

Louvain School of Management

**The impact of prediction market
signals on stock returns: A
comparative exploratory study of
global and local modeling
techniques**

Author(s): Adrien Payen
Supervisor(s): Corentin Vande Kerckhove
Academic year 2024-2025
Dissertation for the master of
Business Analytics
Daytime schedule

Declaration of Generative AI Usage

In the preparation of this thesis, generative artificial intelligence tools were used in a complementary manner to assist with specific tasks such as text reformulation, grammar and spelling correction, and the structuring of technical sections. Specifically, the OpenAI-developed ChatGPT model (version GPT-4, accessed through the ChatGPT interface) was used for language refinement, methodological clarifications, and LaTeX formatting suggestions.

In addition, the Cursor AI development environment, which integrates generative AI capabilities, was used to edit, correct, and optimize the R code employed throughout the thesis. The use of these tools was limited to technical and syntactic support, in accordance with programming best practices.

No automatically generated content was included without critical review and personal adaptation. All outputs were carefully verified, validated, and incorporated into an original and academically rigorous work.

This declaration is made in the interest of transparency and academic integrity. In accordance with institutional guidelines, the use of such tools is explicitly acknowledged and does not compromise the originality or scientific quality of the thesis.

Abstract

This thesis investigates the relationship between prediction market signals and stock returns through a comparative modeling framework. By contrasting global and local forecasting approaches, we examine whether prediction markets generate informational value for financial forecasting, and under what conditions this value is most pronounced. Using data from Kalshi prediction markets and a diverse set of US equities and ETFs, we implemented a rigorous walk-forward validation strategy to evaluate model performance. Our analysis reveals that prediction market signals exhibit varying degrees of influence across different financial instruments: macroeconomic indicators such as GDP show broad market impact, while firm-specific signals demonstrate more targeted effects. The findings indicate that simpler models often outperform more complex ones. ARIMAX models show superior performance in local contexts, while in global settings, Linear Regression outperforms XGBoost. The latter struggle to effectively capture the structure and significance of prediction market signals. This study contributes to the financial forecasting literature by demonstrating that prediction markets can be more effective as complementary indicators rather than standalone forecasting tools, while highlighting the importance of model selection and data quality in capturing relationships between prediction market signals and asset returns. These insights have practical implications for data-driven investment firms, suggesting that the integration of prediction market signals should be done selectively and contextually, taking into account both the nature of the target asset and the type of signal used.

Acknowledgments

I express my sincere gratitude to Delphia for the trust they have placed in me throughout this project. Their support has been invaluable and I am especially grateful for the data they provided which formed the backbone of my research.

I also extend my heartfelt thanks to Professor Vande Kerckhove for his guidance, feedback, and constant encouragement. His expertise has been instrumental in shaping this research and helping me navigate the complexities of this topic.

Finally, I would like to acknowledge all those who have contributed to this project, either directly or indirectly. Your support, insight and encouragement have been greatly appreciated. I am grateful to each one of you.

Contents

1	Introduction	7
1.1	The current world of prediction markets and stock returns	7
1.2	Research question	8
1.3	Managerial relevance	9
1.4	Thesis structure	10
2	Literature Review	11
2.1	Prediction markets	11
2.2	Panel Data	12
2.3	Evolution of Artificial Intelligence	13
2.4	Artificial Intelligence in Finance	14
2.5	Comparative modeling strategy: Global vs. Local approaches	15
2.5.1	Global Modeling	15
2.5.2	Local Modeling	18
2.5.3	Advantages and Limitations of Global and Local Approaches	20
3	Methodological framework	22
3.1	Data sources	22
3.1.1	Prediction Market Data	23
3.1.2	Financial Market Data	24
3.1.3	ETF Market Data	25
3.2	Data pre-processing	25
3.2.1	Common Preprocessing Pipeline	25
3.3	Methodology : Global and Local Approaches	28
3.3.1	Global Model	28
3.3.2	Local Models	29
3.4	Evaluation Metrics	30

	3.4.1	Measure of the tradeoff of bias-variance	30
	3.4.2	Other Metrics: RMSE and Diebold-Mariano test	30
	3.5	Questions Derived from the Framework	31
4	Modeling		32
	4.1	Technical Specifications	32
	4.1.1	Global Modeling	32
	4.1.2	Local Modeling	33
	4.2	Forecast Error Evaluation	34
5	Analysis of Results		35
	5.1	Results from Global Modeling	35
	5.1.1	Global Models performance	35
	5.1.2	Learning Curve	36
	5.1.3	Global Model Feature Importance (XGBoost)	36
	5.2	Results from Local Modeling	37
	5.2.1	Stationarity Assessment	37
	5.2.2	Granger Causality Results	37
	5.2.3	Local Models Performance	38
	5.3	Cross-Strategy Performance Comparison	39
	5.3.1	Comparison of ARIMAX and linear regression	39
	5.3.2	Comparison of XGBoost at the local and global level	40
	5.4	Questions Evaluation	40
6	Discussion		42
	6.1	Methodological Insights	42
	6.2	Limitations and Challenges	43
	6.3	Practical Implications for Investors	43
7	Conclusion		45
8	Appendix		52

1 Introduction

In a rapidly evolving financial landscape, the role of information in shaping market behavior has become increasingly significant. One of the most intriguing and complex challenges is understanding the relationship between prediction markets and their influence on stock returns movements. This thesis addresses the problem of quantifying how predictive signals derived from prediction markets impact stock returns.

1.1 The current world of prediction markets and stock returns

In today's financial ecosystem, markets are increasingly driven by an overwhelming volume of heterogeneous and rapidly evolving information. This includes macroeconomic indicators, corporate disclosures, geopolitical developments, and social media sentiment, all of which influence investor decision-making (Li et al., 2014). Within this landscape, prediction markets, also known as event futures or information markets, emerge as a novel and decentralized tool that aggregates dispersed beliefs about future events. These markets provide probabilistic forecasts that have shown surprising accuracy in domains ranging from elections to economic trends (Wolfers and Zitzewitz, 2004; Berg, Forsythe, et al., 2008).

Prediction markets function under the premise of the "wisdom of crowds", capturing collective intelligence to produce forward-looking indicators that often outperform expert analysis (Surowiecki, 2004). By pricing contracts on outcomes such as GDP growth, Federal Reserve policy decisions, or corporate earnings, these platforms generate real-time predictions that can be integrated into financial models.

The relevance of studying the relationship between prediction markets and stock returns is manifold. First, stock returns are inherently forward-looking, reflecting aggregated market expectations of future firm performance and macroeconomic conditions. Therefore, understanding how prediction market signals correlate or even precede market movements can offer a powerful lens to anticipate returns or manage risk.

Second, recent developments in artificial intelligence (AI) and machine learning provide

new tools to extract signals from complex noisy data, including prediction market prices. AI models can uncover non-linear relationships and hidden patterns, enhancing the predictive power of these markets when applied to financial forecasting (Gu et al., 2020; Kelly et al., 2019). In addition, AI can automate the assimilation of prediction market outputs into trading strategies, offering a potentially valuable alpha in increasingly efficient markets.

Bridging prediction markets and stock returns also opens avenues for behavioral finance, exploring how collective beliefs and biases are priced in assets. In addition, it may contribute to market efficiency research, particularly in evaluating how quickly and accurately new information is reflected in stock prices.

This line of research has practical implications for a wide range of stakeholders. Investors and hedge funds could benefit from integrating prediction market data into their decision-making frameworks to improve returns or hedge against risk. Policy makers may rely on these markets to gauge public expectations regarding economic measures or regulatory changes. Corporations can monitor prediction markets to assess investor sentiment around key business events such as earnings announcements or strategic decisions. Finally, academics and financial analysts can use these tools to gain deeper insight into the interaction between belief formation and asset pricing.

1.2 Research question

This thesis addresses an exploratory question: To what extent can prediction markets contribute to forecasting financial market returns, and can these forecasts be enhanced by adopting a global modeling approach that integrates information across multiple assets?

More generally, we seek to understand whether prediction markets provide useful signals for anticipating stock returns and whether the pooling of data through a comprehensive model can improve predictive performance compared to modeling each stock individually. This inquiry also covers the fundamental question of whether stock returns are predictable to any meaningful degree using available information.

To investigate these issues, we compare two modeling frameworks: a global approach, which uses a single model trained on aggregated data from multiple stocks including signals from prediction markets, and a stock-specific approach, where separate models are tailored to individual assets. This comparison aims to shed light on the potential benefits and limitations of using shared information for financial prediction.

1.3 Managerial relevance

This research has significant managerial relevance for Delphia, a Canadian fintech focused on data-driven investment strategies. By analyzing how predictive signals from prediction markets influence stock returns, Delphia can enhance its investment models. Integrating such signals into its algorithms can improve decision-making processes and boost portfolio performance, as suggested by Waitz and Mild, 2013 and Berg and Rietz, 2003.

Prediction markets, by aggregating public sentiment and expectations about future events, offer valuable insights to anticipate market trends. When combined with current Delphia data models, these signals could improve the precision of forecasts, enabling more informed investment decisions and refined risk management practices.

Moreover, prediction market data can help detect market inefficiencies and emerging trends at an early stage. This proactive capability would enable Delphia to respond quickly to changes in market dynamics, creating a competitive advantage through timely action. Enhanced risk detection is another benefit, particularly in volatile or uncertain conditions, where prediction signals can support more robust portfolio protection and reduced exposure to high-risk assets.

The findings may also open the door to new and innovative investment products. For example, funds specifically designed to exploit patterns in prediction market data could offer clients differentiated and potentially performing options. By broadening its data ecosystem to include such signals, Delphia can further establish itself as a forward-looking fintech leader, leveraging strategic information to attract clients and increase market presence.

Ultimately, this research aligns with Delphia’s mission to harness data to make smarter investment decisions. The insights gained can sharpen its strategic edge, strengthen forecasting capabilities, and support continued innovation in a rapidly evolving financial landscape.

1.4 Thesis structure

This thesis is structured into seven main chapters, each contributing to the comprehensive analysis of prediction market signals’ impact on stock returns through a comparative modeling framework. Chapter 1 introduces the research context, highlighting the significance of prediction markets in today’s financial landscape, and outlining the research objectives and managerial relevance. Chapter 2 presents a comprehensive review of the literature, examining prediction markets, panel data methodologies, and the evolution of artificial intelligence in finance, with particular emphasis on comparative modeling strategies between global and local approaches. Chapter 3 details the methodological framework, describing the data sources (prediction market data, financial market data, and ETF market data), preprocessing procedures, and the implementation of global and local modeling approaches. Chapter 4 focuses on the technical specifications of the modeling process, including the implementation details of global and local models, along with the forecast error evaluation methodology. Chapter 5 presents a comprehensive analysis of the results, beginning with global modeling outcomes (including model performance, learning curves, and feature importance), followed by local modeling results (stationarity assessment, Granger causality, and model performance), and concluding with cross-strategy performance comparisons. Chapter 6 discusses the key findings, methodological insights, limitations, and practical implications for investors. Finally, Chapter 7 concludes the research by summarizing the main insights and contributions to both research and practice.

2 Literature Review

This section examines previous theories and research on the valuation of information and its influence on financial markets. Reviews key studies, theoretical frameworks, and findings relevant to understanding the connection between prediction markets and stock market responses.

2.1 Prediction markets

Prediction markets are platforms where participants trade contracts that pay out based on the outcome of future events (Wolfers and Zitzewitz, 2004). These markets aggregate the individual predictions of participants, turning them into a collective forecast of the likelihood that an event will occur. The concept of prediction markets is rooted in the idea that the collective wisdom of a diverse group of individuals, each with access to different information and insights, can generate more accurate forecasts than any individual expert (Bossaerts et al., 2022). These markets function similarly to financial markets (Wolfers and Zitzewitz, 2004), where participants buy and sell contracts based on their predictions of future outcomes. For example, participants might buy a contract that pays out if a specific political event occurs, such as the election of Trump against Kamala Harris during the 2024 presidential race (Mongrain and Stegmaier, 2024), or if a company meets a certain financial target.

One of the key features of prediction markets is their ability to aggregate information. As discussed by Bossaerts et al., 2022, these markets incorporate diverse viewpoints, enabling them to reflect the collective intelligence of participants. This phenomenon is often referred to as the "wisdom of crowds". Berg, Nelson, and Rietz, 2008, have shown that prediction markets can outperform individual experts in terms of forecasting accuracy, as they incorporate a wide range of opinions.

These markets are typically structured as winner-take-all or vote-share markets. In the winner-take-all markets (Dai et al., 2021), participants predict a single outcome, and those who predict the correct event share the prize. This market model offers a fixed payout for the correct prediction, which is typically the same for all winners. Vote-share markets

involve trading based on the estimated proportion of votes or shares in a specific outcome, often used for more detailed or continuous predictions, such as estimating the exact share of votes in an election or market share in a competition (Dai et al., 2021).

Research has shown that prediction markets can effectively predict not only political events, but also economic trends, such as the likelihood of a recession or the movement of stock prices (Wolfers and Zitzewitz, 2004). These markets function as a tool to forecast economic indicators, company performance, or even geopolitical events. By integrating real-time information and the collective insights of a diverse set of participants, prediction markets offer a unique and valuable source of predictive signals that can inform decision-making in various domains, including finance. For example, Google estimates its market capitalization using prediction markets to forecast its value prior to an initial public offering (Berg et al., 2009).

2.2 Panel Data

Panel data, also known as longitudinal data, refer to datasets that contain observations on multiple entities (such as firms or stocks) over multiple time periods. This structure allows researchers to control for both cross-sectional and temporal variations, enhancing the robustness and accuracy of empirical models (Wooldridge, 2009). In the context of financial research, panel data is particularly advantageous because it enables the modeling of heterogeneous characteristics across assets while capturing dynamic temporal effects such as momentum, volatility clustering, and regime shifts.

The use of panel data facilitates more precise estimation of causal relationships by accounting for unobservable individual-specific effects that are constant over time, as well as for time-specific shocks that affect all individuals simultaneously. Fixed-effects and random-effects models are commonly applied to exploit this structure and mitigate potential omitted variable bias (Baltagi, 2008).

In financial forecasting, especially when dealing with asset returns, panel data allows for the construction of global models that pool information across multiple assets. This approach can improve out-of-sample forecast performance by increasing the effective sample

size and allowing the model to learn from common patterns between different stocks (Gu et al., 2020). Such models are especially powerful in high-dimensional settings where individual time series are too short or noisy to yield reliable predictions on their own.

In the present research, panel data is leveraged to implement a global modeling approach in which one model is trained across a panel of stocks with integrated prediction market signals. This is compared against a local approach based on asset-specific models, in order to assess whether information sharing across assets can enhance prediction accuracy and improve financial insight extraction.

2.3 Evolution of Artificial Intelligence

Artificial Intelligence (AI) has undergone a significant transformation in recent decades, moving from rule-based expert systems to data-driven learning models. The evolution of machine learning (ML) and, more recently, deep learning (DL) has been largely fueled by the exponential increase in computational power, the availability of big data, and algorithmic innovations (LeCun et al., 2015). These advances have led to breakthroughs in fields such as computer vision, natural language processing, and predictive analytics.

The development of ensemble methods (e.g. random forests, gradient boosting machines like XGBoost), regularized regression techniques (e.g. LASSO) and neural networks has significantly broadened the scope of problems AI can address. More recently, Vaswani et al., 2017, demonstrated that transformer models and attention-based architectures significantly advance the ability to reason across sequences by effectively capturing intricate dependencies.

As AI systems have become more sophisticated, they have also gained the ability to handle noisy, nonlinear, and high-dimensional datasets, features that are inherent to financial data. This has opened the door to more accurate, adaptive, and robust decision-making in areas such as credit scoring, fraud detection, and asset management.

2.4 Artificial Intelligence in Finance

AI has found fertile ground in finance due to the large amounts of structured and unstructured data in the sector, the high stakes of predictive accuracy, and the complexity of market dynamics (Fischer and Krauss, 2018). The applications span algorithmic trading, portfolio optimization, sentiment analysis, risk assessment, and anomaly detection (Heaton et al., 2017).

In financial forecasting, machine learning algorithms, such as boosted regression trees (notably XGBoost), have demonstrated superior predictive performance compared to traditional time series models. These models excel at capturing complex, nonlinear relationships and interactions among financial variables, which classical linear methods often fail to detect. For example, Gu et al., 2020 show that gradient-boosted trees effectively leverage cross-sectional and temporal dependencies in asset pricing, outperforming conventional econometric models. Similarly, Krauss et al., 2016 provide empirical evidence that gradient-boosted trees yield robust returns in statistical arbitrage strategies in equity markets, surpassing other machine learning approaches and traditional benchmarks. Such models also have the advantage of integrating heterogeneous data sources, including fundamentals, technical indicators, and increasingly prediction market signals, enabling the synthesis of weak and dispersed signals into actionable forecasts.

Furthermore, the application of AI to panel data, as is done in global modeling frameworks, has shown particular promise. Global models benefit from learning cross-sectional structures (e.g. industry trends, macroeconomic co-movement) and temporal dynamics simultaneously, leading to better generalization and reduced overfitting (Gu et al., 2020).

As highlighted by recent research (Gu et al., 2020; Krauss et al., 2016; Fischer and Krauss, 2018; Heaton et al., 2017), combining traditional economic theory with data-driven AI methods enables the development of hybrid models that maintain interpretability while leveraging the predictive power of complex non-linear algorithms. These approaches are particularly promising in financial modeling, where understanding the underlying economic mechanisms is crucial to achieving accurate forecasts. In this thesis, AI plays a central role in modeling financial returns using signals from prediction markets, no-

tably through scalable, nonlinear models, which effectively exploit both cross-sectional and temporal dependencies within stock panels.

2.5 Comparative modeling strategy: Global vs. Local approaches

2.5.1 Global Modeling

Global modeling refers to the strategy of training a single predictive model on a panel of multiple time series. In the context of financial forecasting, this involves pooling data from various assets into a unified data set and fitting one model that attempts to learn shared patterns throughout the cross section.

The central hypothesis of this approach is that financial assets exhibit common mechanism of generation, such as macroeconomic influences or systematic behavioral patterns, which can be jointly modeled (G. Feng et al., 2018; Gu et al., 2020; Montero-Manso and Hyndman, 2021). By aggregating information across assets, the global approach benefits from a larger effective sample size, potentially improving generalization and reducing overfitting (Hewamalage et al., 2021; Bontempi et al., 2013). Furthermore, global models are well suited for scenarios where exogenous signals (such as prediction market indicators) are expected to have similar effects across assets (Montero-Manso and Hyndman, 2021).

In practice, global modeling has been implemented using machine learning techniques that scale well with large datasets and are capable of capturing complex dependencies across series. These include:

1. Linear Regression models the relationship between a dependent variable y_t and explanatory variables $\mathbf{x}_t$ as:

$$y_t = \mathbf{x}_t^\top \boldsymbol{\beta} + \varepsilon_t,$$

where $\boldsymbol{\beta}$ is the vector of coefficients to estimate and ε_t is an error term assumed to be zero mean and independent (Montgomery, 2012).

When applied globally, linear regression pools data from multiple time series, estimating common coefficients across assets. This leverages larger sample sizes, reducing variance but potentially inducing bias if asset relationships differ significantly (Bontempi et al., 2013). Applied locally, separate regressions are fitted for each asset, capturing idiosyncratic relationships but increasing variance due to smaller sample sizes (Zivanovic, 2001).

2. Long Short-Term Memory (LSTM) networks are a specialized form of Recurrent Neural Networks (RNNs) designed to overcome the limitations of traditional RNNs in learning long-range dependencies. The model was originally introduced by Hochreiter and Schmidhuber, 1997. By introducing gating mechanisms, LSTMs regulate the flow of information through the network:

$$\left\{ \begin{array}{ll} f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) & \text{(Forget gate)} \\ i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) & \text{(Input gate)} \\ o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) & \text{(Output gate)} \\ \tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) & \text{(Candidate cell state)} \\ c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t & \text{(Updated cell state)} \\ h_t = o_t \odot \tanh(c_t) & \text{(Updated hidden state)} \end{array} \right. \quad (1)$$

Each component in the LSTM architecture serves a distinct function:

- Forget gate (f_t): Determines which information from the previous cell state c_{t-1} should be discarded. Sigmoid activation ensures outputs between 0 and 1, effectively “forgetting” or “retaining” specific parts of the memory.
- Input gate (i_t): Controls which new information from the current input x_t and the previous hidden state h_{t-1} should be stored in the cell state.
- Candidate cell state ($\tilde{c}_t$): Represents the new content to be potentially added to the cell state, passed through a hyperbolic tangent to ensure values stay in a manageable range.
- Cell state update (c_t): Combines the retained previous state and the new candidate state, weighted respectively by the forget and input gates, enabling both memory retention and controlled updating.
- Output gate (o_t): Decide which parts of the cell state should be exposed to the next hidden state.

- Hidden state (h_t): The actual output of the LSTM cell, modulated by the output gate and passed through a tanh nonlinearity.

These gates allow the network to selectively retain or discard information, allowing LSTMs to effectively capture long-term temporal dependencies (Hochreiter and Schmidhuber, 1997).

LSTMs have been shown to be effective in global modeling contexts, where they can learn shared temporal structures across multiple time series (Hewamalage et al., 2021). However, their reliance on substantial training data per series limits their applicability in local modeling scenarios.

3. XGBoost (Extreme Gradient Boosting) aggregates K regression trees $f_k \in \mathcal{F}$ to form predictions as

$$\hat{y}_i = \sum_{k=1}^K f_k(\mathbf{x}_i),$$

where each tree f_k is added sequentially to minimize a regularized objective function that combines a convex loss and a complexity penalty (Chen and Guestrin, 2016):

$$\mathcal{L}(\phi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k), \quad \text{with} \quad \Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2,$$

where T is the number of leaves and w are the weights of the leaves. This regularization controls model complexity and helps prevent overfitting. XGBoost employs the second-order Taylor expansion of the loss to optimize the addition of new trees efficiently. Moreover, it uses a novel sparsity-aware algorithm to handle missing values and supports parallelized tree construction, significantly improving training speed and scalability (Chen and Guestrin, 2016).

In financial forecasting contexts, XGBoost can be trained globally on pooled data, capturing complex non-linear relationships shared between assets (Gu et al., 2020). Alternatively, it can be applied locally to individual assets, enabling more tailored modeling but increasing the risk of overfitting, especially with limited data, as noted by Bandara et al., 2020. The algorithm's flexibility and robustness make it a strong candidate for time series

prediction tasks involving heterogeneous datasets.

4. While classical econometric models such as VAR-X and VECM-X provide strong theoretical frameworks for modeling interdependencies and cointegration in multivariate time series, they rely on stringent assumptions such as linearity, stationarity, and normally distributed errors. These assumptions are often violated in real-world financial data, which are typically noisy, nonstationary, and exhibit nonlinear relationships (Gu et al., 2020; Sirignano and Cont, 2019). In contrast, machine learning models such as XGBoost and LSTMs offer a more flexible and scalable framework for capturing complex, nonlinear dynamics without the need for extensive pre-specification of the functional form or strict distributional assumptions (Bandara et al., 2020; Hewamalage et al., 2021).

Moreover, global machine learning models can generalize patterns across a broad cross section of assets, which improves their robustness in high-noise environments and enhances predictive stability in low-data regimes (Bontempi et al., 2013; Montero-Manso and Hyndman, 2021). Their scalability to large datasets, combined with capabilities such as automated feature selection and handling of missing or irregular data, makes them particularly attractive in modern forecasting contexts where heterogeneous and high-dimensional inputs (such as prediction markets or alternative data) are integrated (Chen and Guestrin, 2016). As such, while VAR-X and VECM-X remain valuable for interpretability and causal inference, the predictive power, adaptability, and computational advantages of machine learning models provide compelling reasons for their adoption in global financial modeling strategies.

2.5.2 Local Modeling

Local modeling, on the contrary, involves building and calibrating a distinct predictive model for each individual asset. Each time series is treated separately with a model tailored to its unique characteristics and historical dynamics.

This strategy assumes that financial assets differ substantially in their behavior due to firm-specific factors, sectoral influences, or idiosyncratic responses to external events (Lopez de Prado, 2018; Tashman, 2000). Local models can therefore better capture these

individualized patterns, especially when relationships between variables vary significantly across assets. However, this granularity comes at the cost of a reduced data volume per model and an increase in the complexity of model management (Petropoulos et al., 2022).

Traditionally, local modeling approaches have predominantly relied on classical statistical time series techniques, which are well suited to capture the distinct temporal dynamics of each asset. A prominent example is

1. The auto-regressive integrated moving average model (ARIMA) of order (p, d, q) is given by:

$$\Phi(B)(1 - B)^d y_t = \Theta(B)\varepsilon_t \quad (2)$$

where:

- y_t is the observed time series,
- B is the backshift operator such that $By_t = y_{t-1}$,
- $(1 - B)^d$ is the differencing operator of order d ,
- $\Phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$ is the autoregressive (AR) polynomial of order p ,
- $\Theta(B) = 1 + \theta_1 B + \dots + \theta_q B^q$ is the moving average (MA) polynomial of order q ,
- ε_t is a white noise error term.

In some cases, one or more exogenous variables $\mathbf{x}_t$ can be incorporated into the model, resulting in an ARIMAX specification:

$$\Phi(B)(1 - B)^d y_t = \Theta(B)\varepsilon_t + \beta^\top \mathbf{x}_t \quad (3)$$

where $\mathbf{x}_t$ represents a vector of external covariates and β denotes the corresponding coefficients. Including exogenous inputs allows the model to account for external influences on the target time series, enhancing predictive performance in many practical applications. This formulation follows the classical time series modeling framework introduced by Box et al., 2015.

2.5.3 Advantages and Limitations of Global and Local Approaches

The decision between global and local modeling strategies highlights a fundamental trade-off in statistical learning: the balance between bias and variance, as discussed by Petropoulos et al., 2022 as well as factors such as data efficiency and operational complexity.

Global models, trained on pooled data from multiple assets, benefit from a larger effective sample size. This typically results in lower variance in parameter estimation and more stable predictions, especially when individual time series are short or noisy (Hewamalage et al., 2021; Bontempi et al., 2013). Moreover, by capturing shared temporal structures, global models exploit cross-sectional information that isolated local models might miss (Montero-Manso and Hyndman, 2021; G. Feng et al., 2018). In practice, global models also offer significant operational advantages, as fewer models need to be trained and updated, a benefit emphasized by Januschowski et al., 2022 in large-scale industrial applications.

However, this data aggregation introduces the risk of model misspecification: if asset-level behaviors differ substantially from the average pattern, a global model may introduce systematic bias, underfitting asset-specific dynamics (Petropoulos et al., 2022; Bandara et al., 2020). This risk is particularly acute in financial contexts where heterogeneity across assets is high, and where structural breaks or firm-specific news affect returns in a non-uniform way.

Local models, by contrast, tailor forecasts to each individual asset, offering greater flexibility to capture unique autocorrelation structures, volatility patterns, or sensitivity to exogenous variables (Lopez de Prado, 2018; Box et al., 2015). This approach reduces bias, as each model is free to specialize in the idiosyncrasies of its target series, and has been shown to improve predictive accuracy in contexts characterized by high heterogeneity (Tashman, 2000).

Yet, this increase in specificity comes at the cost of higher variance. When time series are short, local models may overfit the noise, leading to poor generalization performance on unseen data (Januschowski et al., 2022). Furthermore, the proliferation of models leads to

higher computational and maintenance costs, which can pose scalability challenges in systems with hundreds or thousands of assets (Bandara et al., 2020; Petropoulos et al., 2022).

In summary, the literature highlights that the relative performance of global and local models depends on the structure of the data and the characteristics of the forecasting task. From a bias-variance perspective:

- Global models reduce variance but risk higher bias when cross-series heterogeneity is high;
- Local models favor lower bias but higher variance, particularly when data is limited.

As several studies suggest (Bandara et al., 2020; Hewamalage et al., 2021), no single approach consistently outperforms across all settings. Hybrid or hierarchical strategies, which cluster similar time series and apply global models within those groups, are increasingly explored as a compromise to manage this trade-off effectively.

3 Methodological framework

Building on the comprehensive literature review presented in the previous section, this chapter outlines the methodological framework employed to address the research question introduced in Section 2. Our approach leverages insights from prediction markets in Section 2.1, panel data structures in Section 2.2, and advanced artificial intelligence techniques in Section 2.4, specifically tailored for financial forecasting.

Prediction markets serve as a critical source of collective real-time information aggregation, offering signals that reflect the beliefs of diverse participants about future outcomes. Integrating these signals into forecasting models requires methodologies capable of capturing both temporal dynamics and cross-sectional heterogeneity across financial assets.

To this end, panel data methodologies are utilized, allowing us to exploit observations across multiple stocks over time. This allows for control of both asset-specific characteristics and temporal effects, improving model robustness and interpretability.

Furthermore, the evolution of AI, from traditional machine learning models such as XGBoost to recurrent neural networks and probabilistic forecasting methods, provides powerful tools for modeling complex, nonlinear relationships inherent in financial data. These techniques are deployed within both global and local modeling paradigms: the global approach pools data across assets to capture shared patterns, while the local approach focuses on asset-specific models that can capture unique dynamics.

The subsequent sections detail the data sources, feature engineering process, model architectures, and evaluation metrics used in this study. All analyses and source code are available on a GitHub repository to ensure transparency and reproducibility.

3.1 Data sources

The empirical analysis is based on three primary categories of data, each serving a complementary purpose in the modeling and evaluation process.

3.1.1 Prediction Market Data

Forecast data were obtained from Kalshi Inc., 2025, a regulated prediction market platform that offers contracts on a broad spectrum of macroeconomic, geopolitical and firm-specific events. The selection of prediction markets was based on the availability of data for the year 2024. Only markets with sufficiently long horizons and potential economic relevance were retained. In particular, short-term or event-specific contracts, such as individual baseball games, were excluded to focus on broader signals that could plausibly be related to market dynamics.

In total, 16 different prediction markets were included, which can be grouped into three main categories. The first category covers macroeconomic indicators, such as US inflation rates and GDP growth. The second category comprises firm-specific metrics, including Tesla vehicle production, Netflix subscriber growth, and Meta’s daily active users. Finally, the third category consists of event-based or exogenous signals, such as the number of hurricanes and the incidence of measles.

Some of these markets, particularly those related to corporate performance or macroeconomic variables, have an obvious potential connection to financial market behavior. Others, such as hurricane counts or measles incidence, may at first appear to be unrelated to asset returns. However, their inclusion allows to test whether seemingly exogenous and unrelated events can carry predictive content for financial markets, either directly or through correlated risk sentiment. The data coverage varies across contracts: some markets were active throughout most of 2024, while others only became available starting in August and remained accessible through the end of 2024, which reduces their historical depth. Additionally, update frequency differed, with certain markets featuring multiple updates per day, and others only once per day.

Table 1 provides an overview of the number of observations collected for each prediction market included in the study. This variation reflects both differences in contract launch dates and user engagement on the platform. The diversity in temporal granularity also emphasizes the importance of resampling and standardization during preprocessing.

In cases where multiple updates occurred within the same day, the values were averaged to produce a single daily observation. The resulting daily resolution data set is summarized in Appendix 1, which ensures consistency for integration into time-series models.

Market	Number of observations
Apple car	589
Bitcoin (BTC)	460
Ethereum (ETH)	458
Federal Reserve (Fed)	1819
U.S. Gas Prices	435
GDP Growth	1387
Google CEO	568
Hurricane Counts	478
Inflation	560
Tech Layoffs	1707
Measles Incidence	544
Meta DAU	1099
Netflix Subs	1259
SpaceX Launches	445
Tesla Production	1557
WTI Oil	468

Table 1: Prediction markets and corresponding number of observations in 2024.

3.1.2 Financial Market Data

Stock return data for a curated selection of 17 US equities were extracted from the database provided by Delphia Technologies Inc., 2025, a data-driven fintech company. These daily return series span from January 2, 2024 to November 27, 2024.

The selected stocks include prominent companies such as Tesla, Netflix, Meta, Google, and others. The full list of tickers and their respective companies is provided in Appendix 2. This selection covers a diverse set of sectors including technology, energy, finance, pharmaceuticals, automotive, and retail. The variety of firms was chosen to reflect the wide range of economic and political events that can be captured by the prediction markets and to assess whether such signals have predictive value in different industries.

These stock returns serve as the main dependent variable in the modeling phase, where we examine the explanatory and predictive power of Kalshi-based signals on daily equity returns.

3.1.3 ETF Market Data

To examine whether the relationships observed between Kalshi signals and individual stock returns also hold for broader asset classes, exchange-traded funds (ETFs) were incorporated into the analysis. Return data for three major ETFs were downloaded and pre-processed using a Python script that leverages the yfinance library (Aroussi, 2025), which provides a convenient interface to retrieve financial data from Yahoo Finance, 2025. The ETFs included the S&P 500 ETF (SPY), the MSCI World Index ETF (IWDA.AS), and the Nasdaq 100 ETF (QQQ).

The script ensured that the ETF return data were aligned with the exact date range used for the stock-level analysis (2 January 2024 to 27 November 2024), allowing a consistent temporal comparison across datasets. These ETFs were selected to represent a broad spectrum of geographical and sectoral exposures. Their inclusion allows the study to assess whether Kalshi-derived signals exhibit predictive power not only at the individual equity level but also in the context of aggregated market indices.

3.2 Data pre-processing

This section describes the data sources and the pre-processing steps applied to prepare the datasets for empirical modeling, distinguishing between the procedures used for the global and local modeling strategies. The implementation was done in R, with all packages used listed in Appendix 3.

3.2.1 Common Preprocessing Pipeline

Daily Aggregation: Raw Kalshi prediction market data came with heterogeneous temporal structures.

Timestamp	ticker	Value	quarter
2024-01-26 14:00:00	Tesla	481566.5	Q1
2024-01-26 15:00:00	Tesla	481711.8	Q1
2024-01-26 18:00:00	Tesla	480876.7	Q1
2024-01-26 21:00:00	Tesla	480876.7	Q1

To ensure compatibility with daily financial returns and to address issues related to temporal misalignment between forecast and return series (see W. Feng et al., 2023), all prediction signals were aggregated to a daily frequency using a custom function (`process_df_daily`), as described in Algorithm 1 in Appendix 4. This function computes the mean of all intraday values for each contract and date, producing one representative prediction value per day per event.

date	id	pred daily
2024-01-26	Tesla	481257.9

Missing Value: The handling of missing data differed according to the type of data. It is important to note that target variables and exogenous variables cannot be treated in the same way when addressing missing values. For target variables, missing data often implies the impossibility of directly learning from those samples and may require specific imputation strategies or the exclusion of such instances. On the other hand, exogenous variables, which serve as predictors, can often be imputed using statistical or machine learning methods without invalidating the overall modeling process. This distinction is crucial to ensure the integrity and performance of the predictive model.

- **Prediction Market Data:** Between prediction periods (e.g., from Q1 to Q2), some dates were missing due to inactive or unpublished forecasts. To reconstruct a complete daily timeline for each contract, a Kalman filter was applied using a local linear trend state-space model implemented with the `SSModel` function from the `KFAS` package. Smoothed state estimates filled in missing values, ensuring temporal continuity across contract boundaries, following the methodology recommended by Durbin and Koopman, 2012. The detailed algorithm for applying the Kalman filter is presented in Algorithm 2 in Appendix 5.

- **Financial Market Data:** Stock and ETF return data exhibited missing values corresponding to weekends and public holidays, non-trading days during which markets are closed. After verifying that these gaps aligned with expected calendar closures, no imputation was performed. Instead, `na.omit()` was applied to clean the dataset, ensuring that only valid trading sessions (typically five days per week) were retained for modeling purposes.

Data Integration: Once the prediction signals and the financial return series were processed, the two data sets were merged based on the available trading dates of the financial market data, that is, the returns of stocks and ETF. This approach ensured temporal alignment and

eliminated any potential look-ahead bias, by guaranteeing that only information available at or prior to each trading day was used for forecasting. By anchoring the integration to the financial data timeline, we avoided including future prediction signals that would not have been observable at the time of the actual market movement. The final data set consisted of: (i) daily asset returns as target variables, (ii) corresponding prediction market signals, and (iii) metadata such as date, asset ticker, and event identifiers.

Date	id	pred daily	ticker	returns
2024-01-29	Tesla	-470.67267	AAPL	-1.458257e-03
2024-01-29	Tesla	-470.67267	AMZN	1.204971e-02
2024-01-29	Tesla	-470.67267	BAC	6.287425e-03
...	...	...	...	...
2024-01-30	Tesla	-838.51256	AAPL	-1.518802e-02
2024-01-30	Tesla	-838.51256	AMZN	-1.057872e-02
2024-01-30	Tesla	-838.51256	BAC	2.625369e-02
...	...	...	...	...
2024-05-22	Netflix	0.029503250	AAPL	-0.0070995761
2024-05-22	Netflix	0.029503250	AMZN	-0.0040787470
2024-05-22	Netflix	0.029503250	BAC	0.0088809947
...	...	...	...	...
2024-11-26	Apple	4.000	XOM	-0.0130511169
2024-11-26	Apple	4.000	SPY	0.0052214111
2024-11-26	Apple	4.000	IWDA.AS	0.0011463223

To ensure the integrity of temporal causality and prevent future information from influencing model training or evaluation, the data integration process was anchored to financial market timestamps. This approach ensured that only Kalshi prediction signals published before or on the same day as asset returns were used. Although future data might technically be available in the full dataset, it was carefully excluded from the training and testing process to avoid any potential look-ahead bias.

Data splitting: In line with the best practices for data split in time series recommended by Hyndman and Athanasopoulos, 2018, we adopted a walk-forward validation approach. This method ensures strict temporal separation between training and test sets, eliminating look-ahead bias and preserving the integrity of the forecast evaluation.

The procedure is implemented using the `time_series_cv()` function, as detailed in Algorithm 3 (see Appendix 6). After ensuring that the data set is ordered chronologically, the function creates two validation folds with a cumulative training window of 120 days and a fixed test window of 40 days. The cumulative nature of the training window, as suggested by Hyndman and

Athanasopoulos, 2018, enables the model to take advantage of an expanding historical context, which is more reflective of operational forecasting settings. The skip parameter is set to 40 days to control the temporal shift between validation folds, and the slice limit is set to 2 to limit the number of splits for simplicity.

This approach ensures that each fold preserves the chronological order of the data, using earlier observations for training and subsequent ones for testing. In the first fold, the training set contains 35,368 observations, and the test set includes 8,656 observations. In the second fold, the training set is cumulative and includes 44,024 observations, while the test set contains 8,656 observations. This setup allows for robust out-of-sample evaluation while minimizing look-ahead bias.

By relying on two distinct slices rather than a single train-test split, this setup allows evaluation metrics (see Section 3.4) to reflect performance over multiple forecast origins, thus providing a more robust assessment of model accuracy.

Daily aggregation, missing value handling, data integration, and data splitting are consistently applied at the global and local levels to ensure that the evaluation metrics for local and global models are directly comparable.

3.3 Methodology : Global and Local Approaches

After data collection in Section 3.1 and data cleaning in Section 3.2, the methodology consists of implementing two distinct predictive modeling approaches: a global approach and a local approach. These two methods allow us to compare the performance of the model when applied either to the entire data set or to specific subsets.

3.3.1 Global Model

Although recurrent neural networks (RNN) and their LSTM variants have demonstrated a strong ability to capture complex temporal dependencies in sequential data (Hewamalage et al., 2021; Montero-Manso and Hyndman, 2021), their training often requires large datasets and extensive computational resources. Additionally, these deep learning models tend to lack interpretability, which is a key consideration in financial applications where understanding feature importance and model behavior is critical.

XGBoost offers a compelling alternative due to its high predictive accuracy, scalability, and robustness to feature engineering, while maintaining relatively faster training times and better interpretability through feature importance metrics (Chen and Guestrin, 2016; Gu et al., 2020). Furthermore, gradient-boosted trees can effectively capture non-linear interactions often present in financial data without requiring sequential modeling assumptions inherent to RNNs.

In the next Section 4, we implement XGBoost. To assess whether XGBoost introduces a truly significant increase in complexity, a linear regression model will also be produced. This regression model is considered a baseline by Montero-Manso and Hyndman, 2021 and will serve as a benchmark for comparison.

3.3.2 Local Models

ARIMA models are widely used for univariate time series forecasting and are effective in capturing temporal autocorrelation as well as nonstationarities such as trends or seasonality (Box et al., 2015). Their well-established methodology and interpretability make them a suitable choice as a local model to predict each individual asset.

In this study, we specifically use an ARIMAX model (AutoRegressive Integrated Moving Average with eXogenous variables) to locally model each asset return series, as it explicitly allows the inclusion of exogenous variables derived from prediction markets (economic event predictions, firm-specific or exogenous variables). These external variables enrich the model by providing additional information that can improve forecasting accuracy (Box et al., 2015; Hyndman and Athanasopoulos, 2018).

Additionally, a local XGBoost machine learning model is also implemented for each asset to compare its performance with that of the ARIMAX. XGBoost is a powerful gradient boosting algorithm that sequentially builds an ensemble of decision trees to enhance predictive accuracy. Its ability to model complex and nonlinear relationships between explanatory variables and the target variable makes it an interesting candidate for local return forecasting. However, it may pose an increased risk of overfitting, especially when data is limited (Chen and Guestrin, 2016; Bandara et al., 2020).

3.4 Evaluation Metrics

3.4.1 Measure of the tradeoff of bias-variance

One of the central challenges in supervised learning is managing the bias-variance trade-off, which refers to the balance between two sources of error that affect model performance. Bias is the error introduced by approximating a real-world problem, which may be complex, by a simplified model. Variance, on the other hand, measures the model's sensitivity to fluctuations in the training set. A model with high bias tends to underfit the data, failing to capture the underlying trend. In contrast, a model with high variance overfits the training data, capturing noise as if it were signal.

To quantitatively evaluate this bias-variance trade-off, we rely on the Mean Squared Error (MSE), which can be decomposed into three components: noise (irreducible error), variance, and bias squared, as defined by Petropoulos et al., 2022:

$$\text{MSE} = \text{noise} + \text{variance} + \text{bias}^2 \quad (4)$$

To estimate generalization performance and better assess tradeoff, we use the k -fold cross-validation. This technique allows to compute the average MSE in several train-test splits, thus giving a more reliable estimation of the model error on unseen data and helping to distinguish whether the error stems from bias or variance.

3.4.2 Other Metrics: RMSE and Diebold-Mariano test

We include alternative metrics to gain a more comprehensive view of the performance of the model. The root mean squared error (RMSE) provides an interpretable measure of the average prediction error in the same unit as the dependent variable, which helps in comparing the results across assets.

The Diebold-Mariano (DM) test (Diebold and Mariano, 1995) is employed to statistically compare the forecast accuracy of two competing models. The test is based on the null hypothesis that both models have equal predictive accuracy, evaluated using a loss differential series (typically squared errors).

$$H_0 : \mathbb{E}[d_t] = 0 \quad (\text{equal predictive accuracy})$$

$$H_1 : \mathbb{E}[d_t] \neq 0 \quad (\text{difference in predictive accuracy})$$

where $d_t = L(e_{1,t}) - L(e_{2,t})$, with $L(\cdot)$ being a loss function (e.g. squared error), and $e_{1,t}$, $e_{2,t}$

the forecast errors from the two models.

A statistically significant result (typically $p < 0.05$) leads to the rejection of H_0 , indicating that one model significantly outperforms the other in terms of forecast accuracy. This test helps determine whether improvements from incorporating prediction market signals are not only large in magnitude but also statistically meaningful.

3.5 Questions Derived from the Framework

In addition to these primary research questions, several other important issues arise at this stage of the analysis.

- Q1: To what extent do prediction market signals enhance stock return forecasting, and is this enhancement more pronounced in local (stock-specific) models than in global (market-wide) models?

This question investigates whether granular information from prediction markets is better captured when forecasting models are tailored to individual stocks rather than the overall market. The rationale is that local models might be more sensitive to asset-specific information embedded in prediction market signals, potentially yielding higher predictive accuracy.

- Q2: Does the impact of prediction market signals on forecasting performance differ between individual stocks and broader financial instruments such as ETFs?

This question explores whether prediction markets exert a more direct or discernible influence on individual stocks compared to more diversified instruments like exchange-traded funds (ETFs), which aggregate multiple assets. The underlying idea is that the signal-to-noise ratio might be higher for firm-specific assets, making the informational content of prediction markets more effective in that context.

4 Modeling

The previous section (Section 3) outlined the methodological framework used to address our research question. We now turn to the implementation details of both our global and local modeling strategies.

4.1 Technical Specifications

4.1.1 Global Modeling

In the global modeling framework, the objective is to identify generalizable relationships between the Kalshi forecast signals and the returns of financial assets by training a single unified model across all stocks. In contrast to the individual asset modeling approach, this method employs a shared architecture to emphasize predictive features that remain consistent across companies, contracts, and time periods.

Following the pre-processing and data integration steps described in Section 3.2, the data set was structured as a longitudinal panel that covers all the tickers. Each observation corresponds to a unique combination of trading date and stock symbol, with Kalshi signals serving as explanatory variables and daily stock returns as the response variable. To evaluate model performance in a temporally coherent manner, we implemented a two-fold walk-forward validation strategy.

Regarding feature engineering (see Algorithm 5 in Appendix 7), for each training set derived from the two folds mentioned above, the `recipe()` function from RStudio was used to define a data preprocessing plan. Time-based features were extracted using the `step_timeseries_signature()` function, enabling the model to capture seasonal and temporal dynamics. Finally, categorical variables, such as prediction market identifiers, were transformed into binary numerical variables via one-hot encoding. This feature engineering approach follows best practices for panel time series forecasting, as outlined by Dancho, 2021.

Subsequently, the `workflow()` function, see Algorithm 5, was used to fit the desired model, as introduced in Section 3.3. We trained two models: an XGBoost regression model, and a global linear regression model serving as a baseline. Both models were trained and evaluated on the test sets derived from the two walk-forward folds. This setup allows us to assess whether the additional complexity of the XGBoost model is statistically significant and justifiable. The workflow integrates the preprocessing steps from the recipe with the model specification, us-

ing `boost_tree()` or `linear_reg()` to define the model and setting the engine to `xgboost` or `lm`, respectively. The preprocessing steps are attached using `add_recipe()`, ensuring proper data preparation before fitting. The model training is then performed using the `fit()` function on the training data.

Once the model is trained, it is calibrated on the test data using the `modeltime_calibrate()` function. This function assesses model performance on unseen data. Accuracy metrics, such as RMSE, are then computed using the `modeltime_accuracy()` function. Evaluation is carried out globally and per ticker to analyze performance in individual time series, in line with the methodology recommended by Dancho, 2021.

4.1.2 Local Modeling

The local modeling strategy involves developing individual models for each asset, which allows a more accurate representation of the specific dynamics of the asset. This section presents the key preprocessing steps undertaken within this localized framework.

For each asset, the data set is filtered to retain only the return series and the relevant Kalshi contracts associated with that particular stock. This filtering process ensures the exclusion of irrelevant signals, thus reducing noise and dimensionality during the ARIMA model construction.

Given that local ARIMAX models are particularly sensitive to the temporal characteristics of the input data, the Kalshi predictive signals undergo rigorous stationarity checks. A custom function, `check_stationarity`, is employed to apply the augmented Dickey-Fuller (ADF) test recursively, allowing up to two differencing operations if necessary. If the initial series is non-stationary ($p\text{-value} > 0.05$), differencing is applied until stationarity is achieved. This ensures the validity of methods that assume stationarity (see Algorithm 5 in Appendix 8).

Beyond ensuring stationarity, dynamic predictor selection is performed using Granger causality testing. These tests assess whether each differenced Kalshi signal offers statistically significant predictive information for the return series of the corresponding asset. Predictors that meet a predefined significance threshold (typically set at 5%) are retained for inclusion in the ARIMAX model of the specific company. This process results in a more concise and interpretable set of features tailored to each asset. The complete selection procedure is detailed in Algorithm 6 in Appendix 9, which describes the Granger causality-based filtering approach (Granger, 1969).

To preserve temporal integrity and prevent look-ahead bias, walk-forward validation is applied independently to each asset. Each fold of the validation process respects the chronological order, ensuring that the validation data always succeed in its corresponding training data in time.

This local modeling approach facilitates a more granular understanding of asset-level behavior and improves prediction accuracy in cases where global models might overlook nuanced patterns. Although this methodology is computationally intensive, it offers superior interpretability and robustness, particularly when the relevance of the predictive signal varies between assets.

To benchmark the performance of the ARIMAX models, we also trained the XGBoost models locally for each asset. This allows XGBoost to learn asset-specific dynamics, distinguishing this approach from the previously implemented global XGBoost model.

4.2 Forecast Error Evaluation

For each fold, forecast errors $e_t = y_t - \hat{y}_t$ were calculated for both forecasting approaches. These errors were aggregated to facilitate a fair and robust comparison between the global and local models.

While traditional metrics such as RMSE, MAE, and MSE offer intuitive and easily interpretable measures of predictive performance, they provide only point-wise summaries and do not allow formal inference regarding the relative quality of competing models. In contrast, we adopt a more statistically grounded approach employing the Diebold-Mariano (DM) test (Diebold and Mariano, 1995), which assesses whether the observed differences in forecast accuracy between two models are statistically significant.

By analyzing loss differentials—typically defined as the difference in squared or absolute forecast errors—across folds, the DM test determines whether one model consistently outperforms the other beyond what would be expected due to random variation. This approach allows us to move beyond descriptive error measures and towards inferential statistics, offering a deeper understanding of which modeling strategy provides superior predictive performance in a statistically meaningful way.

5 Analysis of Results

This section presents a comprehensive analysis of our empirical findings, organized around three main aspects: global modeling results, local modeling outcomes, and cross-strategy performance comparisons. We begin by examining the performance of global models, followed by an analysis of local approaches, and conclude with a comparative assessment of different modeling strategies.

5.1 Results from Global Modeling

This section presents the results of global modeling applied across all stocks. We compare two models, analyze their performance, examine the XGBoost learning curve, and identify the most important features used by this model.

5.1.1 Global Models performance

In the global approach, two modeling techniques were applied: a traditional linear regression model and a gradient-boosted tree model (XGBoost). The objective here is to evaluate which of these models offers superior predictive accuracy when applied in a unified cross-asset framework.

The comparison between models was conducted over two folds using root mean squared error (RMSE) as the primary evaluation metric. As shown in Table 5, both models produced RMSE values, and linear regression often produced lower errors than XGBoost. This suggests that, in terms of raw predictive performance, the simplicity of linear regression does not necessarily come at a cost in accuracy compared to more complex methods.

To assess whether these differences were statistically significant, we employed the Diebold-Mariano test. The results reveal that in 18 of 40 cases, linear regression significantly outperformed XGBoost at the 5% significance level. In the remaining cases, no statistically significant differences were detected, indicating that the added complexity of XGBoost does not consistently yield better performance in this context (see Table 6 in Appendix 11).

These findings imply that, under current conditions, characterized by limited prediction market data and a relatively short time horizon in early 2024, XGBoost may not be able to fully exploit its potential. Consequently, for Delphia, a simpler linear regression model may be preferable in global modeling scenarios. It offers comparable predictive power without the risk of overfitting or unnecessary computational complexity, making it a more pragmatic and efficient choice in

the current data environment.

5.1.2 Learning Curve

Figures 1 and 2 present the progressive MSE learning curves for two different folds of walk-forward validation. In Figure 1, we observe a clear sign of overfitting: the training error remains very low while the test error fluctuates and stays relatively high, indicating that the model fits the training data well but fails to generalize effectively. This gap suggests the model may be too complex for the amount or quality of training data available in this fold.

In contrast, Figure 2 shows a healthier learning pattern. As the size of the training set increases, the test error decreases and approaches the training error, which itself grows slightly, a desirable behavior in learning curves. This suggests that, for this fold, the model generalizes better and does not suffer from overfitting. The difference between folds indicates that model performance is sensitive to data splits, underlining the importance of robust validation procedures and potential regularization.

5.1.3 Global Model Feature Importance (XGBoost)

Having examined the learning behavior of XGBoost, we now turn to an analysis of the input features on which the model relied the most during training. To better understand the internal decision-making process of the XGBoost model in the global framework, feature importance plots were analyzed for both validation folds. These plots, presented in Figures 3 and 4, highlight which input variables contributed the most significantly to the models predictions.

For fold 1, the most influential feature was `ticker_COIN`, suggesting that asset-specific signals related to the COIN stock were especially informative in the period. Several temporal features such as `date_day`, `date_qday`, and `date_index.num` also ranked highly, indicating that the model leverages temporal structures, such as intra-month and intra-week effects, to enhance predictive accuracy.

In fold 2, the importance of features shifted slightly. While `date_index.num` remained dominant, the presence of multiple asset tickers such as `ticker_TSLA` and `ticker_NVDA` among the top-ranked variables points to a broader cross-sectional learning dynamic. This change suggests that, as more data is introduced in fold 2, the model begins to generalize from patterns observable across multiple assets rather than focusing on one dominant stock.

The presence of asset identifiers and time-based variables among the top features confirms that XGBoost is able to exploit both temporal and cross-sectional structures in the global setup. However, it is important to note that this analysis reveals that the values of exogenous variables are not taken into account, nor are the IDs from prediction markets. Consequently, the features most exploited by the model are primarily the tickers of the target variables. Although these importance rankings do not imply causal relationships, they are useful for identifying the features that drive the model’s predictive power and for guiding future feature engineering efforts.

5.2 Results from Local Modeling

This section presents the internal results derived from the (local) firm-level modeling, serving as a basis for understanding the subsequent conclusions. The analyses focus on data preparation, causal relationships, and model performance at the individual stock level.

5.2.1 Stationarity Assessment

To ensure valid time-series modeling, the Augmented Dickey-Fuller test revealed distinct stationarity patterns across our variables. As shown in Table 7 in Appendix 14, while stock returns were naturally stationary, 12 of 16 prediction market signals required first order differencing to achieve stationarity. The four exceptions Meta daily active users, WTI oil prices, and Apple car development and Google were stationary at the level.

This pattern, clearly documented in Table 7, suggests that prediction markets often reflect cumulative expectations rather than instantaneous shifts, creating persistent trends that require differencing for proper model specification. This preprocessing step was essential to prevent coefficient distortion in subsequent ARIMAX modeling.

5.2.2 Granger Causality Results

Table 8 in Appendix 15 presents the results of 320 pairwise Granger causality tests conducted at the 5% significance level, each corresponding to a prediction pair of market–stock return. Only 16 tests produced p-values below 5%, highlighting a limited number of statistically significant predictive relationships. Overall, macroeconomic signals showed a wider influence than firm-specific indicators. Furthermore, when comparing the RMSE values of the stock predictions using the ARIMA-X models (see Table 9 in Appendix 16), no substantial improvement was observed with the inclusion of exogenous predictors, except for the "GOOG" stock. This suggests

that in most cases, adding external regressors did not significantly improve the accuracy of the forecast.

The GDP prediction market stood out, with Granger-causing returns in six assets (IWDA.AS, XOM, SPY, JPM, QQQ, and COIN), with p-values ranging from 0.0048 to 0.0285.

The bottom portion of Table 8 displays relationships that fall into the marginally significant range ($0.05 < p < 0.10$), including Apple's influence on WMT ($p = 0.0551$) and cryptocurrency predictions on tech stocks.

Notably, prediction markets tied to specific firms, for example Netflix, do not predict their own stock returns. This underlines the value of Granger testing in identifying genuinely influential markets beyond intuitive assumptions.

5.2.3 Local Models Performance

Regarding local modeling approaches, two distinct methods were applied: ARIMA(X) models and XGBoost models in each stock. This section aims to evaluate which of these models statistically outperforms the other when modeling at the firm-specific level.

A comparison based solely on RMSE (Root Mean Squared Error) reveals that ARIMA-X models generally achieve lower prediction errors compared to XGBoost, as evidenced by Table 10 (see Appendix 17).

When extending the evaluation to a statistical test, namely the Diebold-Mariano test, we observe that in 19 out of 40 cases (approximately 47.5%), the ARIMAX model performs significantly better than XGBoost. In the remaining 21 cases, the differences between the two models were not statistically significant, suggesting that ARIMAX or XGBoost could be appropriate for these companies. These findings, summarized in Table 11 in Appendix 18, confirm the statistical superiority of the ARIMAX approach in the majority of cases.

Therefore, a fintech company like Delphia would greatly benefit from adopting ARIMAX models over XGBoost for localized stock-by-stock modeling, given their superior statistical performance in most cases.

5.3 Cross-Strategy Performance Comparison

5.3.1 Comparison of ARIMAX and linear regression

When comparing the performance of ARIMA and Linear Regression (LR) models across both folds, the results suggest that the two approaches yield generally comparable predictive accuracy. As shown in Table 12 (see Appendix 19), ARIMA was identified as the better model in a slight majority of cases in Fold 1, while LR often performs equally well or better in Fold 2.

The RMSE values between ARIMA and LR are typically close, with absolute differences frequently within a narrow margin. One possible explanation for this similarity is the nature of the target variable, daily stock returns, which are generally stationary by construction. Because both ARIMA and LR are well suited to modeling stationary time series, their underlying assumptions and behaviors may lead to convergent forecasts, thereby reducing performance differences. This suggests that in contexts involving stationary return series, simpler models like LR may offer performance levels comparable to more complex time series approaches such as ARIMAX, particularly when exogenous variables offer limited marginal value.

To further validate these observations, we applied the Diebold-Mariano (DM) test to statistically compare the forecast accuracy of ARIMA and LR models across each ticker and fold. The results, presented in Table 13 (see Appendix 20), reveal that in the vast majority of cases, 39 out of 40 comparisons, the test fails to reject the null hypothesis of equal predictive accuracy. This indicates that the differences in RMSE observed earlier are not statistically significant in most instances. Only in a isolated case, for JNJ in Fold 2, did the test find a significant difference favoring the LR model.

These findings reinforce the notion that both models perform similarly when applied to stationary return series. Since stationarity removes trends and seasonality, the remaining structure in the data may be adequately captured by both linear (LR) and time series-based (ARIMA) models. This convergence in model behavior further supports the idea that, under stationary conditions, model selection could be guided more by practical considerations, such as ease of implementation or interpretability, than by differences in forecast accuracy.

5.3.2 Comparison of XGBoost at the local and global level

Our analysis of XGBoost performance across local and global implementations reveals significant differences in their effectiveness. In Table 15 (see Appendix 22), the Diebold-Mariano test results show that the Global XGBoost model significantly outperforms its local counterpart in 10 out of 40 cases (25%). In the remaining 30 cases (75%), no statistically significant differences were detected.

These findings suggest several important insights. The superior performance of the global XGBoost model in cases indicates its effectiveness in capturing cross-asset patterns and shared market dynamics. The performance improvement of the model with larger datasets suggests that it benefits from the increased statistical power provided by the pooled data. The limited success of the Local XGBoost model, particularly in data-rich conditions, suggests that asset-specific patterns may be less dominant than shared market dynamics.

From a practical perspective, these results have important implications for financial forecasting. The Global XGBoost approach offers better scalability and computational efficiency, requiring only one model to be trained and maintained. The model's ability to learn from cross-asset information makes it particularly valuable for diversified portfolios. The results support the use of global modeling approaches when dealing with prediction market signals, as they appear to capture broader market dynamics more effectively.

These findings align with theoretical expectations about the effectiveness of tree-based algorithms in identifying complex patterns. The superior performance of the global approach suggests that prediction market signals may contain information that is more relevant at a market-wide level than at the individual asset level.

5.4 Questions Evaluation

Q1. Local models leverage prediction market signals more effectively than global models: Our analysis provides mixed evidence on the effectiveness of local versus global models in leveraging prediction market signals. The results reveal a nuanced picture that requires careful interpretation.

First, the Granger causality analysis (Table 8) shows that prediction market signals exhibit varying degrees of influence across different assets. Although some firm-specific signals show

significant predictive power for stocks, others demonstrate broader market-wide effects. For instance, GDP predictions Granger cause returns in six different assets, suggesting that some prediction market signals contain information that transcends individual stock boundaries.

Second, feature importance results at the global level indicate that XGBoost tends to prioritize complex interactions between target variables, rather than relying heavily on direct prediction market signals, such as firm identifiers or market-based prediction values. As a result, XGBoost reduces its overall error by exploiting these cross-asset dependencies and implicit patterns, instead of depending on the more straightforward informational content of the prediction market features.

Although local models appear to leverage prediction market signals more directly, particularly those that are asset-specific, global models like XGBoost tend to exploit broader, more complex relationships across assets. This divergence in modeling strategy highlights a trade-off: local models may better capture the immediate informational value of prediction markets for individual assets, whereas global models benefit from learning intricate patterns across the dataset as a whole. These findings suggest that the effectiveness of prediction market signals depends not only on the quality of the signals themselves but also on the modeling framework used to extract value from them.

Q2. Prediction market influence differs between individual stocks and ETFs: Our analysis provides strong evidence that prediction market signals influence individual stocks and ETFs differently. This differential impact manifests in several distinct ways that have important implications for financial forecasting and portfolio management.

The Granger causality tests reveal distinct patterns of influence between individual stocks and ETFs. Although individual stocks show significant responses to both firm-specific and macroeconomic prediction market signals, ETFs demonstrate a more selective response, primarily to broader market indicators. For instance, the GDP prediction market shows a significant Granger causality with SPY, IWDA.AS, QQQ and COIN (p-values ranging from 0.0048 to 0.0285), but fewer significant relationships with individual stocks. This pattern suggests that ETFs are more sensitive to market-wide prediction signals, while individual stocks respond to a broader range of both firm-specific and macroeconomic signals.

The forecasting performance metrics show systematic differences between individual stocks and

ETFs. As documented in Table 12, ETFs (SPY, IWDA.AS, QQQ) generally achieve lower RMSE values compared to individual stocks, suggesting that prediction market signals may be more effectively captured in broader market indices. This finding aligns with the hypothesis that ETFs, by their nature, aggregate and potentially amplify market-wide signals while dampening firm-specific noise. The more stable performance of ETFs in response to prediction market signals suggests that these instruments may be more suitable for strategies based on broad market predictions.

These findings have significant implications for investment strategy and portfolio management. ETFs appear more suitable for strategies based on broad market prediction signals, while individual stocks may offer better opportunities for strategies leveraging firm-specific prediction market signals. The choice between individual stocks and ETFs should consider the nature of the prediction market signals being utilized, with broader market signals being more effective for ETF strategies and firm-specific signals being more relevant for individual stock selection.

6 Discussion

6.1 Methodological Insights

At the global prediction level, features related to ticker identifiers consistently exhibit significantly greater importance than exogenous variables sourced from prediction markets or their identifiers. Indeed, while the XGBoost model demonstrates learning capacity overall, it fails to identify robust explanatory variables that can effectively model the target stock returns. As a result, XGBoost predominantly relies on temporal features and stock identifiers, suggesting that prediction markets do not serve as reliable indicators of stock price movements in this context.

In contrast, local predictions utilizing ARIMA(X) models reveal a more nuanced picture. Only a limited subset of prediction markets demonstrates Granger causality with respect to stock returns. Moreover, prediction markets that one might intuitively expect to be related to specific stocks, for example, the market forecasting Netflix and Netflix stock returns, do not exhibit significant Granger causal relationships. Surprisingly, a prediction market related to measles displayed a stronger Granger causal link with Netflix stock returns. In many cases, relying on local ARIMA models without introducing exogenous variables proved preferable, as these variables did not enhance predictive accuracy.

Furthermore, Granger causality tests indicate that global prediction markets tend to exert broader influences on stock returns. Specifically, the three ETFs examined were Granger-caused by a global prediction market, whereas individual stocks were more influenced by specialized prediction markets. This suggests that local models are more effective at harnessing prediction market data for modeling stock returns, while global models tend to dilute this information, favoring endogenous over exogenous variables.

It is also important to emphasize that data quality substantially affected the outcomes. Several prediction markets suffered from intermittent maintenance, resulting in data gaps. These gaps limited the ability of preprocessing steps to accurately capture the potential impact of prediction markets on stock returns.

6.2 Limitations and Challenges

This study faced several significant challenges. First, prediction market data were sparse and difficult to obtain. Additionally, the data availability was often limited to discrete intervals, quarterly, monthly, or annually, complicating preprocessing and temporal alignment with stock returns. Once events concluded, associated prediction markets typically became inactive, further limiting data continuity. Moreover, prediction markets rely entirely on user participation; without sufficient betting activity, data quality and quantity suffer.

Second, prediction markets remain relatively underdeveloped and constrained. While their use is widespread in the United States, as evidenced during the 2024 presidential election, their adoption is limited or even restricted in other regions. For example, the Polymarket platform was shut down in Belgium in 2025 due to its classification as a gambling site, and additional restrictions were imposed on the Klashi platform shortly thereafter.

Finally, while the primary objective of this thesis was to analyze the impact of prediction markets on stock returns, it is worth noting that the reverse relationship, how stock market performance may influence prediction market outcomes, represents a promising avenue for future research.

6.3 Practical Implications for Investors

The results of this study carry important implications for practitioners in the financial industry, including Delphia. While prediction markets are often touted as valuable tools for aggregating information and anticipating future events, our findings suggest a more nuanced perspective is

warranted.

First, investors and asset managers should exercise caution in relying solely on prediction market data for investment decisions. Both global and local modeling approaches show that prediction markets rarely emerge as key explanatory variables at the aggregate level. Furthermore, Granger causality tests indicate that these markets have limited predictive power over individual stock returns. This suggests that, from a financial standpoint, prediction markets alone are insufficient as standalone forecasting tools.

However, this does not entirely discount their potential value. Prediction markets can serve as complementary indicators when combined with traditional financial variables and models. For instance, local models demonstrate somewhat better integration of prediction market signals, hinting at their utility in niche or event-specific forecasting scenarios. Fintech firms could leverage this insight by incorporating prediction market data selectively, for example, around earnings announcements, regulatory decisions, or macroeconomic events where user-driven sentiment may provide incremental information.

From a strategic viewpoint, investment funds might consider using prediction market information as an additional input to risk management or portfolio allocation frameworks, rather than as primary trade signals. The presence of data quality issues and regulatory constraints, as observed in this study, further reinforces the need for cautious interpretation and robust preprocessing pipelines.

7 Conclusion

This thesis has explored the relationship between prediction market signals and stock returns through a comparative modeling approach. The results obtained provide important insights into the use of prediction markets in the context of financial forecasting.

The comparative analysis between global and local approaches reveals significant differences in how prediction market signals influence returns. Local models, particularly ARIMAX models, have demonstrated a superior ability to capture asset-specific relationships, while global models have shown better performance in detecting shared market dynamics. This distinction highlights the importance of selecting the appropriate modeling approach depending on the application context.

Granger causality tests showed that certain prediction markets, notably the one related to GDP, exert a broader influence on ETF returns than on individual stocks. Conversely, the measles prediction market exerts a more specific influence, particularly on Netflix. This finding suggests that prediction markets can be effective both for anticipating overall market movements and for predicting returns of specific stocks.

The performance comparison between XGBoost models and traditional statistical models reveals that increased complexity in machine learning models does not necessarily guarantee better predictive accuracy. In fact, simpler models, such as linear regression and ARIMA, often achieved superior results, underscoring the importance of parsimony in financial modeling.

The practical implications of these results are significant for fintech companies like Delphia. The integration of prediction market signals into investment strategies should be done selectively and contextually, taking into account the nature of the target asset and the type of signal used. ETFs appear to be more suitable instruments for strategies based on global market predictions, whereas individual stocks may offer better opportunities for strategies exploiting company-specific signals.

This research contributes to the existing literature by demonstrating that the informational value of prediction markets depends not only on the quality of the signals themselves but also on how they are integrated into forecasting models. The results suggest that prediction markets may be more effective as complementary indicators rather than as standalone forecasting tools.

Finally, the identified limitations, particularly regarding data quality and availability, emphasize the need for a cautious approach when using prediction markets for financial decision-making. Future research could explore the reverse impact, how stock market performance influences prediction market outcomes, as well as the integration of these signals with other alternative data sources to improve predictive accuracy.

Bibliography

- Aroussi, R. (2025). Yfinance: Yahoo! finance market data downloader.
- Baltagi, B. H. (2008). *Econometric analysis of panel data* (4th ed.). Wiley.
- Bandara, K., Bergmeir, C., & Smyl, S. (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, 140.
- Berg, J. E., Forsythe, R., Nelson, F. D., & Rietz, T. A. (2008). Results from a dozen years of election futures markets research. In W. Güth & R. Kölle (Eds.), *Handbook of experimental economic methodology* (pp. 217–242). North-Holland.
- Berg, J. E., Nelson, F. D., & Rietz, T. A. (2008). Prediction market accuracy in the long run. *International Journal of Forecasting*, 24(2), 285–300. <https://doi.org/10.1016/j.ijforecast.2008.03.007>
- Berg, J. E., Neumann, G. R., & Rietz, T. A. (2009). Searching for Google’s Value: Using Prediction Markets to Forecast Market Capitalization Prior to an Initial Public Offering [Publisher: INFORMS]. *Management Science*, 55(3), 348–361. Retrieved January 14, 2025, from <https://www.jstor.org.proxy.bib.uclouvain.be:8888/stable/40539152>
- Berg, J. E., & Rietz, T. A. (2003). Results from a dozen years of election futures markets research [Working paper from Iowa Electronic Markets]. *Handbook of Experimental Economic Results*.
- Bontempi, G., Ben Taieb, S., & Le Borgne, Y.-A. (2013, January). Machine learning strategies for time series forecasting. https://doi.org/10.1007/978-3-642-36318-4_3
- Bossaerts, F., Yadav, N., Bossaerts, P., Nash, C., Todd, T., Rudolf, T., Hutchins, R., Ponsonby, A.-L., & Mattingly, K. (2022, September). Price Formation in Field

- Prediction Markets: The Wisdom in the Crowd [arXiv:2209.08778 [q-fin]]. <https://doi.org/10.48550/arXiv.2209.08778>
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control* (5th). John Wiley & Sons.
- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Croissant, Y., & Millo, G. (2024). Plm: Linear models for panel data [<https://cran.r-project.org/package=plm>].
- Dai, M., Jia, Y., & Kou, S. (2021). The wisdom of the crowd and prediction markets. *Journal of Econometrics*, 222(1), 561–578. <https://doi.org/10.1016/j.jeconom.2020.07.016>
- Dancho, M. (2021). Modeling panel data [Accessed: 2025-05-13].
- Dancho, M. (2024a). Modeltime: Time series forecasting for business [<https://business-science.github.io/modeltime/>].
- Dancho, M. (2024b). Timetk: Time series tool kit for analysis and forecasting [<https://business-science.github.io/timetk/>].
- Delphia Technologies Inc. (2025). Delphia [Consulté le 21 mai 2025]. <https://delphia.com>
- Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3), 253–263. <https://doi.org/10.1080/07350015.1995.10524599>
- Durbin, J., & Koopman, S. J. (2012). *Time series analysis by state space methods* (2nd edition). Oxford University Press.
- Feng, G., He, J., & Polson, N. G. (2018). Deep learning for predicting asset returns. *arXiv preprint arXiv:1804.09314*. <https://arxiv.org/abs/1804.09314>
- Feng, W., et al. (2023). High-frequency intraday data for linear regression estimation: An application to capm. *AIMS Mathematics*, 8(3), 3401–3425. <https://doi.org/10.3934/math.2023662>
- Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669. <https://doi.org/10.1016/j.ejor.2017.11.054>

- Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, 37(3), 424–438.
- Grolemund, G., & Wickham, H. (2024). Lubridate: Make dealing with dates a little easier [<https://lubridate.tidyverse.org>].
- Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5), 2223–2273. <https://doi.org/10.1093/rfs/hhaa009>
- Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning in finance. *Annual Review of Financial Economics*, 11.
- Helske, J. (2024). Kfas: Kalman filter and smoother for exponential family state space models [<https://cran.r-project.org/package=KFAS>].
- Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1), 388–427.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd) [Available at: <https://otexts.com/fpp2/>]. OTexts.
- Hyndman, R. J., & Athanasopoulos, G. (2024). *Forecast: Forecasting functions for time series and linear models* [<https://pkg.robjhyndman.com/forecast/>].
- Januschowski, T., Lewandowski, P., Wever, M., et al. (2022). Forecasting with global models of multivariate time series [Accessed: 2025-05-18]. arXiv preprint arXiv:2212.03523. <https://arxiv.org/abs/2212.03523>
- Kalshi Inc. (2025). Kalshi - prediction market for trading the future [Consulté le 18 mai 2025]. <https://kalshi.com>
- Kelly, B., Pruitt, S., & Su, Y. (2019). Market expectations in the cross section of present values. *Journal of Finance*, 74(3), 1253–1295.
- Krauss, C., Do, X. A., & Huck, N. (2016). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500 (FAU Discussion Papers in Economics No. 03/2016). Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics. <https://hdl.handle.net/10419/130166>

- Kuhn, M. (2024). Caret: Classification and regression training [<https://topepo.github.io/caret/>].
- Kuhn, M., & Wickham, H. (2024). Tidymodels: A collection of modeling packages [<https://www.tidymodels.org>].
- Laga, H., & Moustafa, M. (2024). Metrics: Evaluation metrics for machine learning [<https://cran.r-project.org/package=Metrics>].
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Li, Q., Wang, T., Li, P., Liu, L., Gong, Q., & Chen, Y. (2014). The effect of news and public mood on stock movements. *Information Sciences*, 278, 826–840. <https://doi.org/10.1016/j.ins.2014.03.096>
- Lopez de Prado, M. (2018). *Advances in financial machine learning*. John Wiley & Sons.
- Mongrain, P., & Stegmaier, M. (2024). Introduction to Forecasting the 2024 US Elections. *PS: Political Science & Politics*, 1–12. <https://doi.org/10.1017/S1049096524001008>
- Montero-Manso, P., & Hyndman, R. J. (2021). Principles and algorithms for forecasting groups of time series: Locality and globality. <https://arxiv.org/abs/2008.00444>
- Montgomery, D. C. (2012). *Introduction to linear regression analysis*. Wiley.
- Petropoulos, F., Kourentzes, N., Fildes, R., & Hyndman, R. J. (2022). Forecasting: Theory and practice. *Foresight: The International Journal of Applied Forecasting*, 65, 5–13.
- Pfaff, B. (2024). Urca: Unit root and cointegration tests for time series data [<https://cran.r-project.org/package=urca>].
- Sirignano, J., & Cont, R. (2019). Universal features of price formation in financial markets: Perspectives from deep learning. *Quantitative Finance*, 19(9), 1449–1459.
- Surowiecki, J. (2004). *The wisdom of crowds*. Doubleday.
- Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: An analysis and review. *International journal of forecasting*, 16(4), 437–450.
- Trapletti, A., & Hornik, K. (2024). Tseries: Time series analysis and computational finance [<https://cran.r-project.org/package=tseries>].
- Vaswani, A., Shazeer, N., & Parmar, N. e. a. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*.

- Waitz, M., & Mild, A. (2013). Corporate prediction markets: A tool for predicting market shares. *Journal of Business Economics*, 83(3), 193–212. <https://doi.org/10.1007/s11573-012-0645-1>
- Wickham, H. (2024a). Ggplot2: Create elegant data visualisations using the grammar of graphics [<https://ggplot2.tidyverse.org>].
- Wickham, H. (2024b). Reshape2: Flexibly reshape data: A reboot of the reshape package [<https://cran.r-project.org/package=reshape2>].
- Wickham, H. (2024c). Tidyr: Tidy messy data [<https://tidyr.tidyverse.org>].
- Wickham, H. (2024d). Tidyverse: Easily install and load the 'tidyverse' [<https://www.tidyverse.org>].
- Wickham, H., & François, R. (2024). Dplyr: A grammar of data manipulation [<https://dplyr.tidyverse.org>].
- Wickham, H., & Henry, L. (2024). Purrr: Functional programming tools [<https://purrr.tidyverse.org>].
- Wolfers, J., & Zitzewitz, E. (2004). Prediction Markets. *Journal of Economic Perspectives*, 18(2), 107–126. <https://doi.org/10.1257/0895330041371321>
- Wollmann, T. (2024). Moments: Moments, cumulants, skewness, kurtosis and related tests [<https://cran.r-project.org/package=moments>].
- Wooldridge, J. M. (2009). *Introductory econometrics: A modern approach* (4th ed.). South-Western. <http://books.google.ch/books?id=64vt5TDBNLwC>
- Yahoo Finance. (2025). Yahoo [Financial data service, 2025-05-20].
- Zeileis, A., & Grothendieck, G. (2024). Zoo: S3 infrastructure for regular and irregular time series [<https://cran.r-project.org/package=zoo>].
- Zeileis, A., & Hothorn, T. (2024). Lmtest: Testing linear regression models [<https://cran.r-project.org/package=lmtest>].
- Zeileis, A., Leisch, F., Hornik, K., & Kleiber, C. (2024). Strucchange: Testing, monitoring, and dating structural changes [<https://cran.r-project.org/package=strucchange>].
- Zhu, H. (2024). Kableextra: Construct complex table with 'kable' and pipe syntax [<https://haozhu233.github.io/kableExtra/>].
- Zivanovic, R. (2001). Local regression-based short-term load forecasting. *Journal of Intelligent and Robotic Systems*, 31, 115–127. <https://doi.org/10.1023/A:1012094702855>

8 Appendix

Appendix 1

Market	Number of Observations (daily)
Apple car	429
Bitcoin (BTC)	231
Ethereum (ETH)	229
Federal Reserve (Fed)	349
U.S. Gas Prices	272
GDP Growth	292
Google CEO	305
Hurricane Counts	240
Inflation	303
Tech Layoffs	306
Measles Incidence	293
Meta DAU	223
Netflix Subs	287
SpaceX Launches	223
Tesla Production	307
WTI Oil	284

Table 2: Prediction markets and corresponding number of daily observations in 2024.

Appendix 2

Ticker	Company Name
TSLA	Tesla, Inc.
NFLX	Netflix, Inc.
META	Meta Platforms, Inc.
GOOG	Alphabet Inc. (Class C)
COIN	Coinbase Global, Inc.
INTC	Intel Corporation
JPM	JPMorgan Chase & Co.
XOM	Exxon Mobil Corporation
AAPL	Apple Inc.
BAC	Bank of America Corporation
TTE	TotalEnergies SE
PFE	Pfizer Inc.
JNJ	Johnson & Johnson
MSFT	Microsoft Corporation
AMZN	Amazon.com, Inc.
WMT	Walmart Inc.
NVDA	NVIDIA Corporation

Table 3: Selected U.S. equities with their tickers and company names

Appendix 3

Package	Reference
tidymodels	Kuhn and Wickham, 2024
modeltime	Dancho, 2024a
timetk	Dancho, 2024b
tidyverse	Wickham, 2024d
lubridate	Grolemund and Wickham, 2024
tseries	Trapletti and Hornik, 2024
ggplot2	Wickham, 2024a
caret	Kuhn, 2024
dplyr	Wickham and François, 2024
plm	Croissant and Millo, 2024
tidyr	Wickham, 2024c
zoo	Zeileis and Grothendieck, 2024
forecast	Hyndman and Athanasopoulos, 2024
KFAS	Helske, 2024
reshape2	Wickham, 2024b
lmtree	Zeileis and Hothorn, 2024
purrr	Wickham and Henry, 2024
urca	Pfaff, 2024
Metrics	Laga and Moustafa, 2024
strucchange	Zeileis et al., 2024
moments	Wollmann, 2024
kableExtra	Zhu, 2024

Table 4: List of R packages used in the analysis with corresponding bibliographic citations

Appendix 4

Algorithm 1 Process Daily Data

```

1: function ProcessDailyData(df, ticker)
2:   Convert timestamps in df to dates
3:   Add a new column id to df with value ticker
4:   Group df by date and id
5:   Compute pred_daily as the mean of Value
6:   return a summarized data frame with columns: date, id, and pred_daily
7: end function

```

Appendix 5

Algorithm 2 Apply Kalman Filter to Daily Predictions

```

1: function ApplyKalmanFilter(df)
2:   Convert column pred_daily in df to a time series ts_data
   with frequency 365, starting at year 2024, day 1
3:   Define a state space model model as a local level model without intercept:
   ts_data ~ -1 + SSMtrend(degree = 1, Q = 1)
4:   Fit the Kalman filter using KFS(model), store result in kalman_fit
5:   Extract filtered state estimates: kalman_values ← kalman_fit.a
6:   if length(kalman_values) ≠ number of rows in df then
7:     Truncate kalman_values to match number of rows in df
8:   end if
9:   Replace missing values in df.pred_daily with corresponding kalman_values
10:  Replace missing values in df.id with the first non-missing id in df
11:  return updated data frame df
12: end function

```

Appendix 6

Algorithm 3 Create Walk-Forward Time Series Splits

```

1: function CreateWalkForwardSplits(data_tbl)
2:   Sort data_tbl by date in ascending order
3:   Set parameters:
   initial ← 120 days
   assess ← 40 days
   skip ← 40 days
   cumulative ← True
   slice_limit ← 2
4:   Apply time_series_cv() on data_tbl using the above parameters
5:   return time series resamples (2 folds)
6: end function

```

Appendix 7

Algorithm 4 Walk-Forward Training and Evaluation Using XGBoost

- 1: Input: Training dataset `train_data`, test dataset `test_data`
 - 2: Goal: Train an XGBoost model on training fold and evaluate it on test fold
 - 3: Step 1: Define recipe for feature engineering
 - 4: `rec_obj_xgb` $\leftarrow$ Create recipe using value as target and all other variables as predictors
 - 5: Add time series signature based on date
 - 6: Remove date column
 - 7: Remove predictors with zero variance
 - 8: Convert categorical predictors to one-hot encoded format
 - 9: Step 2: Create and train workflow
 - 10: `wflw_xgb` $\leftarrow$ Initialize workflow
 - 11: Add XGBoost regression model to workflow
 - 12: Add feature engineering recipe to workflow
 - 13: Fit workflow on `train_data`
 - 14: Step 3: Store the trained model
 - 15: `model_tbl` $\leftarrow$ Store the trained model in a model table
 - 16: Step 4: Calibrate on test data
 - 17: `calib_tbl` $\leftarrow$ Calibrate model on `test_data` using `id = "ticker"`
 - 18: Step 5: Evaluate accuracy by ticker
 - 19: Compute accuracy metrics grouped by ticker using `calib_tbl`
-

Appendix 8

Algorithm 5 Stationarity Test with Successive Differencing

- 1: function `CheckStationarity(df, ticker)`
 - 2: `df_clean` $\leftarrow$ rows in `df` where returns is not missing
 - 3: `adf_result` $\leftarrow$ Perform ADF test on `df_clean`.returns (null hypothesis: non-stationary)
 - 4: if `adf_result.p_value` $>$ 0.05 then
 - 5: Difference `df`.returns once
 - 6: `df_clean` $\leftarrow$ rows in `df` where returns is not missing
 - 7: `adf_result_diff1` $\leftarrow$ Perform ADF test on `df_clean`.returns
 - 8: if `adf_result_diff1.p_value` $>$ 0.05 then
 - 9: Difference `df`.returns a second time
 - 10: end if
 - 11: end if
 - 12: return `df`
 - 13: end function
-

Appendix 9

Algorithm 6 Granger Causality Test Between Prediction and Stock Return Series

```

1: function GrangerCausalityTest(pred_series, stock_series, max_lag = 5)
2:   Join pred_series and stock_series on the date column
3:   Remove rows with missing values in either pred_daily_pred or pred_daily_stock
4:   Perform Granger causality test:
       test result  $\leftarrow$  grangertest(pred_daily_stock  $\sim$  pred_daily_pred, order =
       max_lag)
5:   return test result
6: end function

```

Appendix 10

Ticker	XGBoost RMSE (F1)	LR RMSE (F1)	Best RMSE (F1)	XGBoost RMSE (F2)	LR RMSE (F2)	Best RMSE (F2)
AAPL	0.0152	0.0087	LR	0.0102	0.0082	LR
AMZN	0.0171	0.0139	LR	0.0102	0.0091	LR
BAC	0.0155	0.0107	LR	0.0132	0.0121	LR
COIN	0.0519	0.0346	LR	0.0346	0.0341	LR
GOOG	0.0198	0.0139	LR	0.0122	0.0118	LR
INTC	0.0487	0.0279	LR	0.0323	0.0175	LR
IWDA.AS	0.0130	0.0081	LR	0.0053	0.0040	LR
JNJ	0.0120	0.0065	LR	0.0116	0.0089	LR
JPM	0.0185	0.0145	LR	0.0116	0.0103	LR
META	0.0215	0.0125	LR	0.0135	0.0122	LR
MSFT	0.0157	0.0098	LR	0.0106	0.0092	LR
NFLX	0.0173	0.0140	LR	0.0145	0.0140	LR
NVDA	0.0352	0.0296	LR	0.0195	0.0208	XGBoost
PFE	0.0145	0.0099	LR	0.0118	0.0115	LR
QQQ	0.0160	0.0133	LR	0.0084	0.0077	LR
SPY	0.0123	0.0088	LR	0.0061	0.0052	LR
TSLA	0.0424	0.0315	LR	0.0270	0.0212	LR
TTE	0.0138	0.0067	LR	0.0085	0.0075	LR
WMT	0.0120	0.0084	LR	0.0092	0.0090	LR
XOM	0.0169	0.0132	LR	0.0123	0.0117	LR

Table 5: RMSEs of XGBoost and Linear Regression for two folds, with the best model indicated for each fold.

Appendix 11

Ticker	Fold	DM Statistic	P-value	Conclusion
AAPL	Fold 1	3.1579	0.0040	Linear Regression
AAPL	Fold 2	2.9619	0.0063	Linear Regression
AMZN	Fold 1	1.5369	0.1364	No difference
AMZN	Fold 2	1.3033	0.2035	No difference
BAC	Fold 1	2.6179	0.0146	Linear Regression
BAC	Fold 2	1.9002	0.0681	No difference
COIN	Fold 1	2.8093	0.0093	Linear Regression
COIN	Fold 2	0.2201	0.8275	No difference
GOOG	Fold 1	3.4849	0.0018	Linear Regression
GOOG	Fold 2	0.7785	0.4431	No difference
INTC	Fold 1	3.1743	0.0038	Linear Regression
INTC	Fold 2	3.2483	0.0031	Linear Regression
IWDA.AS	Fold 1	2.7685	0.0101	Linear Regression
IWDA.AS	Fold 2	2.2626	0.0319	Linear Regression
JNJ	Fold 1	3.0395	0.0053	Linear Regression
JNJ	Fold 2	1.8669	0.0728	No difference
JPM	Fold 1	2.6479	0.0136	Linear Regression
JPM	Fold 2	1.2345	0.2276	No difference
META	Fold 1	2.1021	0.0454	Linear Regression
META	Fold 2	0.8180	0.4205	No difference
MSFT	Fold 1	3.1097	0.0045	Linear Regression
MSFT	Fold 2	2.3344	0.0273	Linear Regression
NFLX	Fold 1	1.8333	0.0782	No difference
NFLX	Fold 2	0.5150	0.6107	No difference
NVDA	Fold 1	1.9298	0.0646	No difference
NVDA	Fold 2	-0.9335	0.3588	No difference
PFE	Fold 1	2.2366	0.0341	Linear Regression
PFE	Fold 2	0.4433	0.6611	No difference
QQQ	Fold 1	1.3731	0.1815	No difference
QQQ	Fold 2	1.5779	0.1262	No difference
SPY	Fold 1	1.8875	0.0703	No difference
SPY	Fold 2	2.0476	0.0504	No difference
TSLA	Fold 1	2.3340	0.0276	Linear Regression
TSLA	Fold 2	1.8181	0.0802	No difference
TTE	Fold 1	4.5207	0.0001	Linear Regression
TTE	Fold 2	1.5080	0.1432	No difference
WMT	Fold 1	1.8429	0.0768	No difference
WMT	Fold 2	0.4040	0.6894	No difference
XOM	Fold 1	2.2797	0.0311	Linear Regression
XOM	Fold 2	0.7911	0.4358	No difference

Table 6: Diebold-Mariano test results.

Appendix 12

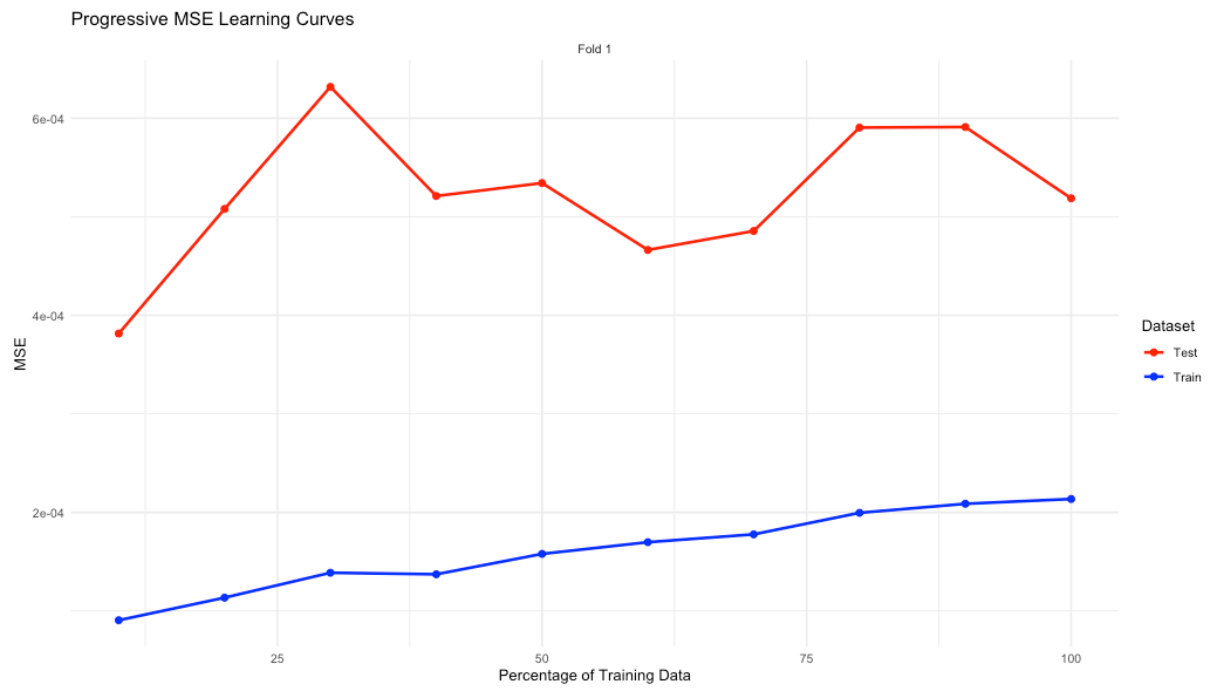


Figure 1: XGBoost learning curve for fold 1.

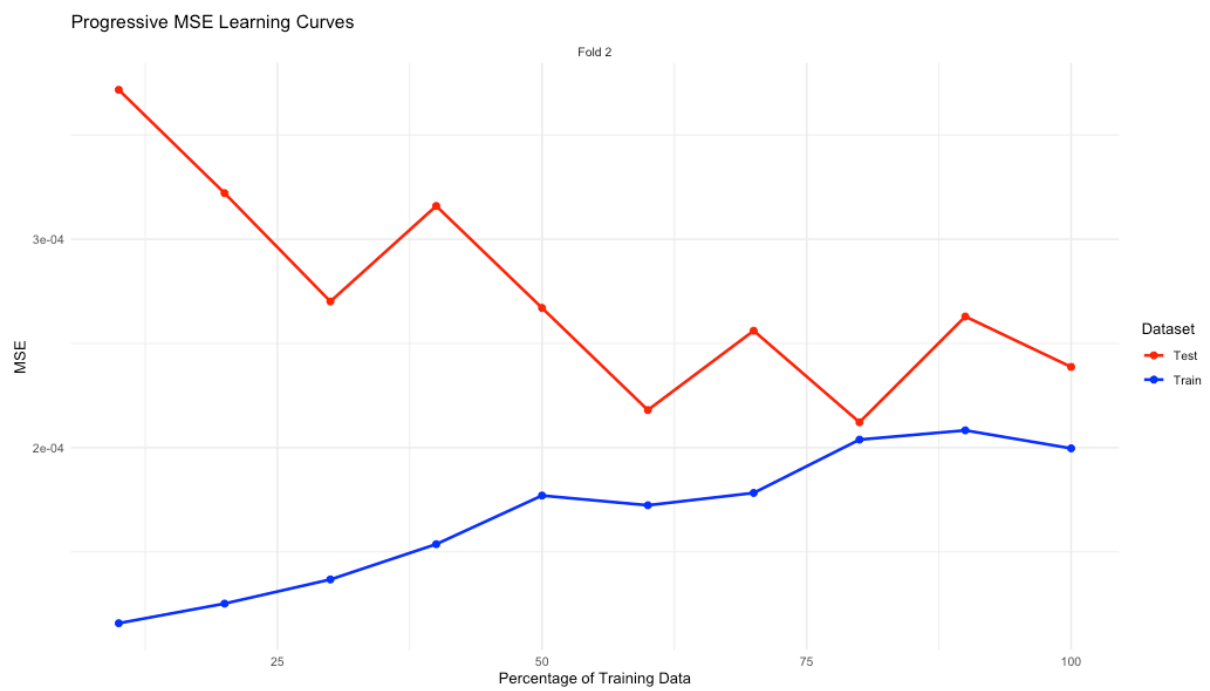


Figure 2: XGBoost learning curve for fold 2.

Appendix 13

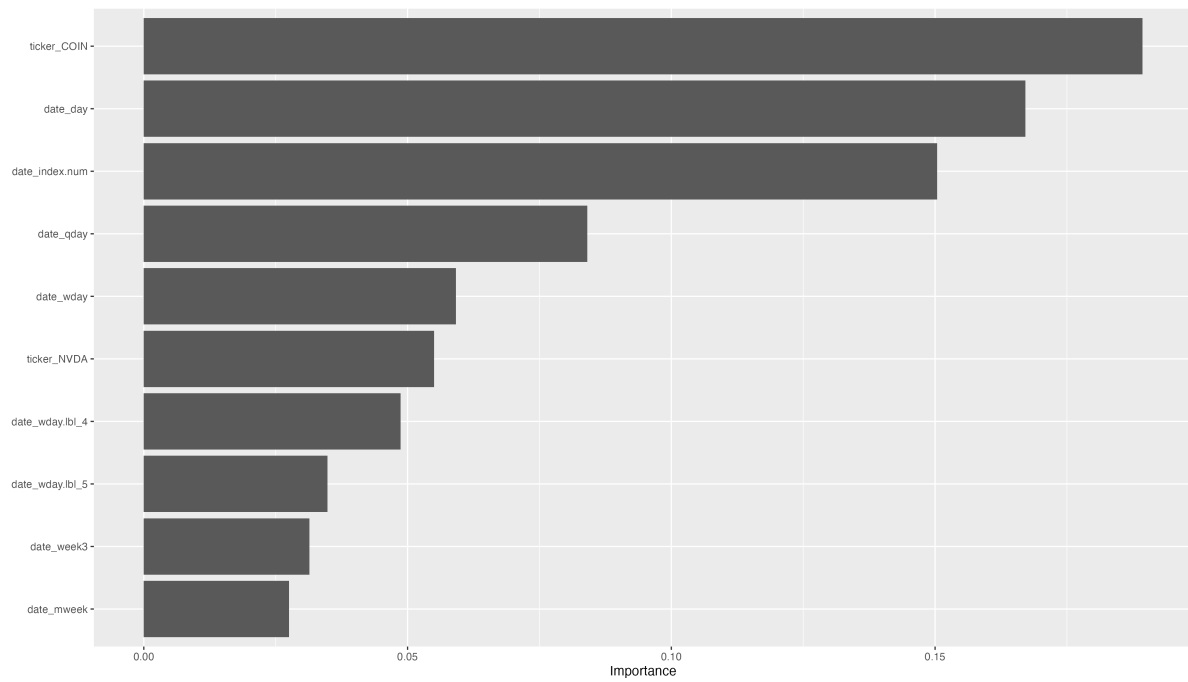


Figure 3: Feature importance for XGBoost model trained on fold 1.

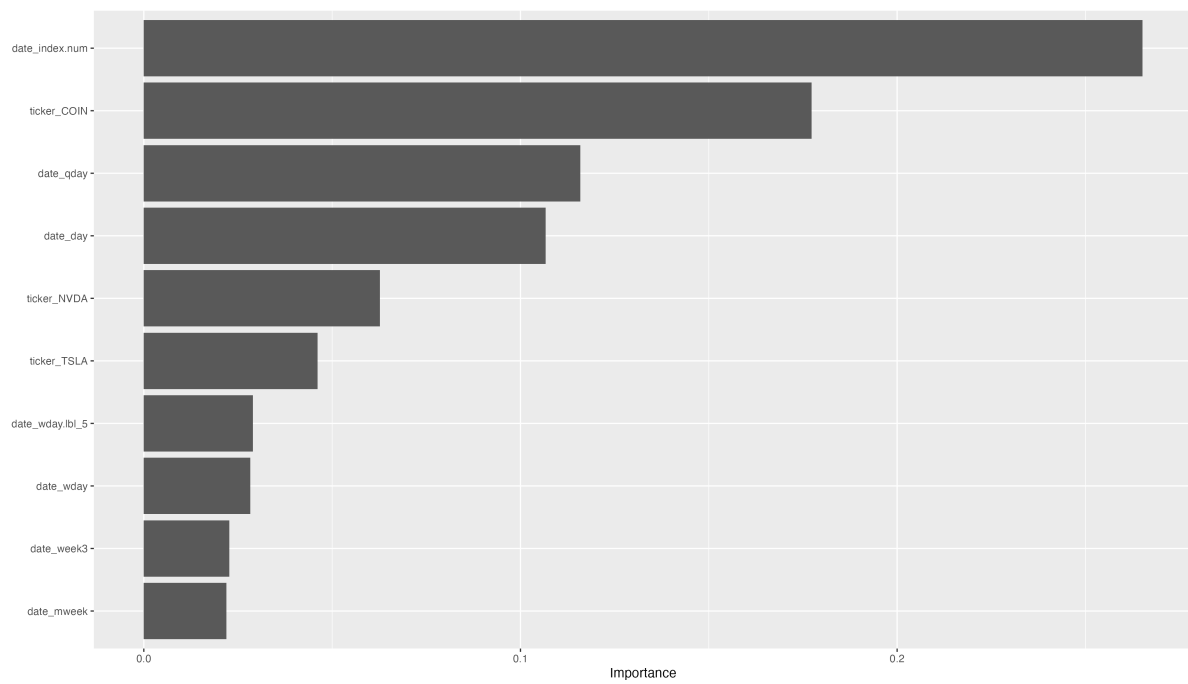


Figure 4: Feature importance for XGBoost model trained on fold 2.

Appendix 14

Pred market	Number of differentiation
TESLA	1
NETFLIX	1
GDP	1
SpaceX	1
Gas US	1
BTC	1
Inflation	1
Hurricanes	1
ETH	1
Measles	1
Fed	1
Layoffs	1
META	0
WTI Oil	0
Apple	0
Google	0

Table 7: ADF test results and required differencing for prediction signals.

Appendix 15

Pred Market	Result	Stock	p-value
GDP	Granger-causes	IWDA.AS	0.0048
Netflix	Granger-causes	PFE	0.0049
Meta	Granger-causes	JNJ	0.0061
GDP	Granger-causes	XOM	0.0073
Hurricanes	Granger-causes	GOOG	0.0081
Fed Rate	Granger-causes	JNJ	0.0088
GDP	Granger-causes	SPY	0.0090
Google	Granger-causes	QQQ	0.0181
GDP	Granger-causes	JPM	0.0200
GDP	Granger-causes	QQQ	0.0202
Measles	Granger-causes	NFLX	0.0203
Fed Rate	Granger-causes	IWDA.AS	0.0208
GDP	Granger-causes	COIN	0.0285
Google	Granger-causes	NFLX	0.0350
Google	Granger-causes	SPY	0.0378
Hurricanes	Granger-causes	TSLA	0.0485
Fed Rate	Does not Granger-cause	TSLA	0.0518
Meta	Does not Granger-cause	PFE	0.0530
Apple	Does not Granger-cause	WMT	0.0551
ETH	Does not Granger-cause	MSFT	0.0574
Google	Does not Granger-cause	MSFT	0.0606
Tesla	Does not Granger-cause	TSLA	0.0630
GDP	Does not Granger-cause	META	0.0746
GDP	Does not Granger-cause	INTC	0.0803
SpaceX	Does not Granger-cause	TSLA	0.0867
BTC	Does not Granger-cause	SPY	0.0872

Table 8: Granger-causal links between prediction markets and asset returns.

Appendix 16

Stock	ARIMA	ARIMAX	Best RMSE (F1)	ARIMA	ARIMAX	Best RMSE (F2)
IWDA.AS	0.0072	0.0073	ARIMA	0.0097	0.0097	Equal
SPY	0.0069	0.0070	ARIMA	0.0107	0.0107	Equal
JNJ	0.0088	0.0093	ARIMA	0.0103	0.0102	ARIMA-X
XOM	0.0116	0.0116	Equal	0.0122	0.0123	ARIMA
PFE	0.0131	0.0130	ARIMA-X	0.0116	0.0117	ARIMA
QQQ	0.0097	0.0096	ARIMA-X	0.0158	0.0158	Equal
GOOG	0.0124	0.0138	ARIMA	0.0146	0.0147	ARIMA
JPM	0.0130	0.0129	ARIMA-X	0.0142	0.0140	ARIMA-X
NFLX	0.0148	0.0148	Equal	0.0142	0.0145	ARIMA
TSLA	0.0247	0.0244	ARIMA-X	0.0356	0.0357	ARIMA
COIN	0.0492	0.0485	ARIMA-X	0.0476	0.0477	ARIMA

Table 9: Comparison of ARIMA and ARIMA-X RMSEs for Fold 1 and Fold 2

Appendix 17

Stock	ARIMA-X RMSE (F1)	XGB RMSE (F1)	Best RMSE (F1)	ARIMA-X RMSE (F2)	XGB RMSE (F2)	Best RMSE (F2)
AAPL	0.0082	0.0125	ARIMA(X)	0.0086	0.0179	ARIMA(X)
AMZN	0.0090	0.0159	ARIMA(X)	0.0142	0.0153	ARIMA(X)
BAC	0.0121	0.0129	ARIMA(X)	0.0106	0.0125	ARIMA(X)
COIN	0.0337	0.0425	ARIMA(X)	0.0349	0.0490	ARIMA(X)
GOOG	0.0120	0.0117	XGBoost	0.0138	0.0179	ARIMA(X)
INTC	0.0171	0.0245	ARIMA(X)	0.0288	0.0411	ARIMA(X)
JNJ	0.0089	0.0116	ARIMA(X)	0.0145	0.0097	XGBoost
JPM	0.0103	0.0130	ARIMA(X)	0.0144	0.0190	ARIMA(X)
META	0.0122	0.0149	ARIMA(X)	0.0127	0.0203	ARIMA(X)
MSFT	0.0094	0.0101	ARIMA(X)	0.0098	0.0125	ARIMA(X)
NFLX	0.0140	0.0156	ARIMA(X)	0.0138	0.0220	ARIMA(X)
NVDA	0.0206	0.0298	ARIMA(X)	0.0299	0.0451	ARIMA(X)
PFE	0.0115	0.0131	ARIMA(X)	0.0098	0.0136	ARIMA(X)
TSLA	0.0215	0.0381	ARIMA(X)	0.0324	0.0444	ARIMA(X)
TTE	0.0077	0.0094	ARIMA(X)	0.0068	0.0116	ARIMA(X)
WMT	0.0088	0.0144	ARIMA(X)	0.0084	0.0104	ARIMA(X)
XOM	0.0116	0.0141	ARIMA(X)	0.0132	0.0159	ARIMA(X)
SPY	0.0051	0.0149	ARIMA(X)	0.0087	0.0132	ARIMA(X)
IWDA.AS	0.0042	0.0088	ARIMA(X)	0.0079	0.0122	ARIMA(X)
QQQ	0.0077	0.0144	ARIMA(X)	0.0133	0.0189	ARIMA(X)

Table 10: RMSE comparison between ARIMA(X) and XGBoost across two folds.

Appendix 18

Ticker	Fold	DM Statistic	P-value	Conclusion
AAPL	Fold 1	-2.7627	0.0106	ARIMA/ARIMAX
AAPL	Fold 2	-3.0462	0.0054	ARIMA/ARIMAX
AMZN	Fold 1	-1.0000	0.3269	No difference
AMZN	Fold 2	-2.6357	0.0142	ARIMA/ARIMAX
BAC	Fold 1	-1.1326	0.2681	No difference
BAC	Fold 2	-1.4496	0.1596	No difference
COIN	Fold 1	-2.4240	0.0229	ARIMA/ARIMAX
COIN	Fold 2	-1.1910	0.2448	No difference
GOOG	Fold 1	-2.2331	0.0347	ARIMA/ARIMAX
GOOG	Fold 2	-0.2948	0.7706	No difference
INTC	Fold 1	-2.5702	0.0165	ARIMA/ARIMAX
INTC	Fold 2	-1.7149	0.0987	No difference
JNJ	Fold 1	-2.6890	0.0126	ARIMA/ARIMAX
JNJ	Fold 2	1.0830	0.2891	No difference
JPM	Fold 1	-1.8833	0.0713	No difference
JPM	Fold 2	-2.2693	0.0321	ARIMA/ARIMAX
META	Fold 1	-1.7836	0.0866	No difference
META	Fold 2	-1.3412	0.1919	No difference
MSFT	Fold 1	-1.9427	0.0634	No difference
MSFT	Fold 2	-0.6603	0.5151	No difference
NFLX	Fold 1	-2.8553	0.0085	ARIMA/ARIMAX
NFLX	Fold 2	-1.1065	0.2791	No difference
NVDA	Fold 1	-1.9884	0.0578	No difference
NVDA	Fold 2	-2.5515	0.0172	ARIMA/ARIMAX
PFE	Fold 1	-2.0539	0.0506	No difference
PFE	Fold 2	-1.2808	0.2120	No difference
TSLA	Fold 1	-3.3878	0.0023	ARIMA/ARIMAX
TSLA	Fold 2	-2.5240	0.0183	ARIMA/ARIMAX
TTE	Fold 1	-4.4132	0.0002	ARIMA/ARIMAX
TTE	Fold 2	-1.3199	0.1988	No difference
WMT	Fold 1	-1.7196	0.0979	No difference
WMT	Fold 2	-2.8598	0.0084	ARIMA/ARIMAX
XOM	Fold 1	-2.0542	0.0505	No difference
XOM	Fold 2	-1.6315	0.1153	No difference
SPY	Fold 1	-2.5312	0.0180	ARIMA/ARIMAX
SPY	Fold 2	-5.5994	0.0000	ARIMA/ARIMAX
IWDA.AS	Fold 1	-2.6851	0.0122	ARIMA/ARIMAX
IWDA.AS	Fold 2	-3.3519	0.0024	ARIMA/ARIMAX
QQQ	Fold 1	-2.0059	0.0558	No difference
QQQ	Fold 2	-3.0308	0.0056	ARIMA/ARIMAX

Table 11: Diebold-Mariano test results indicating which model is significantly better for each ticker and fold.

Appendix 19

Ticker	ARIMA(X) RMSE (F1)	LR RMSE (F1)	Best RMSE (F1)	ARIMA(X) RMSE (F2)	LR RMSE (F2)	Best RMSE (F2)
AAPL	0.0082	0.0087	ARIMA(X)	0.0086	0.0082	LR
AMZN	0.0090	0.0139	ARIMA(X)	0.0142	0.0091	LR
BAC	0.0121	0.0107	LR	0.0106	0.0121	ARIMA(X)
COIN	0.0337	0.0346	ARIMA(X)	0.0349	0.0341	LR
GOOG	0.0120	0.0139	ARIMA(X)	0.0138	0.0118	LR
INTC	0.0171	0.0279	ARIMA(X)	0.0288	0.0175	LR
JNJ	0.0089	0.0065	LR	0.0145	0.0089	LR
JPM	0.0103	0.0145	ARIMA(X)	0.0144	0.0103	LR
META	0.0122	0.0125	ARIMA(X)	0.0127	0.0122	LR
MSFT	0.0094	0.0098	ARIMA(X)	0.0098	0.0092	LR
NFLX	0.0140	0.0140	LR	0.0138	0.0140	ARIMA(X)
NVDA	0.0206	0.0296	ARIMA(X)	0.0299	0.0208	LR
PFE	0.0115	0.0099	LR	0.0098	0.0115	ARIMA(X)
TSLA	0.0215	0.0315	ARIMA(X)	0.0324	0.0212	LR
TTE	0.0077	0.0067	LR	0.0068	0.0075	ARIMA(X)
WMT	0.0088	0.0084	LR	0.0084	0.0090	ARIMA(X)
XOM	0.0116	0.0132	ARIMA(X)	0.0132	0.0117	LR
SPY	0.0051	0.0088	ARIMA(X)	0.0087	0.0052	LR
IWDA.AS	0.0042	0.0081	ARIMA(X)	0.0079	0.0040	LR
QQQ	0.0077	0.0133	ARIMA(X)	0.0133	0.0077	LR

Table 12: RMSE comparison between ARIMA(X) and LR models across two folds.

Appendix 20

Ticker	Fold	DM Statistic	P-value	Conclusion
AAPL	Fold 1	1.0509	0.3030	No difference
AAPL	Fold 2	1.1466	0.2620	No difference
AMZN	Fold 1	0.7382	0.4670	No difference
AMZN	Fold 2	-0.8952	0.3789	No difference
BAC	Fold 1	0.6490	0.5220	No difference
BAC	Fold 2	0.7427	0.4643	No difference
COIN	Fold 1	1.4823	0.1503	No difference
COIN	Fold 2	-0.1920	0.8492	No difference
GOOG	Fold 1	1.7500	0.0919	No difference
GOOG	Fold 2	0.8656	0.3946	No difference
INTC	Fold 1	0.7245	0.4752	No difference
INTC	Fold 2	0.4023	0.6908	No difference
JNJ	Fold 1	2.0149	0.0544	No difference
JNJ	Fold 2	3.1875	0.0037	Linear Regression
JPM	Fold 1	-1.0899	0.2858	No difference
JPM	Fold 2	0.4722	0.6408	No difference
META	Fold 1	0.0847	0.9331	No difference
META	Fold 2	1.0923	0.2847	No difference
MSFT	Fold 1	0.8791	0.3874	No difference
MSFT	Fold 2	1.3581	0.1861	No difference
NFLX	Fold 1	1.0576	0.3000	No difference
NFLX	Fold 2	0.7259	0.4744	No difference
NVDA	Fold 1	-0.3143	0.7558	No difference
NVDA	Fold 2	1.2644	0.2173	No difference
PFE	Fold 1	0.1143	0.9099	No difference
PFE	Fold 2	1.1411	0.2642	No difference
TSLA	Fold 1	-1.0593	0.2992	No difference
TSLA	Fold 2	-0.7756	0.4450	No difference
TTE	Fold 1	1.4920	0.1477	No difference
TTE	Fold 2	1.2564	0.2201	No difference
WMT	Fold 1	-1.3202	0.1983	No difference
WMT	Fold 2	0.9903	0.3312	No difference
XOM	Fold 1	0.8762	0.3889	No difference
XOM	Fold 2	0.1040	0.9180	No difference
SPY	Fold 1	-0.5233	0.6052	No difference
SPY	Fold 2	1.0303	0.3123	No difference
IWDA.AS	Fold 1	1.4036	0.1718	No difference
IWDA.AS	Fold 2	-1.0080	0.3224	No difference
QQQ	Fold 1	0.9489	0.3514	No difference
QQQ	Fold 2	0.9469	0.3524	No difference

Table 13: Diebold-Mariano test conclusion by fold

Appendix 21

Ticker	Local XGB (F1)	Global XGB (F1)	F1 Better	Local XGB (F2)	Global XGB (F2)	F2 Better
AAPL	0.0125	0.0135	Local XGBBoost	0.0179	0.0088	Global XGBBoost
AMZN	0.0159	0.0165	Local XGBBoost	0.0153	0.0100	Global XGBBoost
BAC	0.0129	0.0132	Local XGBBoost	0.0125	0.0129	Local XGBBoost
COIN	0.0425	0.0482	Local XGBBoost	0.0490	0.0342	Global XGBBoost
GOOG	0.0117	0.0182	Local XGBBoost	0.0179	0.0132	Global XGBBoost
INTC	0.0245	0.0478	Local XGBBoost	0.0411	0.0269	Global XGBBoost
JNJ	0.0116	0.0069	Global XGBBoost	0.0097	0.0098	Local XGBBoost
JPM	0.0130	0.0171	Local XGBBoost	0.0190	0.0107	Global XGBBoost
META	0.0149	0.0179	Local XGBBoost	0.0203	0.0143	Global XGBBoost
MSFT	0.0101	0.0139	Local XGBBoost	0.0125	0.0110	Global XGBBoost
NFLX	0.0156	0.0162	Local XGBBoost	0.0220	0.0148	Global XGBBoost
NVDA	0.0298	0.0332	Local XGBBoost	0.0451	0.0227	Global XGBBoost
PFE	0.0131	0.0120	Global XGBBoost	0.0136	0.0122	Global XGBBoost
TSLA	0.0381	0.0379	Global XGBBoost	0.0444	0.0226	Global XGBBoost
TTE	0.0094	0.0117	Local XGBBoost	0.0116	0.0095	Global XGBBoost
WMT	0.0144	0.0101	Global XGBBoost	0.0104	0.0092	Global XGBBoost
XOM	0.0141	0.0161	Local XGBBoost	0.0159	0.0113	Global XGBBoost
SPY	0.0149	0.0112	Global XGBBoost	0.0132	0.0061	Global XGBBoost
IWDA.AS	0.0088	0.0110	Local XGBBoost	0.0122	0.0046	Global XGBBoost
QQQ	0.0144	0.0167	Local XGBBoost	0.0189	0.0086	Global XGBBoost

Table 14: Comparison of RMSE scores and best performing model (Local vs Global XGBBoost) across two folds.

Appendix 22

Ticker	Fold	DM Statistic	P-value	Conclusion
AAPL	Fold 1	1.7484	0.0927	No difference
AAPL	Fold 2	3.3064	0.0028	Global XGBoost
AMZN	Fold 1	-1.5066	0.1445	No difference
AMZN	Fold 2	2.6230	0.0144	Global XGBoost
BAC	Fold 1	-1.1637	0.2555	No difference
BAC	Fold 2	0.3576	0.7235	No difference
COIN	Fold 1	0.1795	0.8590	No difference
COIN	Fold 2	1.1553	0.2585	No difference
GOOG	Fold 1	-0.0337	0.9734	No difference
GOOG	Fold 2	-1.0755	0.2920	No difference
INTC	Fold 1	-1.6035	0.1214	No difference
INTC	Fold 2	-1.1295	0.2690	No difference
JNJ	Fold 1	1.9301	0.0650	No difference
JNJ	Fold 2	1.9568	0.0612	No difference
JPM	Fold 1	1.2045	0.2397	No difference
JPM	Fold 2	2.2068	0.0364	Global XGBoost
META	Fold 1	1.8266	0.0797	No difference
META	Fold 2	0.2877	0.7759	No difference
MSFT	Fold 1	-1.4993	0.1463	No difference
MSFT	Fold 2	-0.7357	0.4685	No difference
NFLX	Fold 1	3.4101	0.0022	Global XGBoost
NFLX	Fold 2	0.2901	0.7741	No difference
NVDA	Fold 1	1.8713	0.0731	No difference
NVDA	Fold 2	1.8213	0.0801	No difference
PFE	Fold 1	1.1783	0.2498	No difference
PFE	Fold 2	0.9563	0.3477	No difference
TSLA	Fold 1	1.5672	0.1296	No difference
TSLA	Fold 2	2.6795	0.0126	Global XGBoost
TTE	Fold 1	-0.1184	0.9067	No difference
TTE	Fold 2	-0.1427	0.8876	No difference
WMT	Fold 1	0.2930	0.7720	No difference
WMT	Fold 2	3.3189	0.0027	Global XGBoost
XOM	Fold 1	-0.4304	0.6706	No difference
XOM	Fold 2	2.0480	0.0508	No difference
SPY	Fold 1	1.4975	0.1468	No difference
SPY	Fold 2	4.3403	0.0002	Global XGBoost
IWDA.AS	Fold 1	2.4180	0.0226	Global XGBoost
IWDA.AS	Fold 2	2.7054	0.0117	Global XGBoost
QQQ	Fold 1	1.6407	0.1134	No difference
QQQ	Fold 2	2.2596	0.0325	Global XGBoost

Table 15: Diebold-Mariano test conclusion by fold

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Louvain School of Management

Place des Doyens, 1 bte L2.01.01, 1348 Louvain-la-Neuve
Boulevard Emile Devreux 6, 6000 Charleroi, Belgique
Chaussée de Binche 151, 7000 Mons, Belgique

www.uclouvain.be/lsm