sequential decision-making problem, RL can optimize for long-term predictive stability rather than static selection criteria.

Overall, this work highlights the potential of macro-financial variables in forecasting market trends and serves as a foundation for more sophisticated predictive models in quantitative finance.