Seminar Paper

B.Sc.-Studiengang "'Wirtschaftsingenieurwesen"'

University of Hamburg
Faculty of Business Administration
Professorship of Data Science
Prof. Dr. Anne Lauscher

Interest Rates and Their Predictive Power on the Stock Market

Submitted by:

Aaron Gresser

7617171

Grandweg 16

22529 Hamburg

0160 97537566

Aaron.Gresser@gmail.com

Supervised by:

Prof. Dr. Anne Lauscher

Submission Date and Location:

30.06.2025, Hamburg

Abstract

Predicting stock market movements remains a major challenge due to the complexity of financial markets and the diversity of market participants. Interest rate change announcements by central banks, such as the European Central Bank (ECB), are key macroeconomic events that can trigger immediate reactions in asset prices. This paper investigates whether these announcements create predictable movements in stock market indices, and whether incorporating sentiment analysis of ECB press conference releases can improve short-term market forecasts. To address these questions, we compare a simple baseline approach, linear regression, and polynomial regression. Sentiment scores are generated from full press conference transcripts using a fine-tuned financial language model (FinBERT) and combined with financial features. Model performance is evaluated using mean squared error (MSE) and the coefficient of determination (R^2) on a custom dataset covering multiple ECB announcement cycles.Results show that polynomial regression provides the clearest improvements, especially for three-day forecasts, while sentiment analysis enhances training performance but does not generalize to the test set. These findings highlight both the potential and limits of integrating sentiment analysis with financial features, and suggest that future work should focus on larger datasets and more robust modeling strategies to better capture the effects of central bank communication.

Contents

List of Figures

1 Introduction

The development of financial markets is notoriously difficult to predict due to their inherent complexity. Markets are shaped by participants with diverse backgrounds and motivations, and as entry barriers continue to decrease, an ever growing number of actors enter these systems. This sheer mass of competing participants makes it increasingly challenging to identify a genuine informational edge in an increasingly complex and chaotic world market.

The central research question guiding this work asks whether interest rate change announcements by the European Central Bank (ECB) induce predictable movements in the stock market. In pursuit of this question, this paper further examines two extensions: first, whether incorporating sentiment analysis of the ECB press conference releases enhances the performance of stock price predictions, and second, whether implementing more complex machine learning algorithms increases this effect further.

To address these questions, three modeling approaches are compared: a simple baseline, linear regression, and polynomial regression. Sentiment scores for the full press conference releases are generated using fine-tuned large language models (LLMs), with FinBERT selected after cross-validation of two financial variants. These sentiment features are then combined with financial data. Model training and hyperparameter tuning are conducted on a custom dataset spanning multiple ECB announcement cycles, and forecasting performance is evaluated using mean squared error (MSE) and the coefficient of determination (R^2).

The structure of the paper is as follows. Section 2, Theoretical Background, explains the relevant financial context and the machine learning methods applied in this analysis. Section 3, Data Description, introduces the data sources, dataset construction, and preprocessing steps, including sentiment-score generation. Section 4, Methodology, outlines the methodological approach, model selection, and hyperparameter configuration. Section 5, Experimental Setup, provides a deep dive into the models, presents their parameters, and explains the evaluation strategies. Section 6, Evaluation and Analysis, analyzes and interprets the results of the models. Finally, Section 7, Conclusion, summarizes the key findings, discusses limitations, and suggests directions for future research.

2 Theoretical Background

This Section outlines the theoretical foundations relevant to both the financial context and the machine learning approaches applied in this paper. By reviewing key economic principles, introducing central machine learning algorithms, and detailing essential data preprocessing techniques, it provides the necessary background for the analyses conducted in the following sections.

2.1 Economic Foundations

Stock Markets

Stock markets are complex systems where a wide range of participants with different interests and backgrounds come together to trade. Because so many actors interact and react to new information, price movements are often hard to predict and can be driven by a mix of data, speculation, and market sentiment. This complexity makes it challenging to consistently identify clear patterns or forecast future developments.

Interest Rates and Deposit facility

Interest Rates serve as the primary monetary policy instrument used by central banks to influence economic conditions. The ECB sets different types of interest rates, but this paper focuses on the deposit facility rate. This is the rate at which banks receive interest when making overnight deposits with the ECB.adjusting this rate, the ECB can influence banks incentives to either hold excess funds at the central bank or to reinvest them into the broader economy.(ECB 2025)

Efficient Market Hypothesis (EMH)

The Efficient Market Hypothesis (EMH) posits that financial markets efficiently incorporate all available information into asset prices. EMH is commonly described in three forms: the weak form (prices reflect all past market data), the semi-strong form (prices reflect all publicly available information), and the strong form (prices reflect all information, including private or insider knowledge). According to EMH, it is not possible to consistently achieve returns above the market average except by chance, as new information is rapidly and fully reflected in prices.

2.2 Machine Learning Algorithms

Linear Regression

Linear regression fits a linear model to the data by minimizing the residual sum of squares between the observed targets and predicted values (ordinary least squares). The prediction formula is:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + ... + w_p x_p \tag{2.1}$$

(Scikit-Learn 2025, User Guide: 1.1.Linear Models)

Polynomial Regression

Polynomial regression extends linear regression by including polynomial terms of the input features, allowing the model to capture non-linear relationships. The model equation for degree d is:

$$\hat{y} = w_0 + w_1 x + w_2 x^2 + \ldots + w_d x^d \tag{2.2}$$

Bayesian Ridge Regression

Bayesian Ridge Regression models regression in a probabilistic way by assuming the coefficients w are random variables with a normal (Gaussian) distribution. It automatically estimates the regularization parameters α and λ from the data. The distribution of the coefficients is:[1]

$$p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1} I_p) \tag{2.3}$$

This means each coefficient w_j is normally distributed with mean 0 and variance λ^{-1}:

$$w_j \sim \mathcal{N}(0, \lambda^{-1}) \tag{2.4}$$

A key advantage of Bayesian Ridge Regression is that it outputs not only a prediction for the target value, but also an estimate of the uncertainty $\hat{\sigma}$ for each prediction, allowing us to quantify our confidence in the results. This method is especially robust when the data is noisy or the problem is ill-posed[2].

(Scikit-Learn 2025, User Guide: 1.1.10.1 Bayesian Ridge Regression)

[1] I_p denotes the identity matrix of size $p \times p$, where p is the number of features (predictors) in the model.

[2] An *ill-posed* problem may lack a solution, have multiple solutions, or be highly sensitive to input data; see `https://en.wikipedia.org/wiki/Well-posed_problem`.

2.3 Data Preprocessing Fundamentals

Standardization and Normalization

Standardization and normalization are essential steps to ensure features are on a comparable scale before . Standardization, as implemented by the StandardScaler, transforms each feature by subtracting the mean (μ) and dividing by the standard deviation (σ):

$$z = \frac{x - \mu}{\sigma} \;\rightarrow\; \mu_z = 0,\ \sigma_z = 1 \tag{2.5}$$

This results in features with zero mean (μ_z) and unit variance (σ_z^2).
(Scikit-Learn 2025, User Guide: 7.3.Preprocessing data)

Polynomial Transformation

Polynomial transformation creates new features by generating all polynomial combinations of the original features up to the specified degree. For two input features, X_1 and X_2, a degree-2 polynomial transformation produces the following expanded feature vector:

$$(1,\ X_1,\ X_2,\ X_1^2,\ X_1X_2,\ X_2^2) \tag{2.6}$$

This transformation allows linear models to capture nonlinear relationships by including squared and interaction terms among the features
(Scikit-Learn 2025, User Guide: 7.3.7.1. Polynomial features).

Categorical Encoding

Since most machine learning algorithms cannot directly handle categorical variables, these features must be transformed into numeric representations. One-hot encoding is a common technique that creates a separate binary feature for each category, preventing the introduction of artificial order and ensuring that categorical data is accurately represented for analysis.
(Scikit-Learn 2025, User Guide: 7.3.4. Encoding categorical features).

3 Data Description

This section presents the dataset, preprocessing, and feature engineering steps. The raw and processed data files are located in the directories "01_RawData" and "03_Dataset_Creation", with final datasets in "99_Kaggle\Uploads to Kaggle\Data Sets".

Visualizations

For detailed visualizations of feature distributions and correlations, see *Appendix A: Dataset*. The code used for these visualizations is provided in "03_Dataset Visualization".

3.1 Financial Data

Financial data was downloaded from Yahoo Finance using the yfinance[3] library. The original data contains daily open, high, low, and close prices as well as trading volume.

Preprocessing

During preprocessing, the columns High, Low, and Open were deleted, since the main focus of this analysis is on the interest rate data. Therefore, each instance only contains the last 4 closing prices. These are labeled as close_t-4, close_t-3, close_t-2, close_t-1, and the targets are close_t+1, close_t+2, close_t+3. Additionally, the data is one-hot encoded by index, as we have three indices: DAX, MDAX, and SDAX.

Feature Engineering

Because the dataset is so small, the focus was on minimizing the number of features. For this reason, the DAX column was deleted in the first step, since DAX can also be represented with: MDAX = 0 and SDAX = 0. Furthermore, the close prices vary widely in value. There are two problems here: (1) Each stock index has a completely different absolute valuation, and (2) the price values are spread over 3 years, and the values of the stock indices do not represent a stationary time series, which can also cause problems in machine learning. To reduce these problems, all price values are scaled. All close_t-i and close_t+i values are scaled as percentage changes relative to close_t-1, according to the following formula:

$$\frac{close_{t\pm i} - close_{t-1}}{close_{t-1}} \cdot 100$$

[3] https://pypi.org/project/yfinance/

This also has the effect that close_t+i becomes 0 throughout the dataset, so this column can also be deleted.

3.2 ECB Data

For the analysis, two main data sources from the official website of the ECB(ECB 2025) were used:

1. Interest Rate Changes (.xlsx): The daily Deposit Facility Rate of the ECB and its daily changes.

2. Press Conference Statements (.pdf): For each monetary policy decision, the ECB provides a full transcript of the press conference as a PDF. These contain the opening statements of the ECB president and thematically structured explanations.

Preprocessing

The press conference statement PDF files were converted to .txt files using the pypdf[4] library to make them usable for sentiment analysis. First, each PDF was fully converted into a single text string. Then, the content was segmented into its six thematic sections (Conclusion, Inflation, Economic Activity, Risk Assessment, Press Conference, Financial Monetary Conditions) to enable more specific analyses if necessary.
The interest rate dataset was filtered so that only those days for which a press conference transcript is available remain, and for each of these days, the Interest Rate Old (rate before the conference) and the Interest Rate Change (the change decided) are available.

Sentiment Analysis

For sentiment analysis, the complete text of the ECB press conference is used, since preliminary experiments indicated that this yields the most meaningful results. Two segmentation methods are used: On the one hand, the entire press conference text is split into overlapping token chunks, using a sliding window of 512 tokens and a stride of 400 tokens (i.e., 112 token overlap), creating coherent text windows that preserve longer context passages without exceeding the transformer model's maximum input length. In parallel, sentence-based segmentation is performed using the NLTK PunktSentenceTokenizer[5], to which industry-specific abbreviations such as "e.g.", "cpi", or "ecb" are added in advance to prevent incorrect sentence breaks.

[4] `https://pypdf.readthedocs.io/en/stable/user/extract-text.html`
[5] `https://www.nltk.org/api/nltk.tokenize.punkt.html`

A pre-trained financial transformer model (FinBERT[6] or Financial-RoBERTa[7]) then evaluates each chunk and each sentence. The model provides probabilities for "positive", "neutral", and "negative". The optimism score of a segment is calculated as

$$\text{Optimism Score} = P(\text{positive}) - P(\text{negative})$$

resulting in values in the interval $[-1, +1]$. For the overall evaluation, the arithmetic mean of all chunk or sentence scores is calculated. This pipeline combines standardized text preprocessing with a clear, comparable metric and thus enables direct comparison of different models and segmentation methods.

Example:

Table 1 shows sample optimism scores for three selected ECB press conferences, comparing both FinBERT and RoBERTa models using sentence and chunk segmentation.

Date	FinBERT_Sentences	FinBERT_Chunks	RoBERTa_Sentences	RoBERTa_Chunks
09_June_2022	0.196	0.119	0.036	-0.069
21_July_2022	0.260	0.245	0.094	0.231
08_September_2022	0.012	-0.006	-0.174	-0.245

Table 1: Example sentiment analysis outputs for selected ECB press conferences. For the full results, see "Appendix F: Sentiment Analysis Outputs".

[6] https://github.com/ProsusAI/finBERT

[7] https://huggingface.co/ynie/roberta-financial-news-sentiment-en

4 Methodology

Dataset

The dataset covers the last 3 years (09.06.2022 – 05.06.2025) and thus includes the last 25 ECB press conferences. This time limitation was chosen because the financial environment is constantly changing, and especially during the Corona pandemic the financial market behaved very differently compared to recent years. To minimize this distortion, the period is therefore limited. Two outlier dates were excluded due to exceptional external influences: 09.06.2022 (interest rate regime change) and 12.12.2024 (concurrent Federal Reserve meeting)

Feature Engineering

The models predict the percentage change in closing prices on the day of the press conference (Close), as well as one and two days after (Close_t+1, Close_t+2), each relative to the previous day's close. To focus on the ECB's interest rate decisions and the value of sentiment analysis, only a minimal financial feature set and selected sentiment scores are used. In total, the dataset consists of 69 instances with 11 features (5 financial, 2 one-hot encoded, and 4 sentiment scores). The base dataset uses 7 features, while the sentiment analysis dataset includes 8. Table 2 summarizes all features included in the dataset.

Feature Name	**Description**
Close_t-4 to Close_t-2	% change in closing prices, relative to Close_t-1
Index_MDAX, Index_SDAX	One Hot encoded index category
Interest Rate_Old	Interest rate before the press conference
Interest Rate_Change	Change in interest rate decided
FinBERT/RoBERTa Scores	Sentiment scores (Chunks and Sentences)
Close, Close_t+1, Close_t+2	Target variables, relative to Close_t-1

Table 2: Overview of dataset features and their descriptions. All features are of type float.

Train/Test Split

The training set consists of 54 instances from 18 days, while the test set contains 15 instances from 5 days. Test days were selected randomly (seed 33) under two conditions:
(1) all three instances, from each of the five chosen days, are included in the test set to prevent data leakage from correlated price changes.
(2) each interest rate change category (>0, $=0$, <0) must appear at least once in the test set, to ensure a similar distribution.

This results in the following distribution:[8]

	Training Set	Test Set
Number of Instances	54	15
Interest Rate Change >0% (<0%)	44.4 (22.2)	40.0 (40.0)

Table 3: Overview of training and test set composition. Values in parentheses indicate the percentual proportion of negative Interest Rate Changes, values without positive changes.

	Close	Close_t+1	Close_t+2
Mean	0.28 [0.66]	0.35 [-0.02]	0.26 [-0.44]
Std. Dev.	1.27 [0.94]	1.60 [1.16]	1.69 [1.34]
>0% Change	59.3 [66.7]	63.0 [46.7]	50.0 [33.3]
<0% Change	40.7 [33.3]	37.0 [53.3]	50.0 [66.7]

Table 4: Descriptive statistics of target variables for training and test sets. Bracketed values correspond to the [Test] set.

Hyperparameter Configuration

Hyperparameters were selected using an extensive Cross Validation (CV) grid search, as documented in "04_Hyperparameter_Configuration".This search was performed on the training set without sentiment analysis scores. To prevent data leakage, shuffle was set to False, and the number of folds n was chosen so that the validation set size was always divisible by 3, ensuring that only whole days were included in each fold. Parameters were evaluated based on their average CV mean squared error (CV_MSE), with the lowest CV_MSE determining the optimal choice.

Initially, different CV fold counts ($n = 9$ and $n = 18$) and scaling methods (None vs. StandardScaler) were compared. Using $n = 9$ folds yielded the best generalization and least overfitting, while StandardScaler consistently achieved the lowest CV_MSE. With these settings fixed, regularization methods (none, Ridge Regression, Bayesian Ridge Regression) were evaluated, and Bayesian Ridge Regression was selected. Its parameters were further tuned in the next step. For sentiment analysis, four combinations (Chunks vs. Sentences and FinBERT vs. RoBERTa) were compared using the previously selected hyperparameters. The FinBERT x Sentences combination performed best on average across all models.

To ensure comparability between linear and polynomial regression, the polynomial grid search only tuned the polynomial order n and Bayesian Ridge parameters, using the training set without sentiment analysis scores.

[8]For detailed visualizations of the train and test set distributions, see "Appendix B: Train/Test Split Visualizations". The code used for these visualizations is provided in "03_Dataset Visualization".

5 Experimental Setup

All algorithms used in this setup were implemented in Python using the Scikit-Learn library. Hyperparameters were selected as described in Section 4, including the sentiment pipeline configuration. The FinBERT language model with sentence tokenization yielded the best predictive performance. To provide a benchmark, a simple baseline model was implemented. The main algorithms used were Linear Regression and Polynomial Regression, both utilizing the *StandardScaler* for feature scaling and *Bayesian Ridge Regression* for regularization to prevent overfitting. Bayesian Ridge Regression also provided an uncertainty estimate for each prediction in the form of $\hat{\sigma}$.

Baseline Approach

The baseline model predicts each target value by the mean of the training set:

$$\hat{y}_t = \bar{y} \; = \; \frac{1}{N} \sum_{i=1}^{N} y_i. \tag{5.1}$$

This simple mean predictor serves as a reference point for assessing the performance gains of using complexer models.(Scikit-Learn 2025, .dummy.DummyRegressor)

Linear Regression

For the simple Models, the Linear Regression Algorithm was used. Most of the Parameters (α_1, α_2, λ_1, λ_2, λ_{init}) are near,or at, their default value. Not so n_iter, which stands for the Iterations, after the Regression stops.[9]

The Bayesian Ridge Regression was implemented with the following parameters:

	n_iter	tol	α_1	α_2	λ_1	λ_2	α_{init}	λ_{init}	fit_intercept
Close	1	1e-07	1e-06	1e-05	1e-05	1e-06	0.1	None	False
Close_t+1	1	1e-07	1e-05	1e-06	1e-06	1e-05	1.0	0.01	False
Close_t+2	1	1e-07	1e-06	1e-05	1e-05	1e-06	0.01	None	False

Table 5: Model Parameters for the Bayesian Ridge Regression; Linear Regression

Polynomial Regression

Polynomial Regression was used for the more complex models. By applying a polynomial feature transformation, this approach can capture non-linear relationships between features. This is likely why configuring the regression was easier in this case. All polynomial models

[9]In the hyperparameter tuning process, it became apparent that the algorithm consistently converged to the mean prediction. This was likely due to the small dataset size or the unequal distribution of target values, resulting in low inherent information value because of the train/test split.

were implemented with polynomial order $n = 2$ and identical regression parameters for all targets.

The Bayesian Ridge Regression was configured as follows:

	n_iter	tol	α_1	α_2	λ_1	λ_2	α_{init}	λ_{init}	fit_intercept
All	1	1e-07	1e-06	1e-05	1e-05	1e-06	0.01	None	False

Table 6: Model Parameters for the Bayesian Ridge Regression; polynomial Regression (n=2)

Training

Each algorithm was used to train three separate models, one for each target variable (Close, Close_t+1, Close_t+2). This resulted in six models per dataset, and a total of twelve models across both the standard and sentiment datasets. Models were trained on the entire training set (54 instances). The standard dataset included 7 numerical features, while the sentiment dataset included 8. For evaluation, the trained models predicted the target values for the test set (15 instances).[10].

5.1 Evaluaton Metrics

The following metrics were used to evaluate model performance:

Coefficient of Determination (R^2)

The coefficient of determination indicates the proportion of variance in the observed data explained by the model:

$$R^2 \;=\; 1 \;-\; \frac{\sum_{i=1}^{N} \big(y_i - \hat{y}_i\big)^2}{\sum_{i=1}^{N} \big(y_i - \bar{y}\big)^2} \tag{5.2}$$

A perfect model achieves $R^2 = 1.0$, while a constant mean predictor yields $R^2 = 0.0$ and poorer models can produce negative scores.(Scikit-Learn 2025, metrics.r2_score)

Mean Squared Error (MSE)

The MSE computes the average of the squared differences between true and predicted values, which serves as a risk metric

$$\text{MSE} \;=\; \frac{1}{N} \sum_{i=1}^{N} \big(y_i - \hat{y}_i\big)^2 \tag{5.3}$$

[10]Visualizations of the results are provided in "Appendix C: Model Results" The code can be found in "05_Modell Training"

Lower values indicate better model performance, with 0.0 corresponding to perfect predictions.(Scikit-Learn 2025, model_evaluation.mean-squared-error)

6 Results and Evaluation

This section presents the quantitative results of the forecasting experiments, showing how each model performed across one, two, and three day prediction horizons. First, the evaluation metrics for all models are summarized in a table. Afterwards, model performances are compared, patterns are interpreted, and the impact of sentiment analysis and model complexity is discussed. Finally, setup failures and their causes are analyzed.

Results

Table 7 lists all 12 models with their performance on both the Train and Test sets. For each prediction horizon, the best values for each evaluation metric are highlighted in bold. The optimal value for R^2 is 1.00, while the optimal value for MSE is 0.00. The Baseline serves as a benchmark for comparison.

Prediction	Model	Typ	Train		Test	
			MSE	R^2	MSE	R^2
1 day	Baseline	-	1.58	0.00	0.97	-0.17
	Linear Regression	Standard	1.25	0.20	1.35	-0.62
	Linear Regression	with Sentiment Analysis	**1.17**	**0.26**	1.43	-0.72
	Polynomial Regression	Standard	1.32	0.16	**0.89**	**-0.07**
	Polynomial Regression	with Sentiment Analysis	1.19	0.24	0.99	-0.18
2 days	Baseline	-	2.50	0.00	**1.39**	**-0.11**
	Linear Regression	Standard	1.61	0.35	4.10	-2.26
	Linear Regression	with Sentiment Analysis	**1.58**	**0.37**	4.24	-2.38
	Polynomial Regression	Standard	1.86	0.25	2.16	-0.72
	Polynomial Regression	with Sentiment Analysis	1.78	0.29	2.29	-0.82
3 days	Baseline	-	2.79	0.00	2.16	-0.30
	Linear Regression	Standard	2.65	0.05	1.97	-0.18
	Linear Regression	with Sentiment Analysis	2.63	0.06	1.96	-0.17
	Polynomial Regression	Standard	1.90	0.32	**1.37**	**0.18**
	Polynomial Regression	with Sentiment Analysis	**1.62**	**0.42**	1.43	0.14

Table 7: Model Performance Comparison on the Trainset and Testset, categorized by prediction target and sorted by model.

Evaluation

The first striking result is that in the two day forecast category, no model outperforms the Baseline on the Test set, suggesting the train/test split may have been suboptimal. In the one day forecast, the polynomial standard model beats the Baseline in MSE but yields a negative R^2, indicating it still underperforms a constant mean predictor. Only in the three day horizon do both polynomial models achieve positive R^2 and clearly better MSE than the Baseline.

Overall, the best performing model on the Test set across all forecasts is the Polynomial Regression without Sentiment Analysis. The Polynomial Regression with Sentiment Analysis ranks second, yet only on the three day horizon does it perform meaningfully better than the Baseline.

In contrast, the Train set reveals a different pattern. As expected, every model beats the Baseline across all three horizons. Interestingly, the sentiment-augmented model consistently outperforms its counterpart without Sentiment Analysis in both R^2 and MSE. This suggests that Sentiment Analysis does provide additional predictive information. However, it also implies these models may be more prone to overfitting given the small dataset, potentially reducing their generalization ability. Another reason for this phenomenon may be that both the standard and sentiment models use identical regularization parameters, which were optimized for the standard model. This was a specific choice to enable better comparison between models. However, since the sentiment model has additional information content, it might actually require stronger regularization to prevent overfitting.

As a third pattern, it can be observed that polynomial models always outperform linear models on the Test set, regardless of the prediction horizon. However, on the Train set, polynomial models only outperform linear models for the three day forecast. For the one and two day forecasts, the linear models actually perform better on the training data. This suggests that the higher complexity of the polynomial models provides a generalization advantage on unseen data, and that non-linear feature dependencies become more relevant for longer-range predictions. Which makes sense since longer range predictions involve increasingly complex dependencies that non-linear models can capture more effectively.

6.1 Error Analysis

As mentioned in the Evaluation, there are several shortcomings of the setup. The dataset is not large enough to yield statistically significant results, and due to data sparsity and

unequal distributions in the train and test sets, obtaining meaningful inference is a matter of chance. It could be considered a stroke of luck, or perhaps just gamblers luck, that any predictions performed better than the baseline.

Statistical Power Limitations

With only 69 instances in the Dataset, this study does not have enough data to reliably detect real effects. Small sample sizes mean that even if a true relationship exists, the model might not be able to pick it up. This low statistical power increases the risk of missing actual patterns and makes it hard to draw strong conclusions. As a result, any positive results could just be due to random chance.

Overfitting Mechanisms

When the dataset is small, models are more likely to memorize noise instead of learning real patterns. In this case, using degree 2 polynomial regression on 7 or 8 features with only 54 training samples almost guarantees overfitting. This risk is further increased by the *curse of dimensionality*: as the number of features grows, the data becomes sparse in the high-dimensional space, making it much harder for the model to find meaningful patterns and increasing the likelihood of fitting to noise instead(IBM 2025, overfitting-vs-underfitting).As a result, the model can easily fit the training data very well, but then fails to generalize to new, unseen data. This is seen in the results, where some models perform much better on the training set than on the test set. Overfitting is a major risk in machine learning, especially when the number of features is high compared to the number of samples.

Mulitple Testing Bias

Multiple testing bias happens when many models or feature combinations are tried on the same data. The more comparisons you make, the higher the chance that one model will seem to perform well just by luck, even if there is no real effect. In this study, testing different model types and feature sets increases the risk that some results look promising by coincidence. This can lead to overestimating the true predictive power of the models. To address this, larger datasets or statistical corrections would be needed, but these were not possible here.

7 Conclusion

This paper analyzed the effect of sentiment analysis and model complexity on their performance in predicting the percentage changes of stock market indices. It presented the dataset characteristics, explained the intentions behind the train/test split, and described the data preprocessing steps. Furthermore, the methodology for hyperparameter configuration, model selection, and evaluation was introduced.

Applicability & Limitations

The evaluation in Section 6 revealed critical flaws in the dataset size and the distributions within the train and test sets. Therefore, the statistical significance of the findings remains unclear. It was shown that linear regression does not significantly outperform the baseline approach, and polynomial regression only reliably showed results in the 3 day forecast.

Research Outcomes

Regarding the research question, the findings are not conclusive, and thus no significant results were obtained. However, the evaluation indicated that the more complex model performed better than the simple model and the baseline, despite the large differences in distribution between the train and test sets. Also it was shown, that the Sentiment Analysis feature only showed improvements on the Train set, while it lowered the results on the Test set. As discussed, this may be due to the hyperparameter choices. Moreover, since the press conference statement summarizes the current economic situation, the more complex model might already extract this knowledge from the prior changes in stock index prices combined with the interest rate and its change. This idea stems from the fact that the polynomial transformation of the features results in combined features, which theoretically should capture the relationship between these variables.

Future Directions

Given the growing importance of large language models and advances in neural language processing, topics like this research question are currently highly researched. Due to the limited conclusiveness of the results, improvements should be made. These improvements should include a much larger dataset, an adjusted feature engineering pipeline—especially for semantic analysis—and the use of a specifically fine-tuned large language model for the analysis. It should also be tested whether simply summing the sentiment scores of the press release provides better analysis. As this was just a basic approach, future projects should

directly work with neural networks such as long short-term memory recurrent networks or transformer networks. Additionally, it should be considered whether individual parts of the full press release, or combinations thereof, provide better sentiment results than the full document.

It is also worth noting that the approach of Christoph Kilian Theil, Samuel Broscheit, and Heiner Stuckenschmidt in their paper *PRoFET: Predicting the Risk of Firms from Event Transcripts* already provides a working architectural framework to combine textual features with financial features.(Theil, Broscheit, and Stuckenschmidt 2019) Therefore it should be a direct inspiration for any further attempts of explaining the Relationship of Interest Rates and Their Predictive Power on the Stock Market.

Appendix A: Dataset

European Central Bank: `https://data.ecb.europa.eu/`

yahoo Finance: `https://finance.yahoo.com`

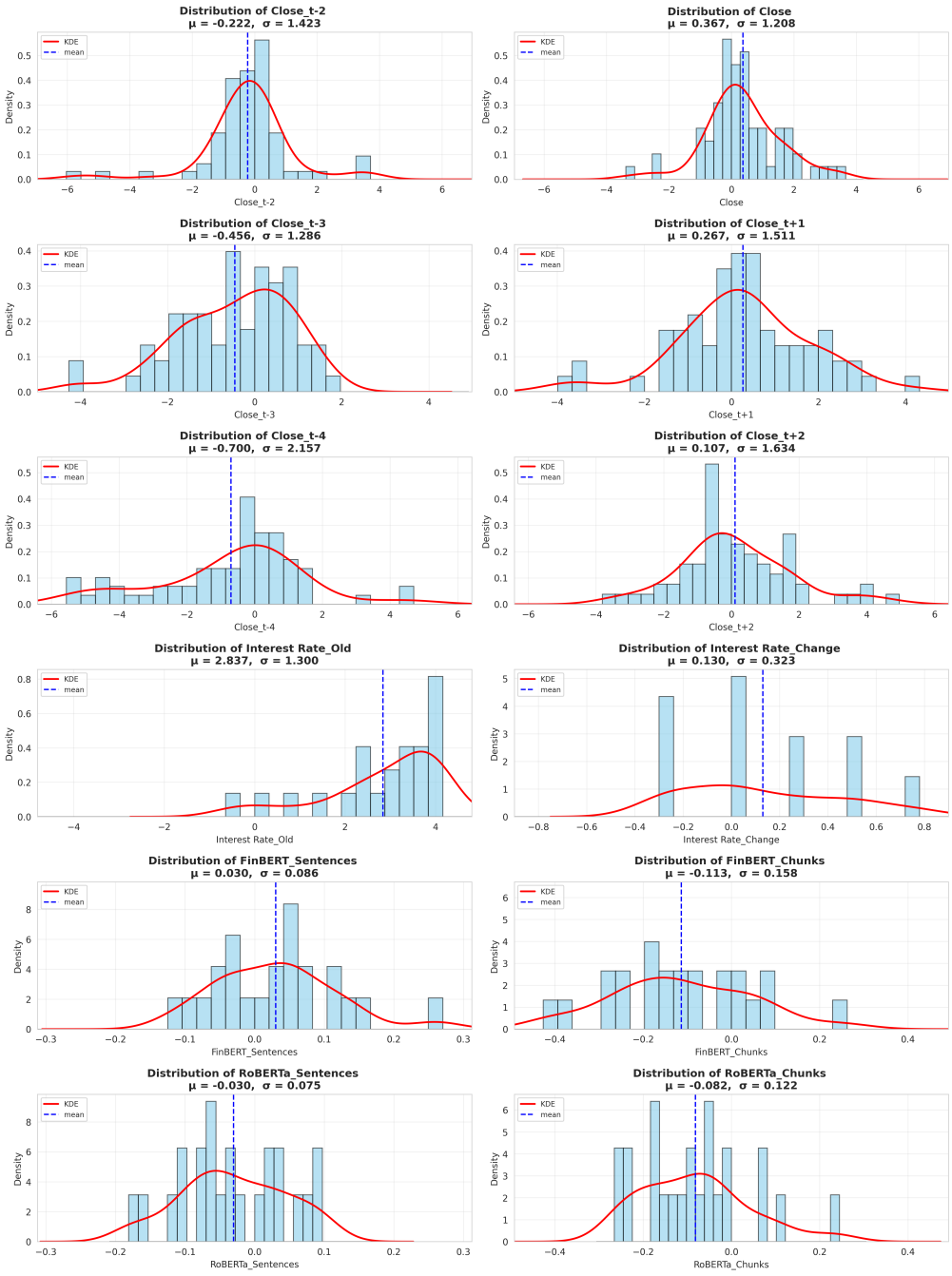

Figure 1: Distribution of the dataset variables showing the spread and frequency of each feature.

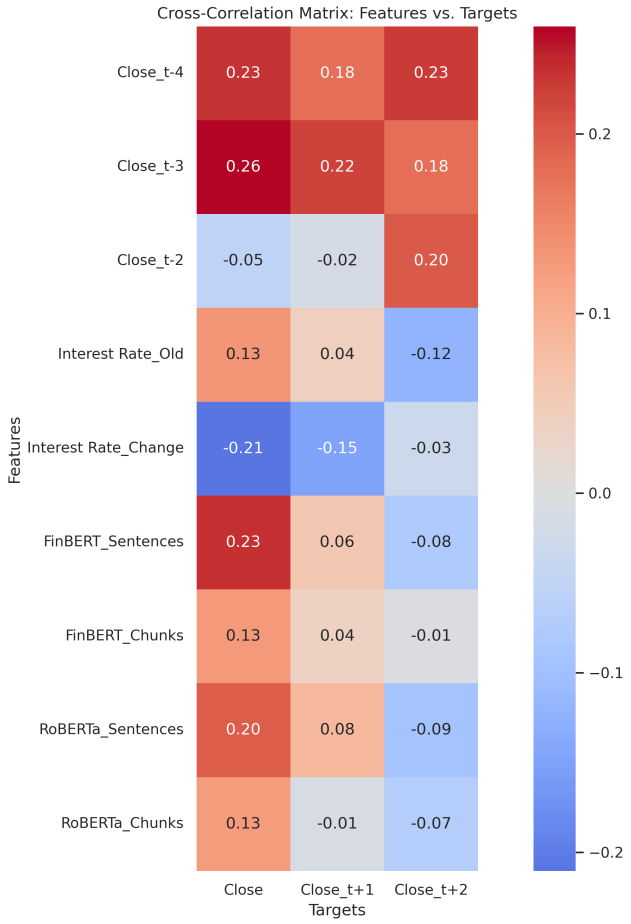

Figure 2: Heatmap illustrating the cross-correlation between features and target variables within the dataset, highlighting relationships and dependencies.

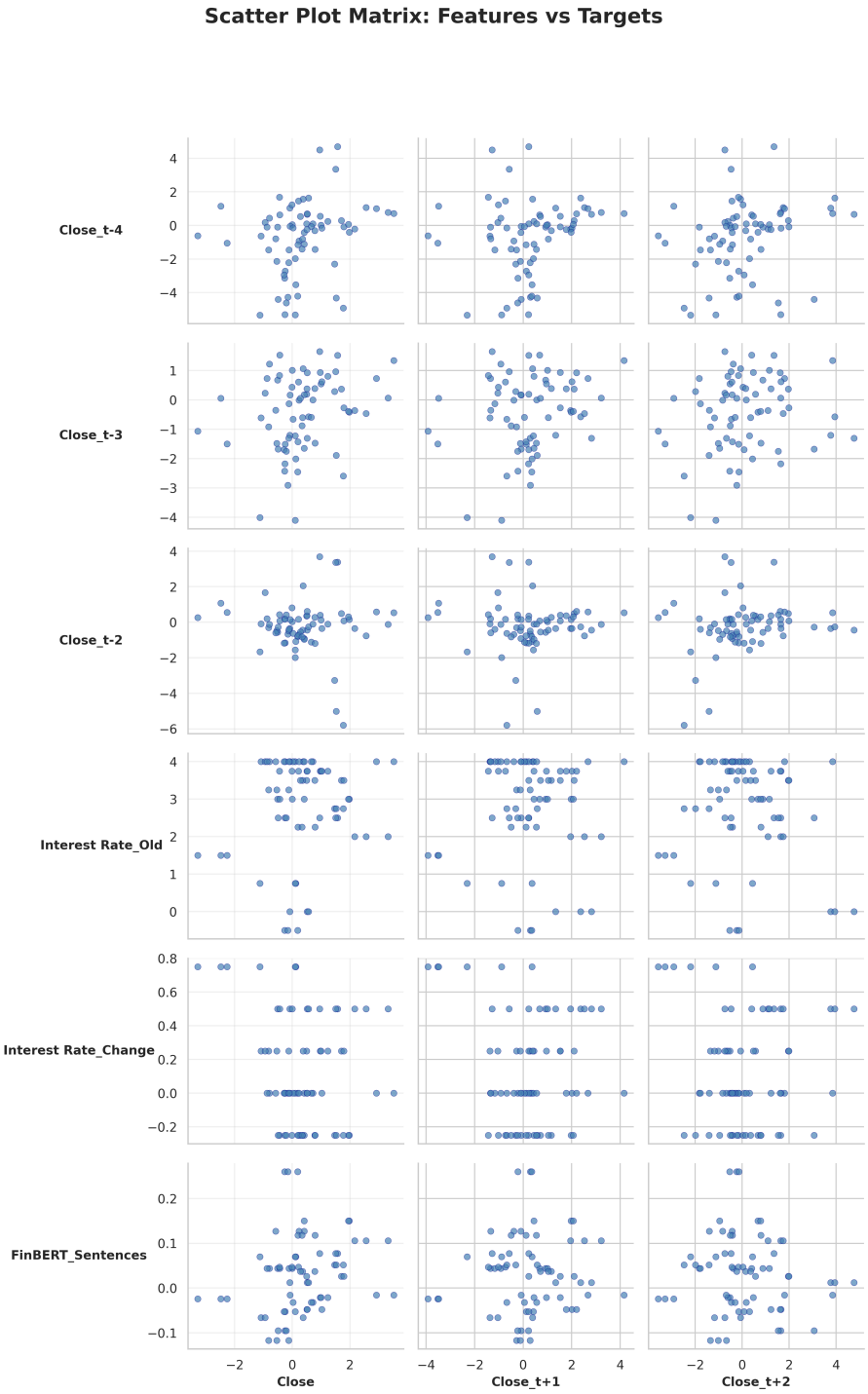

Figure 3: Pair plot visualizing the relationships and distributions between features and target variables in the dataset.

Appendix B: Train/Test Split Visualizations

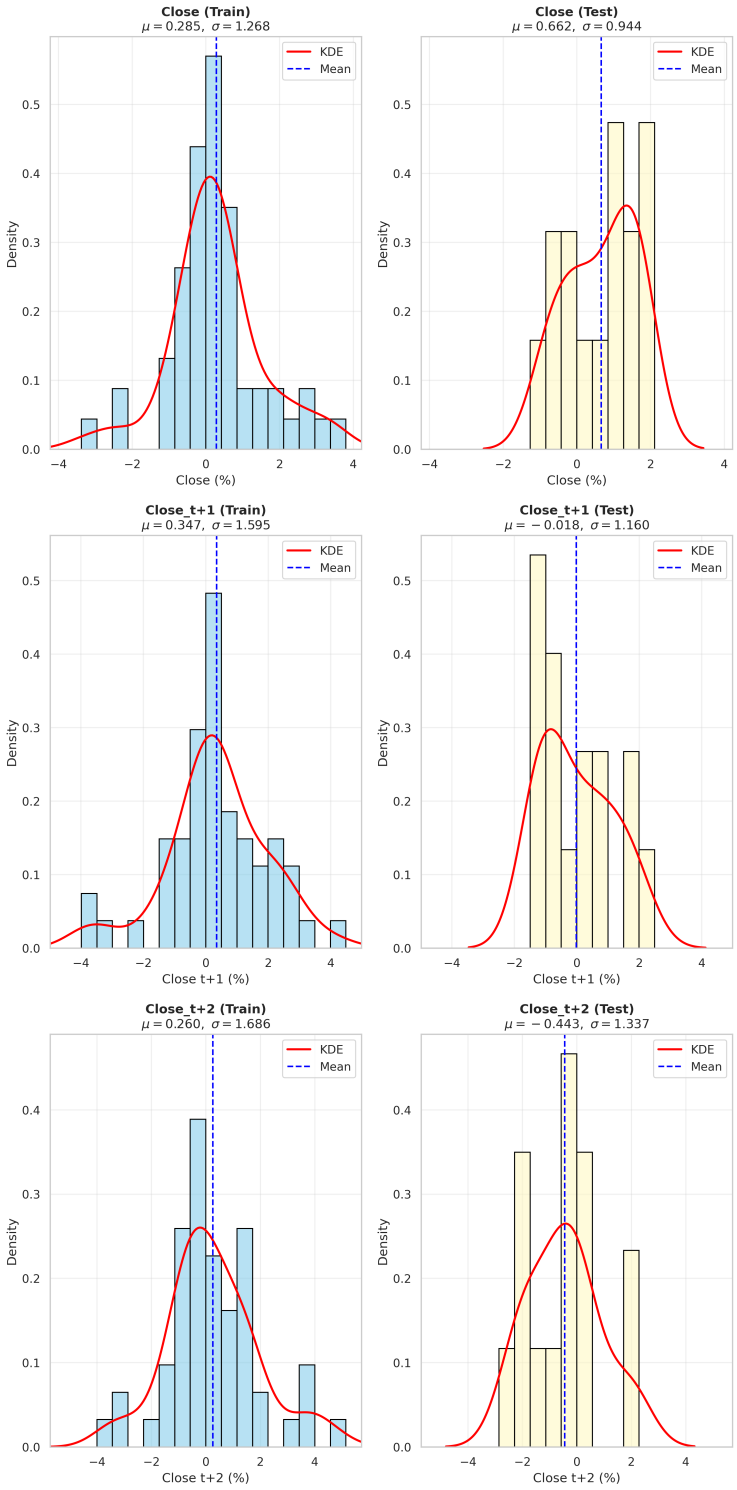

Figure 4: Distribution of the target variables in the train and test split, showing how the data is divided.

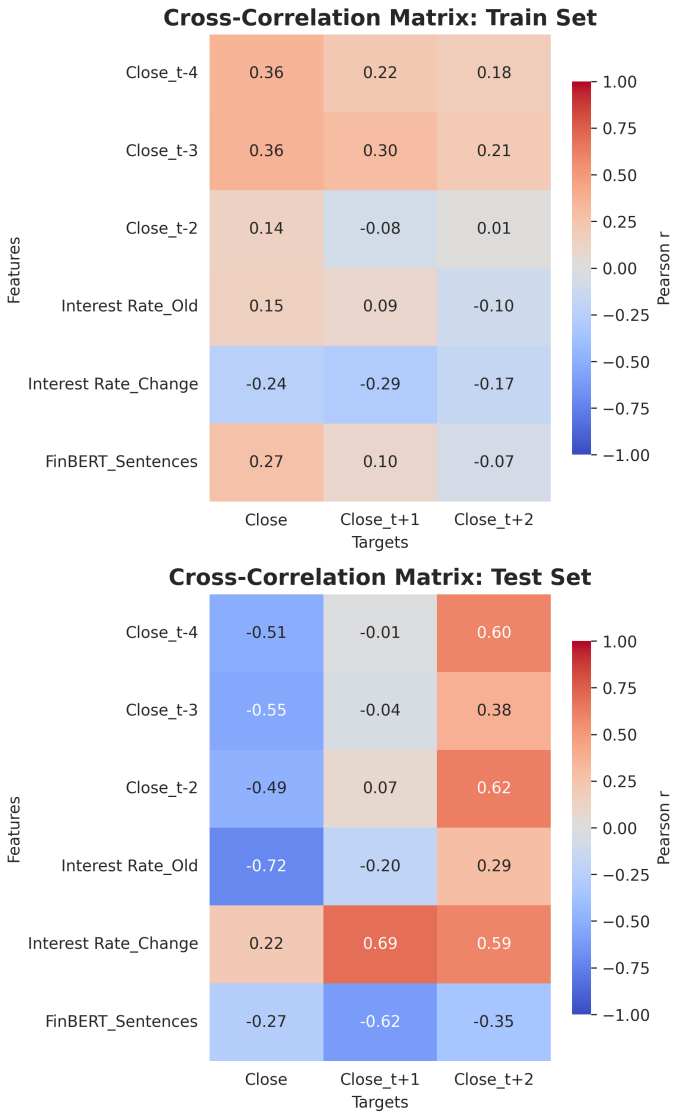

Figure 5: Heatmap showing the cross-correlation between features and targets in the train/test split dataset, useful for understanding variable interactions.

Appendix C: Model Results

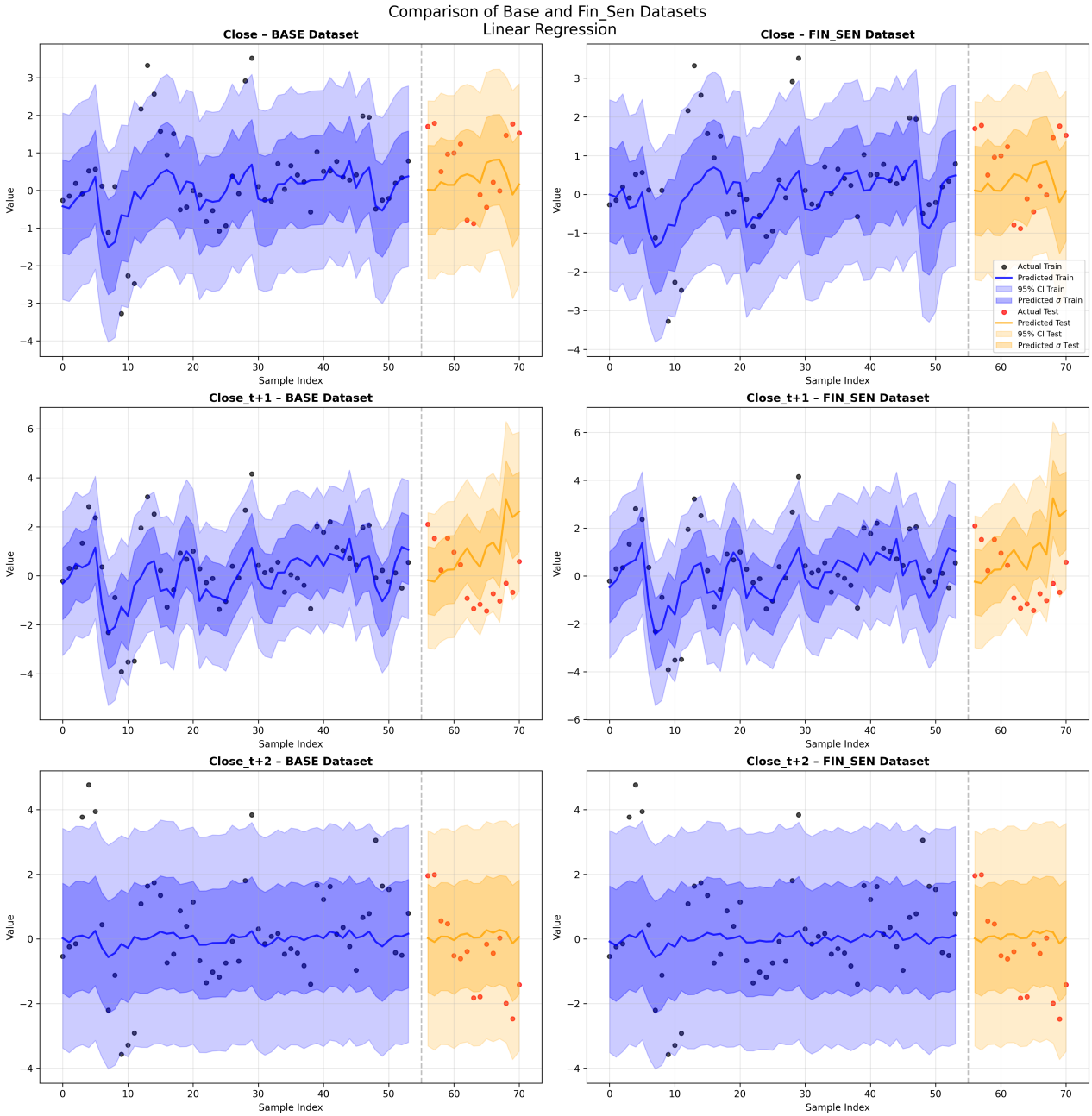

Figure 6: Comparison of linear Bayesian Ridge model performance, showing prediction lines and estimated standard deviation (σ).

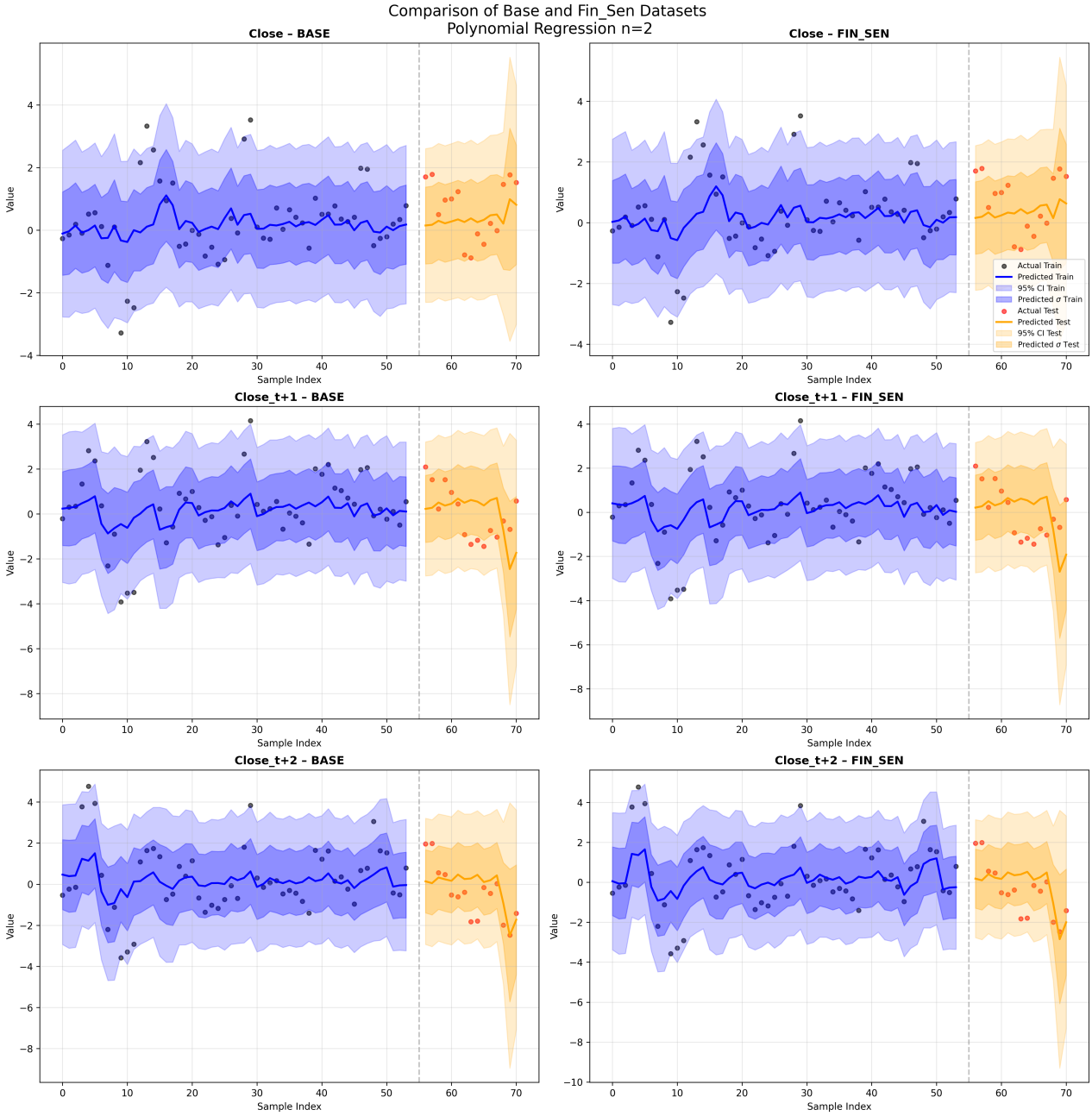

Figure 7: Comparison of polynomial Bayesian Ridge model performance, showing prediction lines and estimated standard deviation (σ).

Appendix D: Feature Importance Matrices

1 day Prediction

Standard		with Sentiment Analysis	
Feature	**Weight**	**Feature**	**Weight**
Close_t-4	0.304	FinBERT_Sentences	0.276
Interest Rate_Change	-0.303	Close_t-4	0.248
Close_t-3	0.231	Interest Rate_Change	-0.231
Index_SDAX	0.117	Close_t-3	0.216
Interest Rate_Old	-0.098	Index_SDAX	0.111

2 days Prediction

Standard		with Sentiment Analysis	
Feature	**Weight**	**Feature**	**Weight**
Close_t-3	0.995	Close_t-3	1.040
Close_t-2	-0.728	Close_t-2	-0.768
Interest Rate_Old	-0.612	Interest Rate_Old	-0.726
Interest Rate_Change	-0.610	Interest Rate_Change	-0.666
Close_t-4	0.264	Close_t-4	0.292

3 days Prediction

Standard		with Sentiment Analysis	
Feature	**Weight**	**Feature**	**Weight**
Close_t-3	0.114	Interest Rate_Change	-0.112
Interest Rate_Change	-0.112	Close_t-3	0.110
Interest Rate_Old	-0.100	Interest Rate_Old	-0.105
Close_t-4	0.087	Close_t-4	0.086
Index_SDAX	0.033	FinBERT_Sentences	-0.049

Table 8: Top six feature importances (by absolute weight) for the linear regression models, shown separately for one-, two-, and three-day prediction horizons. Results are presented side-by-side for the standard feature set and for the feature set including sentiment analysis.

1 day Prediction

Standard		with Sentiment Analysis	
Feature	**Weight**	**Feature**	**Weight**
Interest Rate_Change	-0.089	FinBERT_Sentences	0.111
Close_t-4	0.088	Close_t-4	0.103
Close_t-3	0.087	Interest Rate_Change	-0.102
Index_SDAX^2	0.055	Close_t-3	0.102
Interest Rate_Change^2	-0.043	Index_SDAX^2	0.065
Close_t-2 Interest Rate_Change	-0.038	Interest Rate_Change FinBERT_Sentences	0.059

2 days Prediction

Standard		with Sentiment Analysis	
Feature	**Weight**	**Feature**	**Weight**
Close_t-3	0.157	Close_t-3	0.158
Interest Rate_Change	-0.137	Close_t-4	0.133
Close_t-4	0.130	Interest Rate_Change	-0.132
Close_t-2 Interest Rate_Change	-0.118	Close_t-2 Interest Rate_Change	-0.113
Index_SDAX^2	0.103	Index_SDAX^2	0.104
Interest Rate_Old^2	0.094	Interest Rate_Change FinBERT_Sentences	0.100

3 days Prediction

Standard		with Sentiment Analysis	
Feature	**Weight**	**Feature**	**Weight**
Close_t-4 Interest Rate_Old	-0.177	Interest Rate_Old^2	0.199
Close_t-4 Interest Rate_Change	0.157	Interest Rate_Change FinBERT_Sentences	0.188
Close_t-2 Interest Rate_Change	-0.142	Close_t-4 Interest Rate_Change	0.157
Close_t-4^2	0.140	Close_t-4 Interest Rate_Old	-0.147
Close_t-3	0.134	Interest Rate_Change	-0.135
Interest Rate_Old^2	0.131	Close_t-3	0.132

Table 9: Top six feature importances (by absolute weight) for the polynomial regression models, for one-, two-, and three-day prediction horizons. Standard models use only financial features; models "with Sentiment Analysis" include the FinBERT sentiment score. Results are shown side-by-side for direct comparison.

Appendix E: Source Code

Git directory: `https://github.com/Aaron-7617171/Datascience`

Appendix F: Sentiment Analysis Outputs

Date	FinBERT_Sentences	FinBERT_Chunks	RoBERTa_Sentences	RoBERTa_Chunks
09_June_2022	0.196	0.119	0.036	-0.069
21_July_2022	0.260	0.245	0.094	0.231
08_September_2022	0.012	-0.006	-0.174	-0.245
27_October_2022	0.070	0.096	-0.058	-0.004
15_December_2022	-0.024	-0.195	-0.102	-0.140
02_February_2023	0.106	0.003	0.059	0.069
16_March_2023	0.077	0.030	-0.105	-0.073
04_May_2023	0.044	0.088	-0.015	-0.229
15_June_2023	-0.117	-0.270	-0.167	-0.240
27_July_2023	0.026	-0.018	0.016	-0.085
14_September_2023	-0.021	-0.126	0.038	-0.175
26_October_2023	-0.066	-0.142	-0.114	-0.161
14_December_2023	-0.016	-0.255	-0.061	-0.178
25_January_2024	-0.053	-0.185	-0.032	-0.178
07_March_2024	-0.032	-0.230	0.015	-0.027
11_April_2024	0.044	-0.191	0.091	-0.056
06_June_2024	0.127	-0.084	0.081	0.119
18_July_2024	0.047	-0.113	-0.074	-0.255
12_September_2024	-0.048	-0.387	-0.042	-0.055
17_October_2024	0.037	-0.283	-0.080	-0.108
12_December_2024	0.092	-0.103	0.058	0.133
30_January_2025	0.150	0.059	0.002	-0.020
06_March_2025	0.052	-0.141	-0.064	-0.091
17_April_2025	-0.095	-0.410	-0.036	0.064
05_June_2025	0.118	-0.095	0.031	-0.043
Average	0.039	-0.104	-0.024	-0.073

Table 10: Full sentiment analysis outputs for all ECB press conference dates.

Bibliography

ECB (June 30, 2025). *European Central Bank*. URL: https://www.ecb.europa.eu/ecb-and-you/explainers/html/monetary_policy.en.html.

Géron, Aurélien (2023). *Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme*. Trans. by Kristian Rother and Thomas Demmig. 3., aktualisierte und erweiterte Auflage. Heidelberg: O'Reilly. 1 p. ISBN: 978-3-96009-212-4.

IBM (July 2, 2025). *International Business Machines Corporation (IBM)*. URL: https://www.ibm.com/think.

Scikit-Learn (June 30, 2025). *Scikit Learn*. URL: https://scikit-learn.org/stable/.

Theil, Christoph Kilian, Samuel Broscheit, and Heiner Stuckenschmidt (Aug. 2019). "PRoFET: Predicting the Risk of Firms from Event Transcripts". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. Twenty-Eighth International Joint Conference on Artificial Intelligence {IJCAI-19}. Macao, China: International Joint Conferences on Artificial Intelligence Organization, pp. 5211–5217. ISBN: 978-0-9992411-4-1. DOI: 10.24963/ijcai.2019/724. URL: https://www.ijcai.org/proceedings/2019/724 (visited on 06/29/2025).

Appendix: Declaration on the Use of Generative AI Systems

In creating this work, I have used the following artificial intelligence (AI) based systems:

1. Perplexity
   (a) Claude 3.7
   (b) Claude 4.0
   (c) GPT-4.1
2. ChatGPT
   (a) GPT-3.5
   (b) GPT-4o

I further declare that I have:

☒ actively informed myself about the capabilities and limitations of the above-mentioned AI systems,

☒ marked the passages, which were generated completely by the above-mentioned AI systems,

☒ verified that the content generated with the help of the above-mentioned AI systems and adopted by me is factually correct,

☒ am aware that as the author of this work, I bear responsibility for the information and statements made in it.

I have used the above-mentioned AI systems as described below:

Work Step	**AI System(s) Used**	**Description of Usage**
Literature search	—	—
Literature analysis	—	—
Literature management and citation management	—	—
Selection of methods and models	ChatGPT, Perplexity	Consulted ChatGPT about analysis methods; searched for examples via Perplexity; selected method myself.
Data collection and analysis	Perplexity	Suggestions for Data Sources
Generation of program codes	ChatGPT, Perplexity	Used ChatGPT to create large formulas in LaTeX more quickly; used Perplexity for programming (few-shot coding, explanation of code parts, help with assembling program parts)
Creation of visualizations	—	—
Interpretation and validation	—	—
Structuring the text of the work	ChatGPT, Perplexity	Suggestions for outline structure
Formulation of the text of the work	ChatGPT, Perplexity	Reformulation of unsatisfactory paragraphs. Spell checking.
Translation of the text of the work	Perplexity	Translation of this Declaration into English, translation of my own german phrases into english
Editing the text	—	—
Other	—	Generated Text: 3.2 Sentiment Analysis; In Chapter 2:Theoretical Background (Efficient Market Hypothesis, Polynomial Regression); 6.1 Error Analysis: Mulitple Testing Bias

Hamburg, 30.06.2025

Aaron Gresser