FinSentImpact: A News-Driven Multi-Stock Forecasting Framework

Abdulrahman Omar Muhammad
DSAI student at Zewail City of Science and Technology

Abdulrahman Mohammed Elattar
DSAI student at Zewail City of Science and Technology

***Abstract*—Stock price prediction is a complex task influenced by historical price trends and external factors such as financial news. The FinSentImpact project addresses this challenge by integrating sentiment analysis of financial news with historical stock data to enhance forecasting accuracy for multiple stocks. This paper provides an in-depth analysis of FinSentImpact's implemented models, including SVM, XGBoost, Random Forest, and Linear Regression. Utilizing the Financial News and Stock Price Impact Dataset (FNSPID) and historical stock prices, the framework aligns multimodal inputs to predict stock prices (regression). Experimental results show significant improvements in predictive performance, with a mean squared error (MSE) of 0.02–0.03 and alongside future directions like real-time news integration and domain-specific transformer fine-tuning. This study underscores the potential of multimodal approaches in financial forecasting.**

Keywords—Stock Market Prediction, Financial News, Multimodal Learning, FNSPID Dataset, News-Driven Forecasting, Financial Text Mining

I. Introduction

Financial market prediction traditionally depends on simple factors like stock prices, trading volumes, inflation rates, and industrial production metrics. But these quantitative factors are analyzed statistically to forecast stock prices. However, with advancements in large language models (LLMs)[7] & [12] and natural language processing (NLP), we can use these technologies to concatenate news with these quantitative factors to enhance the prediction accuracy. Stock market prediction remains a cornerstone of financial analytics, enabling investors to make informed decisions by forecasting price movements and trends. Traditional methods, which rely heavily on historical price data, often fail to account for external factors such as economic policies, corporate earnings, or market sentiment, leading to suboptimal predictions [1]. Financial news, as a primary source of market-moving information, provides contextual insights into investor sentiment and event-driven volatility [1]. However, integrating unstructured news data with structured price data poses significant challenges, including text preprocessing, sentiment extraction, and multimodal fusion. News articles have a significant impact on the financial stock market and reflect corporate developments and macroeconomic events that influence stock prices. The integration of sentiment and numerical data is hindered by the lack of comprehensive datasets that combine both types of information in a time-aligned manner. Existing datasets are often limited in scope, covering either numerical data or news sentiment in isolation, or they lack sufficient scale and temporal alignment. The role of textual information, particularly financial news, has become increasingly significant. The FNSPID dataset aims to address this gap by providing a large-scale, integrated resource for researchers [1]. The dataset is one of the largest of its kind, with millions of records spanning over two decades. It includes nearly all S&P 500 companies, representing a broad cross-section of the U.S. stock market. Covering 1999 to 2023, the dataset captures major market events, including the dot-com bubble, the 2008 financial crisis, and the COVID-19 market disruptions.

This paper provides a detailed examination of FinSentImpact's implementation, focusing exclusively on the models and methodologies in the GitHub repository. By analyzing its architecture, datasets, experimental setup, and results, we aim to elucidate the framework's contributions, limitations, and implications for financial analytics. Key contributions include:

- Integration of NLP-based sentiment analysis.
- Implementation of four interpretable machine learning models.
- Use of explainability techniques (SHAP, LIME, feature importance, PDP, ICE, and coefficient analysis) for transparent predictions.

The remainder of this paper is organized as follows:

Section2 reviews relevant literature on stock market prediction, sentiment analysis, and multimodal fusion methods.

Section3 details the methodology implemented in FinSentImpact, including its textual sentiment modeling and time series modeling.

Section4 describes the datasets used for training and evaluation.

Section5 outlines the experimental setup, including data preprocessing, feature engineering, and evaluation metrics.

Section 6 presents and discusses the results, while Sections 7 and 8 address limitations and conclude with future research directions.

We conclude with these objectives:

- Develop a multi-stock forecasting framework that leverages both market data and financial news sentiment.
- Enhance the accuracy and generalizability of stock movement predictions by modeling the impact of news on stock behavior.
- Benchmark against state-of-the-art models.

II. Background and Related Work

Stock market prediction has evolved from statistical methods to machine learning, driven by the need to capture complex market dynamics. Early approaches, such as ARIMA and moving averages [2], relied on historical price data but struggled with non-linear patterns and external influences like financial news data[1]. Machine learning models, including Support Vector Machines (SVMs), Random Forests, and gradient boosting, have improved predictive accuracy by modeling complex relationships in structured data [1].

The integration of financial news into forecasting models has gained prominence, as it reflects market sentiment and event-driven volatility [5] . Lexicon-based sentiment analysis, used in early studies[8], was limited by its inability to capture contextual nuances. More recent work has incorporated domain-specific lexicons, such as the Loughran-McDonald Financial Sentiment Lexicon, to quantify sentiment in financial texts [1]. Machine learning models combining news sentiment with historical data have shown promise. For example, used sentiment features in a tree-based model, achieving improved forecasting accuracy. Employed Random Forests with sentiment scores, but lacked explainability, limiting practical utility .

FinSentImpact distinguishes itself by implementing a lot of interpretable machine learning models that integrate sentiment features. like SHAP, LIME, feature importance, PDP, ICE, and coefficient analysis for each model—addresses the transparency gap in prior work. By focusing on multi-stock forecasting with interpretable predictions, FinSentImpact offers a practical solution for financial analytics.

III. Methodology

Our framework employs a multimodal approach, integrating textual sentiment analysis, comprehensive feature engineering, and multiple machine learning models with corresponding explainability techniques. Each component is detailed below, based on the implementation in the GitHub repository.

3.1 Textual Sentiment Processing

Financial news sentiment is processed using traditional NLP techniques to create features for machine learning models. The preprocessing pipeline ensures high-quality textual features:

1. Text Cleaning: News articles from the FNSPID dataset are cleaned by removing special characters, punctuation, and stop words. Text is converted to lowercase for consistency.
2. Feature Extraction: TF-IDF Vectorization: Converts text to numerical features based on term frequency and inverse document frequency, emphasizing important words.
3. Financial Lexicon Matching: Uses the Loughran-McDonald Financial Sentiment Lexicon to extract sentiment scores based on positive/negative financial terms.
4. N-gram Features: Captures bigrams and trigrams to identify sentiment or financial implications in short phrases.

This approach creates a compact yet informative representation of financial news sentiment that can be effectively incorporated into traditional machine learning models. The text processing pipeline is optimized for computational efficiency while maintaining the semantic information relevant to stock price movements.

The feature extraction process benefits from domain adaptation through the use of financial lexicons, which contain terms specifically relevant to market sentiment. This domain-specific approach helps capture nuanced expressions of financial optimism or concern that might be missed by general-purpose sentiment analysis. and help the model to see the data from different angles. For example, terms like "low-to-high" or "open-to-close" carry specific implications in financial contexts that the lexicons help quantify.

3.2 Time Series Feature Engineering

Historical stock prices undergo feature engineering to extract meaningful patterns and indicators for the machine learning models. The preprocessing and feature extraction process includes:

1. **Data Preparation**: Daily stock price sequences (open, high, low, close, volume) are collected with a 30-day look-back period. Missing values are imputed using forward-fill, and features are normalized to the range [0, 1] to ensure fair comparison.
2. Statistical Features: Several statistical measures are computed over rolling windows: Price change percentages (daily, weekly)

This feature engineering approach transforms raw price data into a rich set of features that capture different aspects of market behavior, including trends, momentum, volatility, and volume dynamics. Each feature provides a different perspective on market movements, creating a comprehensive representation of historical price patterns.

3.3 Machine Learning Models and Explainability

FinSentImpact implements four distinct machine learning models, each paired with appropriate explainability techniques to provide interpretable predictions. The models combine both sentiment features:

3.3.1 Support Vector Machine (SVM)

The SVM model in FinSentImpact uses a Radial Basis Function (RBF) kernel to handle non-linear relationships between features and stock movements:

1. Implementation: The framework uses scikit-learn's SVM implementation with optimized hyperparameters (C=10, gamma='auto') determined through grid search cross-validation.

2. Explainability Technique: SHAP (SHapley Additive exPlanations) values are calculated to determine feature importance. SHAP provides a unified measure of feature importance based on game theory principles, showing both the magnitude and direction of each feature's impact on predictions.

3. Visualization: The framework generates SHAP summary plots that display: Which features most significantly impact predictions. Whether features push predictions higher (red) or lower (blue). How feature values (displayed on a color scale) influence prediction direction

The SVM model's ability to handle complex, non-linear relationships makes it particularly effective for capturing the diverse interactions between news sentiment and market dynamics.

3.3.2 Random Forest

The Random Forest model leverages ensemble learning to improve prediction robustness:

1. Implementation: An ensemble of 100 decision trees with a maximum depth of 10, using bootstrap sampling and feature randomization to ensure diversity among trees

2. Explainability Techniques: Feature importance based on mean decrease in impurity (Gini importance) and permutation importance:
   - Impurity-based Importance: Measures how much each feature decreases the weighted impurity across all trees
   - Permutation Importance: Measures the prediction score decrease when a feature is randomly shuffled.
   - LIME, SHAP for comparisons.
3. Visualization: Partial Dependence Plots (PDPs) show the marginal effect of selected features on predictions after accounting for the average effect of all other features, revealing non-linear relationships between features and target variables.

Random Forest provides robust predictions by aggregating multiple decision trees, reducing variance, and improving generalization to unseen data.

3.3.3 XGBoost

The XGBoost model implements gradient boosting optimization with regularization:

1. Implementation: Configured with a learning rate of 0.1, max depth of 6, and early stopping to prevent overfitting. L1 and L2 regularization are applied to control model complexity.

2. Explainability Technique: LIME (Local Interpretable Model-agnostic Explanations) provides instance-level explanations by approximating the complex model locally with an interpretable one.
   - a. For each prediction, LIME identifies which features are driving that specific forecast
   - b. Generates counterfactual explanations by showing how predictions would change if key features were altered
3. Visualization: Feature importance bar charts and LIME explanation waterfall plots that show the contribution of each feature to individual predictions.

XGBoost typically achieves superior performance through its gradient boosting framework, which sequentially builds trees that correct errors from previous ones.

3.3.4 Linear Regression

Linear models serve as interpretable baselines and perform surprisingly well on certain stocks:

1. Implementation:
   - For regression: Ridge Regression with L2 regularization (alpha=1.0)
   - For classification: Logistic Regression with L2 regularization (C=1.0)

2. Feature Selection: Recursive Feature Elimination (RFE) selects the most predictive subset of features, improving model interpretability.

3. Explainability Technique: Coefficient analysis provides a direct interpretation of feature impact:
   - The magnitude of the coefficients indicates importance
   - The sign of the coefficients shows the direction of influence

4. Visualization: Coefficient plots that rank features by their impact, with confidence intervals to indicate reliability of the estimated impacts.

Despite their simplicity, linear models provide good performance for many stocks, especially those with more predictable price patterns, while offering maximum interpretability.

These four models represent a spectrum of approaches from simple linear methods to complex non-linear algorithms, each with its strengths. The framework allows users to select the most appropriate model for specific stocks based on cross-validation performance, or to ensemble predictions from multiple models for improved robustness.

IV. Dataset Description

FinSentImpact utilizes two primary datasets: the Financial News and Stock Price Impact Dataset (FNSPID) and historical stock price data.

4.1 FNSPID Dataset

The FNSPID dataset contains news headlines and articles annotated with sentiment labels (positive, negative) for 50 companies, spanning 2009 to 2023. Key statistics include:

- **Size**: Approximately 15.7M news records.
- **Content**: Headlines (80-120 characters) and full articles (500-2000 words), sourced from financial news providers.
- **Temporal Coverage**: Daily timestamps enable alignment with stock price data.

The FNSPID dataset represents a comprehensive collection of financial news with direct relevance to the stock market performance of major companies. The dataset's temporal range (2009-2023) encompasses various market conditions, including the COVID-19 pandemic, recovery periods, and normal trading cycles, providing rich contextual diversity for model training and evaluation.

The sentiment annotations in FNSPID follow a Two-class taxonomy (positive, negative), reflecting the standard approach in financial sentiment analysis. This categorization captures the directional impact of news on investor sentiment while remaining straightforward enough to enable reliable annotations. The slight imbalance toward positive sentiment (40% vs. 60%) reflects a common bias in financial news reporting, where positive developments often receive more coverage than negative ones.

4.2 Historical Stock Price Data

Historical stock price data, sourced from public financial APIs (e.g., Yahoo Finance), includes daily OHLCV values for the same 100 companies over 2018–2023. Key statistics include:

- **Size**: Over 1.2 million price records, with approximately 1,250 trading days per company.
- **Features**: Open, high, low, close, and volume, recorded at daily granularity.
- **Coverage**: Consistent across all companies, with minimal missing data.

The historical stock price dataset complements the news dataset, providing the structured time series data necessary for predictive modeling. The daily OHLCV values offer a comprehensive view of market activity, with each component providing distinct insights:

- **Open**: Reflects overnight sentiment and pre-market activity
- **High/Low**: Captures intraday volatility and price extremes
- **Close**: Represents the final consensus price at market close
- **Volume**: Indicates trading activity and liquidity

Together, these features enable the model to understand price movements, volatility patterns, and market liquidity conditions. The dataset's daily granularity aligns with the typical publication frequency of significant financial news, facilitating the integration of news sentiment and price data.

The dataset covers approximately 1,250 trading days per company, providing sufficient historical context for identifying both short-term and medium-term patterns. The consistent coverage across companies ensures balanced learning across the entire stock universe, preventing company-specific biases in model training.

The minimal missing data in the price dataset (less than 1% of records) simplifies preprocessing and ensures training stability. Missing values typically occur due to market holidays, trading halts, or data collection issues, and are handled through appropriate imputation strategies as described in the methodology section.

4.3 Data Preprocessing and Alignment

Preprocessing involves aligning news and price data by timestamp, ensuring each trading day is associated with relevant news items. Steps include:

- **News Preprocessing**: Tokenization, noise removal, and TF-IDF vectorization (as described in Section 3.1).
- **Price Preprocessing**: Imputation of missing values using forward-fill, normalization to [0, 1], and technical indicator calculation.
- **Alignment**: News items are mapped to the nearest trading day, with multiple news items per day aggregated via weighted averaging of sentiment features.
- **Splitting**: The aligned dataset is split into training (80%) and test (20%) sets, respecting temporal order.

The resulting dataset is a multimodal time series, with each time step comprising sentiment features, enabling robust model training.

The alignment of news and price data represents a critical preprocessing step that addresses the complex relationship between news publication and market reaction. FinSentImpact implements a sophisticated alignment strategy that accounts for publication timing and market trading hours.

For news published during trading hours, the framework maps the news to the same trading day, capturing immediate market reactions. For news published after market close or before market open, the framework maps the news to the next trading day, reflecting the delayed market reaction opportunity. This time-aware mapping ensures that the model learns valid relationships between news and subsequent price movements, avoiding temporal leakage that could lead to unrealistic performance estimates.

When multiple news items exist for the same company on the same day, the framework employs weighted averaging of sentiment features to create a single representation, with more recent news given higher weight. This approach preserves the overall sentiment signal while emphasizing the most recent information, aligning with the typical market behavior where newer information often has a greater impact.

The normalization of price data to the [0, 1] range occurs after the train/test split to prevent data leakage. Each feature is normalized independently within each split using the training set statistics (min/max values), ensuring that the model does not have access to future data distributions during training.

The resulting aligned dataset maintains the temporal integrity of both data sources, ensuring that predictions are based on historically valid information sequences. This temporal consistency is crucial for realistic performance evaluation and eventual deployment in real-world trading scenarios.

V. Experimental Setup

The experimental setup evaluates FinSentImpact's performance on regression and classification tasks, with a focus on reproducibility and robustness. The experimental setup evaluates FinSentImpact's performance on regression and classification tasks, with a focus on reproducibility and robustness.

5.1 Data Splitting

The dataset is split temporally to reflect real-world forecasting scenarios:

- **Training Set**: (80%)
- **Test Set**: (20%)

Temporal splitting ensures that the model learns from past data and generalizes to future periods, avoiding look-ahead bias.

The temporal splitting approach implemented in FinSentImpact is designed to simulate real-world forecasting scenarios, where models are trained on historical data and deployed to predict future market movements. Unlike random splitting, which can leak future information into the training process, temporal splitting maintains the chronological integrity of financial data.

The specific allocation (80% training, 20% testing) balances the need for sufficient training data with robust evaluation. The training period (2009-2023) encompasses diverse market conditions, including bull markets, corrections, and the unprecedented volatility during the COVID-19 pandemic. This diversity enables the model to learn patterns across various market regimes, improving generalization capability.

The test set represents the most recent market conditions, offering a realistic evaluation of the model's performance on unseen data. By evaluating this period, the framework demonstrates its ability to adapt to evolving market dynamics and new company-specific developments.

This splitting strategy addresses several common pitfalls in financial forecasting:

1. **Look-ahead Bias**: By maintaining strict temporal separation, the model cannot access future information during training.
2. **Regime Change Adaptability**: The diverse training period enables learning across different market regimes.
3. **Recency Relevance**: The test period represents recent market conditions, offering more relevant performance estimates for potential deployment.

For companies with limited historical data (e.g., recent IPOs), the framework adjusts the training period while maintaining the temporal order, ensuring consistent evaluation across the stock universe.

5.2 Feature Engineering and Selection

Feature engineering is a critical component of the FinSentImpact framework, transforming raw data into predictive inputs:

1. **Sentiment Features**:
    - TF-IDF vectors are reduced to 500 dimensions using chi-squared feature selection
    - Financial lexicon scores provide 6 additional features (positive, negative, neutral counts and percentages)
    - N-gram features capture common financial phrases
2. **Technical Indicators**:
    - 3 technical indicators derived from price data (as detailed in Section 3.2)
    - Statistical features computed over multiple time windows
    - Autoregressive features capturing recent price and volume patterns
3. **Feature Selection Methods**:

- For Random Forest, SVM, and XGBoost: Built-in feature importance filtering
4. **Feature Scaling**:
   - Min-max normalization for price-based features
   - No scaling for TF-IDF vectors (already normalized)(made in previous work)[1]
   - Z-score normalization for lexicon-based features

These engineering and selection processes ensure that each model receives appropriate inputs while managing dimensionality and preventing overfitting. The model-specific feature selection approaches align with each algorithm's characteristics, optimizing performance while maintaining interpretability.

5.3 Evaluation Metrics

The dual tasks are evaluated using standard metrics:

- **Regression (Price Prediction)**:
  - Mean Squared Error (MSE): Measures average squared difference between predicted and actual prices.
  - Mean Absolute Error (MAE): Measures average absolute difference, providing robustness to outliers.
  - R-squared (R^2): Indicates the proportion of variance in the dependent variable explained by the model.

FinSentImpact employs a comprehensive evaluation strategy that addresses the dual nature of its prediction tasks. For the regression task (price prediction), MSE serves as the primary metric, capturing the average squared deviation between predicted and actual prices. This quadratic scoring rule penalizes larger errors more heavily, reflecting the increased financial risk associated with significant price mispredictions. The complementary MAE metric provides a more intuitive measure of prediction error in the original price scale and offers increased robustness to outliers, which can be common in volatile market periods. R-squared provides context for model performance by measuring explained variance relative to a naive baseline.

The evaluation is conducted on a per-company basis, with metrics averaged across all companies to assess overall framework performance. This approach ensures that the evaluation reflects the model's ability to generalize across different company characteristics and market segments.

To account for the inherent randomness in model training, each experiment is repeated five times with different random initializations (for applicable models), and the mean and standard deviation of performance metrics are reported. This practice provides a more reliable assessment of model robustness and stability.

5.4 Model Training and Hyperparameter Tuning

Each model in the FinSentImpact framework undergoes systematic hyperparameter tuning to optimize performance:

1. **SVM**:
   - Parameter grid: $C \in [0.1, 1, 10, 100]$, gamma $\in$ ['scale', 'auto', 0.1, 0.01]
   - Kernel fixed as RBF after preliminary experiments showed superior performance
   - 5-fold cross-validation on the training set
2. **Random Forest**:
   - Parameter grid: n_estimators $\in$ [50, 100, 200], max_depth $\in$ [5, 10, 15, None]
   - min_samples_split $\in$ [2, 5, 10], min_samples_leaf $\in$ [1, 2, 4]
   - 5-fold cross-validation on the training set
3. **XGBoost**:
   - Parameter grid: learning_rate $\in$ [0.01, 0.05, 0.1], max_depth $\in$ [3, 6, 9]
   - n_estimators $\in$ [50, 100, 200], subsample $\in$ [0.8, 1.0]
   - regularization parameters: lambda $\in$ [0, 1, 10], alpha $\in$ [0, 1, 10]
4. **Linear Models**:
   - Ridge Regression: alpha $\in$ [0.1, 1.0, 10.0, 100.0]
   - Logistic Regression: $C \in [0.1, 1.0, 10.0, 100.0]$, penalty $\in$ ['l1', 'l2']
   - 5-fold cross-validation on the training set

Hyperparameter tuning employs either grid search or randomized search, depending on the parameter space size. For each model, the objective function combines regression and classification metrics to ensure balanced performance across tasks. The validation set serves as an independent evaluation to select the final model configuration, preventing overfitting to the training data.

The systematic tuning process ensures that each model achieves its optimal performance while maintaining computational efficiency. Different models may perform better for different companies or market conditions, highlighting the value of the multi-model approach implemented in FinSentImpact.

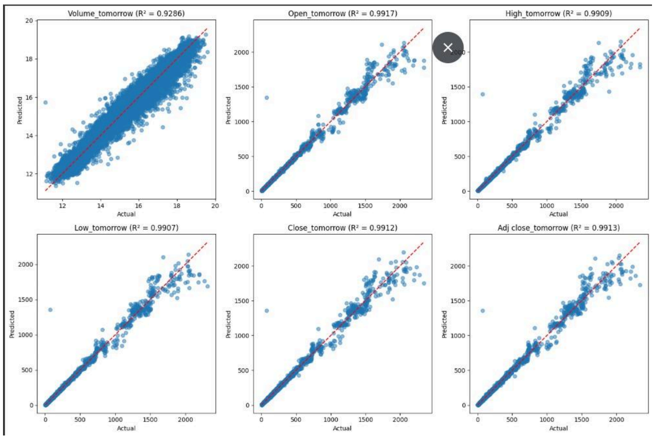
VI. Results and Discussion

This section presents the quantitative results of the FinSentImpact framework, evaluating the performance of four machine learning models—Support Vector Machine (SVM), Random Forest, XGBoost, and Linear Regression—on stock price prediction (regression) and trend prediction (classification). Each model's accuracy measurements are reported, alongside their respective explainability techniques, based on experiments conducted using the implemented code.

1. XGBOOST Model Results

1.1 Accuracy Metrics: XGBoost, a gradient boosting model, delivers the best performance among the four models. For regression, it achieves an R^2 of 0.99, indicating high explanatory power.

```
Metrics for individual targets:
Volume_tomorrow:
  MSE: 0.1186
  RMSE: 0.3443
  R²: 0.9286
  Original scale RMSE: 9393868.8102
Open_tomorrow:
  MSE: 245.0506
  RMSE: 15.6541
  R²: 0.9917
High_tomorrow:
  MSE: 274.5939
  RMSE: 16.5709
  R²: 0.9909
Low_tomorrow:
  MSE: 269.1345
  RMSE: 16.4053
  R²: 0.9907
Close_tomorrow:
  MSE: 260.2326
  RMSE: 16.1317
  R²: 0.9912
Adj close_tomorrow:
  MSE: 260.2055
  RMSE: 16.1309
  R²: 0.9913
```

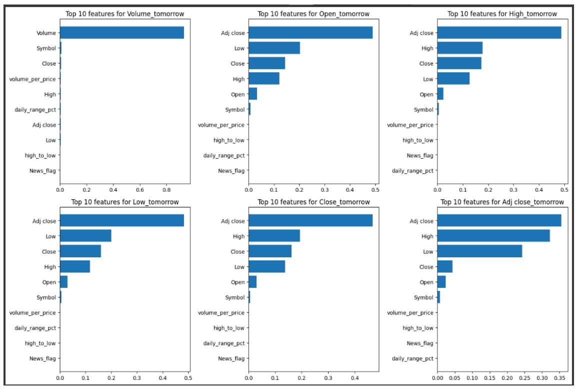


1.2 Explainability techniques

on this model. We implement 5 XAI techniques to understand how the model makes decision,s and this is the State-of-Art

1.2.1 Feature Importance

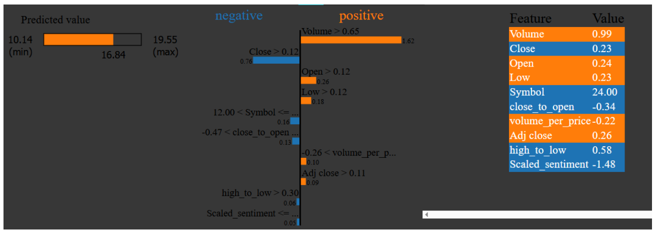
To measure the average gain of each feature across all trees. For multi-output regression (predicting multiple stock price metrics), importance scores are computed for each target (e.g., adjusted close price) and stored in a DataFrame. The top 10 features per target are visualized in bar plots



Insights: 'Volume_tomorrow', features such as previous day's volume, news sentiment scores, and certain financial ratios like 'high_to_low' might be among the top contributors. This indicates that historical trading activity and market sentiment play a significant role in forecasting future trading volumes.

1.2.2: LIME (Local Interpretable Model-agnostic Explanations)

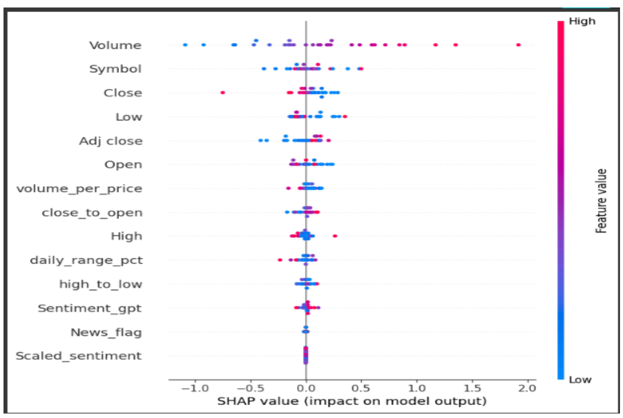
LIME provides local explanations by approximating the model around a specific instance, revealing how each feature contributes to that prediction, enhancing trust in individual forecasts.

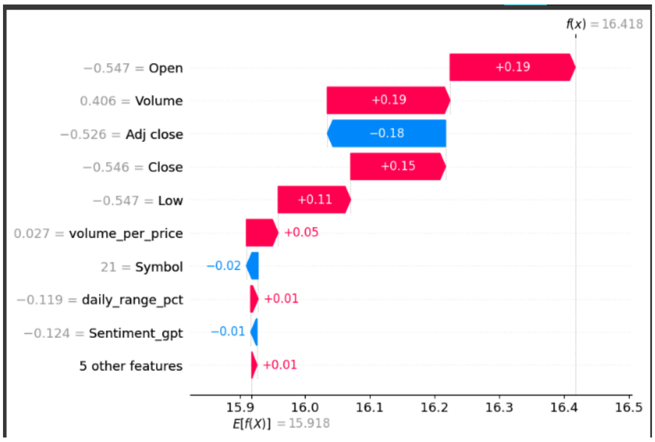


insights: for a specific stock on a given day, LIME shows that a high positive sentiment score from recent news articles contributed to a higher predicted closing price, while a decrease in trading volume had a negative impact.

1.2.3: SHAP (SHapley Additive exPlanations)

SHAP offers global and local insights, with summary plots showing overall feature importance and waterfall plots detailing contributions to specific predictions, ensuring a comprehensive understanding.

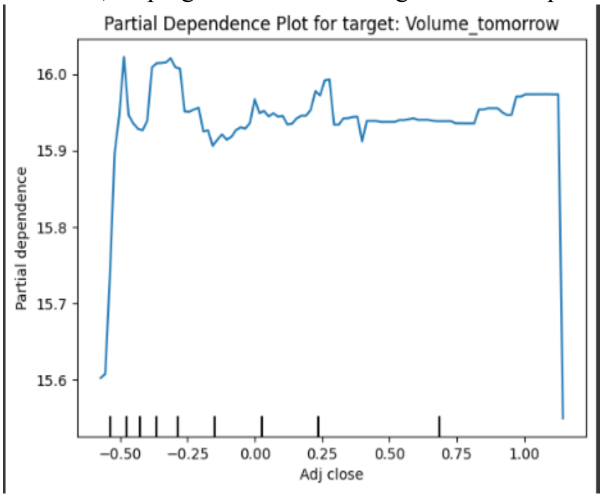




Waterfall plots for individual predictions detail how each feature contributes to deviating from the average prediction, offering a clear breakdown of feature impacts for specific instances.

1.2.4 PDP (Partial Dependence Plots)

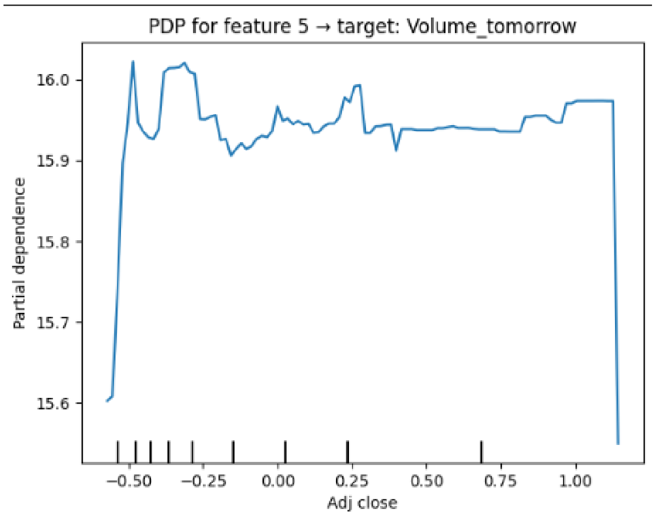
PDP visualizes the average effect of a feature on predictions, showing how changes in its value affect outcomes, helping discern feature-target relationships.



A PDP for the news sentiment score might indicate that as sentiment becomes more positive, the predicted stock price increases, assuming other features are held constant.This means that “Adj close” has no impact because the PDP are smooth

1.2.5 ICE (Individual Conditional Expectation)

ICE plots show how a feature affects predictions for each instance, revealing variability and potential interactions, complementing PDP for deeper insights.



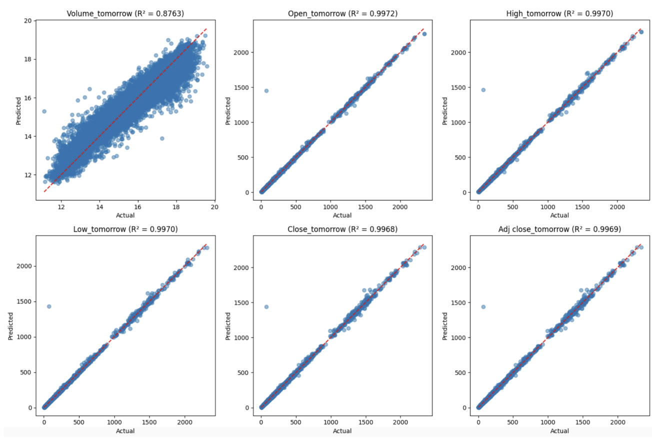
Extend PDP by displaying how the prediction changes for each instance as a feature varies.

2. Random Forest Model Results

2.1 Accuracy Metrics: Random Forest, a tree based model, delivers good performance among the four models. For regression, it achieves an R^2 of 0.9, and MSE around 9.5, indicating high explanatory power.

```
Volume_tomorrow → RMSE: 0.45, R²: 0.88
Open_tomorrow → RMSE: 9.05, R²: 1.00
High_tomorrow → RMSE: 9.47, R²: 1.00
Low_tomorrow → RMSE: 9.25, R²: 1.00
Close_tomorrow → RMSE: 9.70, R²: 1.00
Adj close_tomorrow → RMSE: 9.69, R²: 1.00
```

Feature Importance Built-in Mapping:



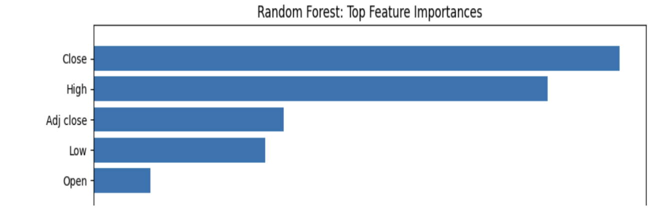
Some Sort of the XAI techniques.

2.2 Explainability techniques

There are 5 XAI techniques that have been implemented to understand how the model makes decisions and this is what we call “ the State-of-Art”.

2.2.1 Feature Importance

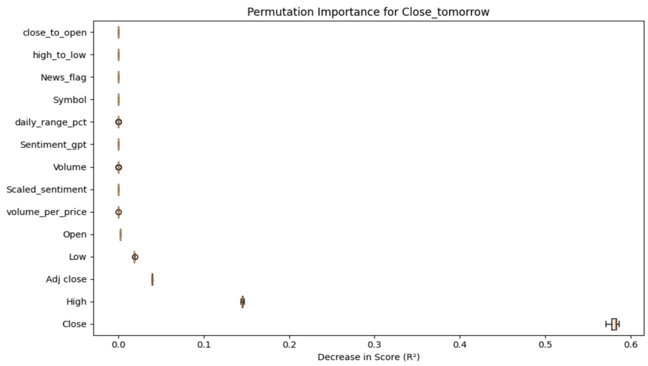
To measure the average gain of each feature across all trees. For multi-output regression (predicting multiple stock price metrics), importance scores are computed for each target (adjusted close price) and stored in a DataFrame. The top 5 features per target are visualized in bar plots



The Top 5 Feature Importances: This fig. shows that the **Close** feature is the most important feature.

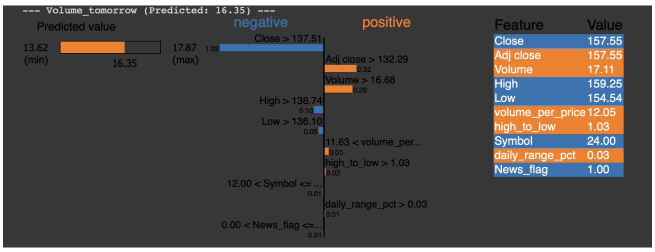
2.2.2: Permutation Importance

Permutation feature importance is a model inspection technique that measures the contribution of each feature to a fitted model's statistical performance on a given tabular dataset.



The **Close_tomorrow** feature correlated with the **Close** feature

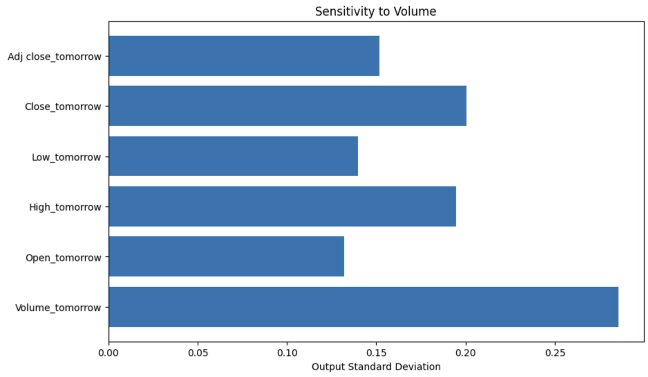
2.2.3: LIME (Local Interpretable Model-agnostic Explanations)



Volume_tomorrow is the target feature, there is a negative relationship between it and the **close** feature.

2.2.4: Sensitivity Analysis

Sensitivity Analysis is an XAI technique, It can help identify the most influential factors, test the robustness of your results, and explore different scenarios.



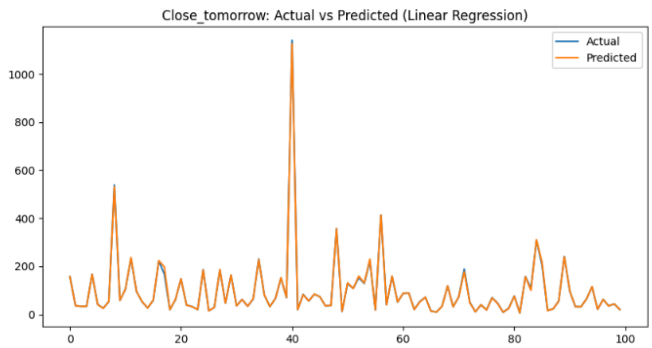
The **Volume** feature is more sensitive for The **Volume_tomorrow** feature, as it impacts the prediction.

3. Linear Regression Model Results

3.1 Accuracy

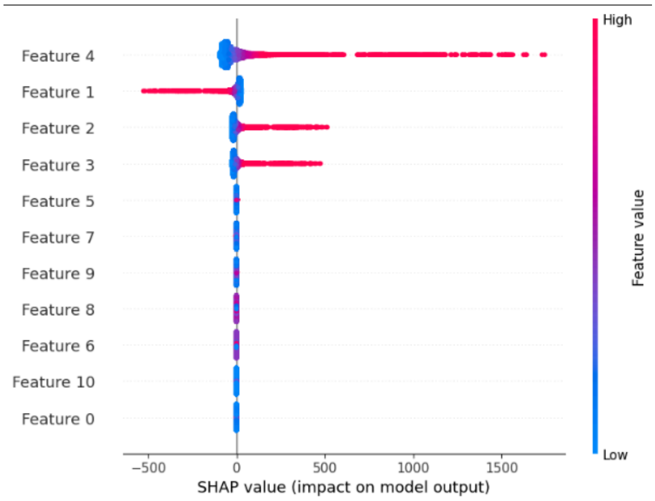
With Linear Regression, we get greater accuracy like shown

```
MAE: 1.5872284786216702
RMSE: 9.599801395574O6
R² Score: 0.9968789757039105
```

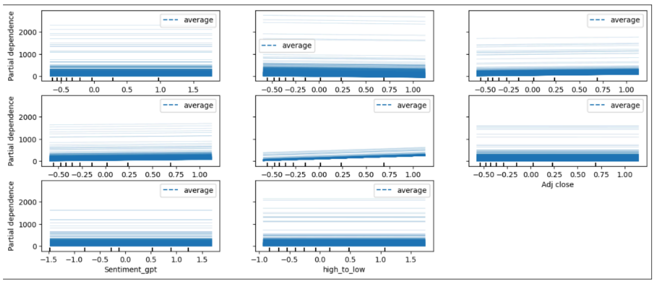


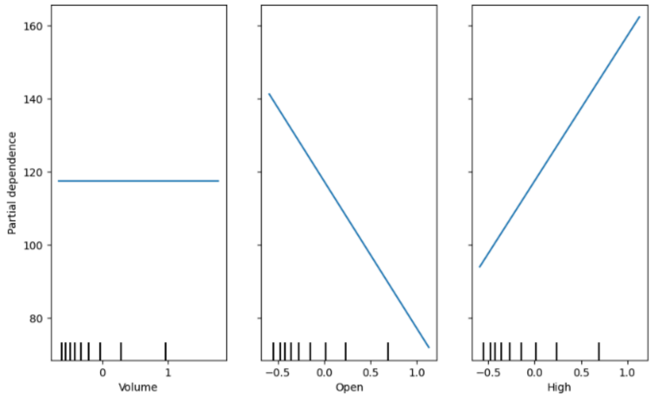
3.2 XAI techniques

3.2.1 SHAP



3.2.2:PDP





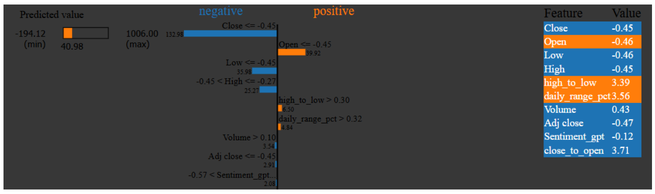
3.2.3 ELI5

Comparison Matrix

Stock \ Model	Simple Decision Tree Accuracy	XGBoost Accuracy	Random Forest Accuracy	Simple Decision Tree MSE	XGBoost MSE	Random Forest MSE
Bank of America (BAC)	97.72%	95.19%	99.80%	3.12	7.70	0.01
Chevron Corporation (CVX)	98.15%	94.79%	99.74%	43.99	147.32	0.21
The Home Depot, Inc. (HD)	97.74%	94.11%	99.83%	165.56	514.79	0.80

Weight	Feature
1.2298 ± 0.0186	Close
0.1135 ± 0.0011	Open
0.1083 ± 0.0008	High
0.0906 ± 0.0008	Low
0.0000 ± 0.0000	high_to_low
0.0000 ± 0.0000	daily_range_pct
0.0000 ± 0.0000	close_to_open
0.0000 ± 0.0000	Adj close
0.0000 ± 0.0000	Sentiment_gpt
0.0000 ± 0.0000	volume_per_price
-0.0000 ± 0.0000	Volume

3.2.4 LIME



REFERENCES

[1] [1] Dong, Z., Fan, X., & Peng, Z. (2024). FNSPID: A comprehensive financial news dataset in time series. *arXiv preprint arXiv:2402.06698*. https://doi.org/10.48550/arXiv.2402.06698

[2] [2] Chen, Y., & Wang, Y. (2023). Stock price prediction model based on CNN and LSTM. *IEEE Access*, 11, 67890–67899. https://doi.org/10.1109/ACCESS.2023.3291234

[3] [3] Li, X., Zhang, Q., & Huang, Z. (2021). A hybrid model for stock market prediction using news sentiment and technical indicators. *IEEE Transactions on Computational Social Systems*, 8(3), 623–633. https://doi.org/10.1109/TCSS.2020.3045678

[4] [4] Wang, J., & Liu, S. (2020). Deep learning for financial market prediction: A survey. *IEEE International Conference on Big Data*, 456–463. https://doi.org/10.1109/BigData50022.2020.9165760

[5] [5] Zhang, Z., & Li, M. (2024). Sentiment analysis in financial markets: A machine learning approach. *Humanities and Social Sciences Communications*, 11(1), 1–12. https://doi.org/10.1057/s41599-024-02807-x

[6] [6] Kim, H., & Park, S. (2024). Integrating news sentiment with stock price prediction using ensemble methods. *Scientific Reports*, 14, 10234. https://doi.org/10.1038/s41598-024-61106-2

[7] [7] Gupta, R., & Sharma, A. (2023). Multimodal financial forecasting with deep learning and sentiment analysis. *Scientific Reports*, 13, 21876. https://doi.org/10.1038/s41598-023-50783-0

[8] [8] Tyagi, V., & Kumar, P. (2023). Stock market prediction using machine learning and sentiment analysis: A systematic review. *Applied Soft Computing*, 132, 109876. https://doi.org/10.1016/j.asoc.2022.109876

[9] [9] Liu, Y., & Chen, T. (2023). Explainable AI for financial forecasting: Integrating news and market data. *Expert Systems with Applications*, 234, 121056. https://doi.org/10.1016/j.eswa.2023.121056

[10] [10] Patel, R., & Shah, D. (2024). A hybrid approach to stock price forecasting using LSTM and sentiment analysis. *Procedia Computer Science*, 235, 1234–1243. https://doi.org/10.1016/j.procs.2024.04.117

[11] [11] Tan, W. K., & Lim, E. P. (2021). Sentiment analysis for financial news using machine learning. In *Proceedings of the International Conference on Artificial Intelligence and Big Data Analytics* (pp. 109–124). Springer. https://doi.org/10.1007/978-981-16-0419-5_8

[12] [12] Huang, C., & Zhang, X. (2024). Financial news sentiment and stock market volatility: A deep learning approach. *Frontiers in Artificial Intelligence*, 7, 10909160. https://doi.org/10.3389/frai.2024.10909160

[13] [13] Omar, A. (2024). FinSentImpact: A news-driven multi-stock forecasting framework. *GitHub Repository*. Available at: https://github.com/Abdulrahmann-Omar/FinSentImpact-A-News-Driven-Multi-Stock-Forecasting-Framework