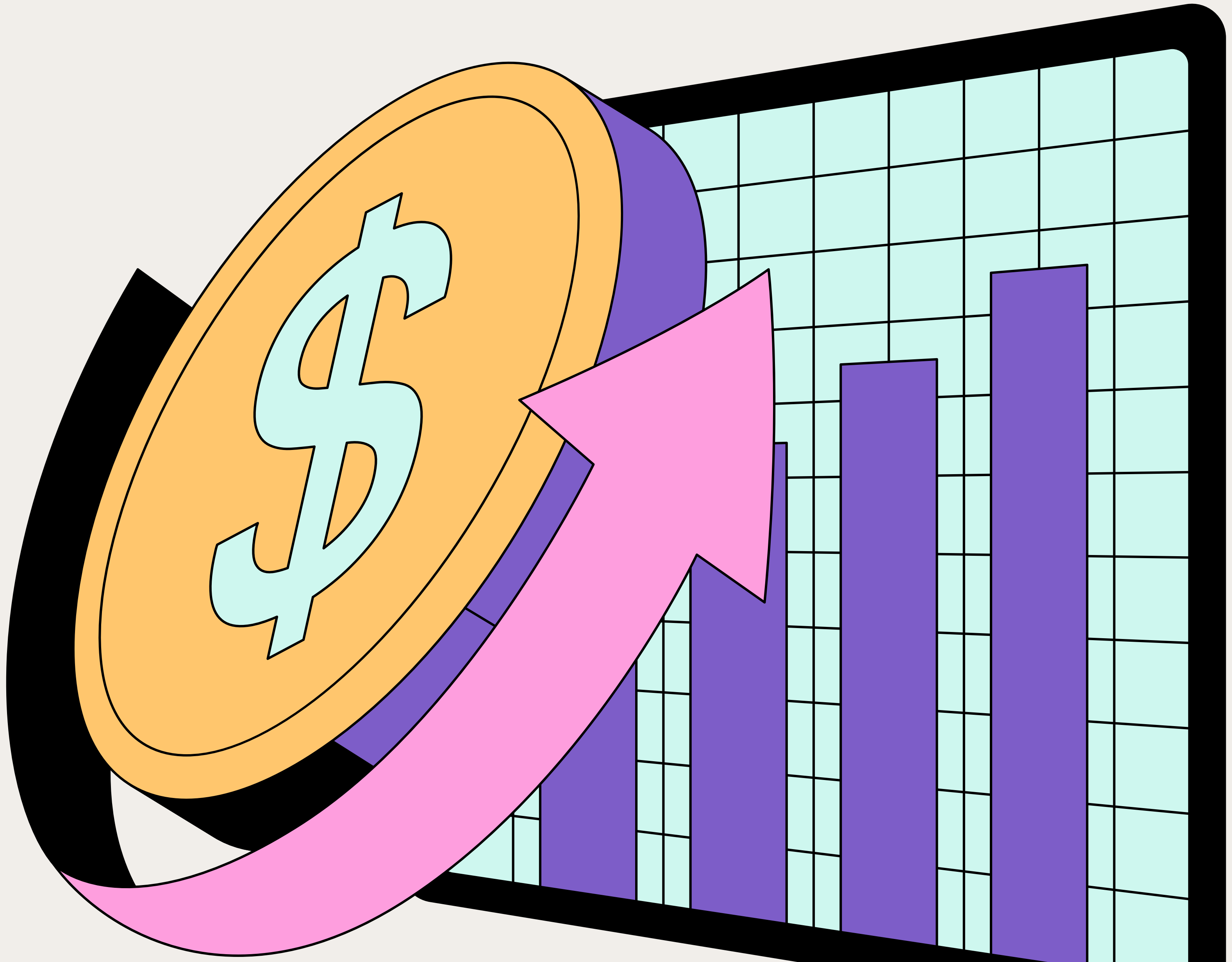
Understanding The

Stock Market

Stock Prices Prediction



DEVANSH PURSNANI
UID: 2023800091
BATCH:B DIVISION:B
BRANCH: CSE

OVERVIEW

ABSTRACT
ML predicts stock prices accurately

INTRODUCTION
Tech stock price forecasting project

DATASET
NASDAQ data with lag features

METHODOLOGY
Tree models beat linear regression

EXPERIMENTAL SETUP
Python tools with time-series validation

RESULTS AND DISCUSSION
Decision Tree wins

ERROR ANALYSIS
Struggles with market shocks

CONCLUSION AND REFERENCES
Reliable predictions for trading

INTRODUCTION

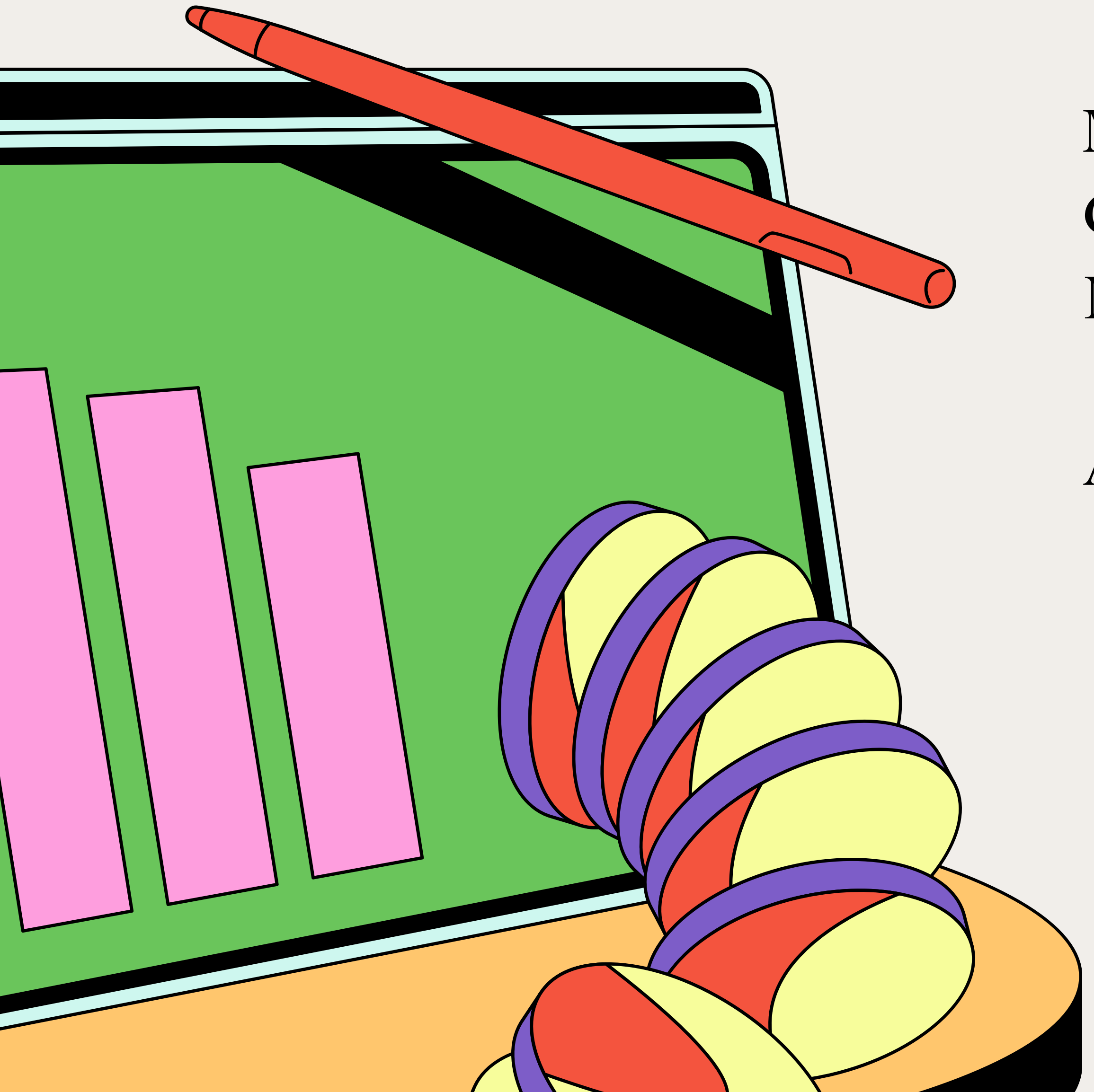
OBJECTIVE: PREDICT NEXT-DAY CLOSING PRICES FOR MAJOR TECH STOCKS (AAPL, GOOGL, MSFT, AMZN) USING HISTORICAL NASDAQ DATA.

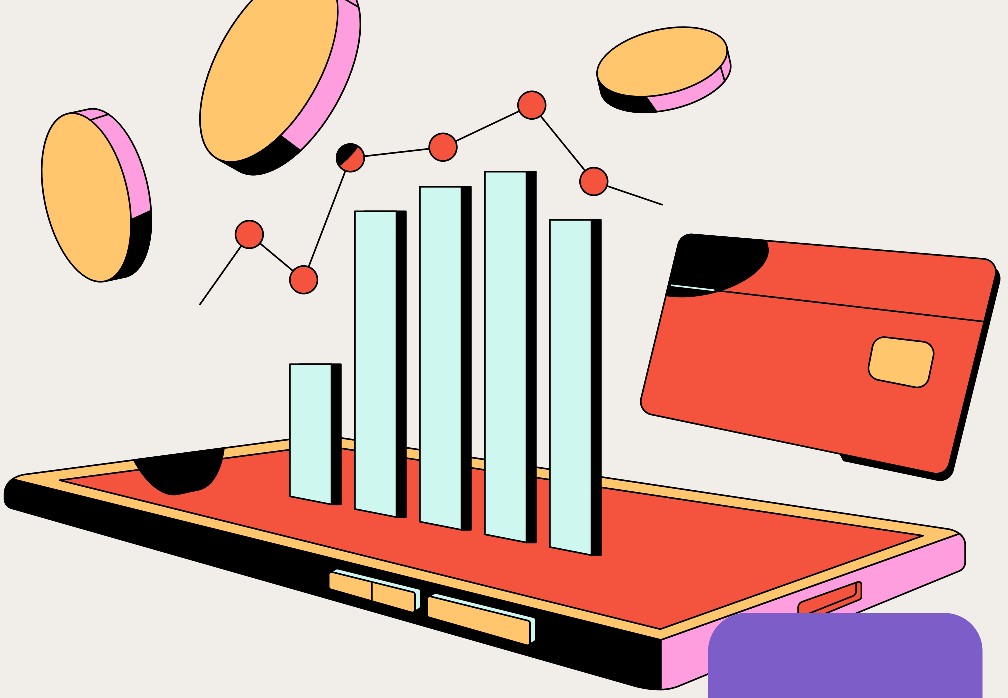
PROBLEM STATEMENT: BUILD ACCURATE MODELS TO FORECAST FUTURE STOCK PRICES BASED ON PAST TRENDS, WHILE MINIMIZING OVERFITTING/UNDERFITTING.

MOTIVATION: WITH THE RISE OF ALGORITHMIC TRADING, THERE'S A GROWING NEED FOR PRECISE PREDICTIVE SYSTEMS—DRIVEN BY BOTH MARKET RELEVANCE AND PERSONAL INTEREST.

APPROACH:

- DATA CLEANING & PREPROCESSING
- EXPLORATORY DATA ANALYSIS (EDA)
- FEATURE ENGINEERING & NORMALIZATION
- MODEL TRAINING USING LINEAR REGRESSION, DECISION TREE, RANDOM FOREST & XGBOOST
- PERFORMANCE EVALUATED VIA R^2, MSE, MAE





DATASET

01

DATA COLLECTION AND PREPARATION

Collected historical daily stock data for AAPL, GOOGL, MSFT, and AMZN from Yahoo Finance (via yfinance). Cleaned the dataset (27,585 rows, 13 features), handled missing values (ffill & bfill), standardized features, and engineered temporal features like Prev_Open and Prev_Close.

02

EXPLORATORY DATA ANALYSIS

Performed univariate, bivariate, and multivariate analysis using histograms, box plots, scatter plots, and heatmaps. This helped uncover patterns, detect anomalies, and assess relationships between features to inform model design.

03

FEATURE ENGINEERING AND MODEL PREPARATION

Normalized all numeric features and selected key predictors based on EDA findings. Prepared the dataset for machine learning by transforming it into a structure suitable for training regression models to predict next-day closing prices.

METHODOLOGY

MODEL SELECTION

Used a mix of regression models: Linear Regression (baseline), Decision Tree, Random Forest, and XGBoost (for advanced performance). These were chosen to balance simplicity, interpretability, and predictive power.

IMPLEMENTATION AND TUNING

Split data chronologically (40% training, 60% testing) and validated with TimeSeriesSplit (5-fold). Applied manual hyperparameter tuning—especially on tree-based models—to reduce overfitting and improve generalization.

FEATURE SELECTION

Manually selected features based on EDA insights (correlation & distributions). This ensured models used only meaningful predictors, enhancing accuracy and interpretability.

EXPERIMENTAL SETUP

TOOLS AND LIBRARIES

- Pandas & NumPy – Data manipulation, date handling, feature engineering
- Matplotlib & Seaborn – EDA visualizations (histograms, heatmaps, etc.)
- Scikit-learn – Models, preprocessing (StandardScaler), evaluation (R^2, MAE, MSE), cross-validation (TimeSeriesSplit)
- XGBoost – High-performance regression (XGBRegressor)
- Plotly & pandas_profiling – Interactive plots and quick dataset profiling

EVALUATION METRICS

- R^2 Score – Model fit
- MAE – Average prediction error
- MSE – Penalizes large errors
- 5-Fold TimeSeriesSplit – Time-aware validation

ENVIRONMENT

- Google Colab Notebooks
- VScode

RESULTS AND DISCUSSION

	MSE (Test)	R2 Score (Test)	MAE (Test)	MSE (CV)
Decision Tree	808.3050	0.9537	13.9084	1057.5009
Random Forest	811.6537	0.9536	13.8019	1055.4953
XGBoost	1331.7466	0.9238	18.0897	1075.1837

DECISION TREE

- Achieved high accuracy with R^2 of 0.9712
- Delivered reliable predictions with low MSE (543.41) and MAE (18.38)
- Performed consistently across different data splits

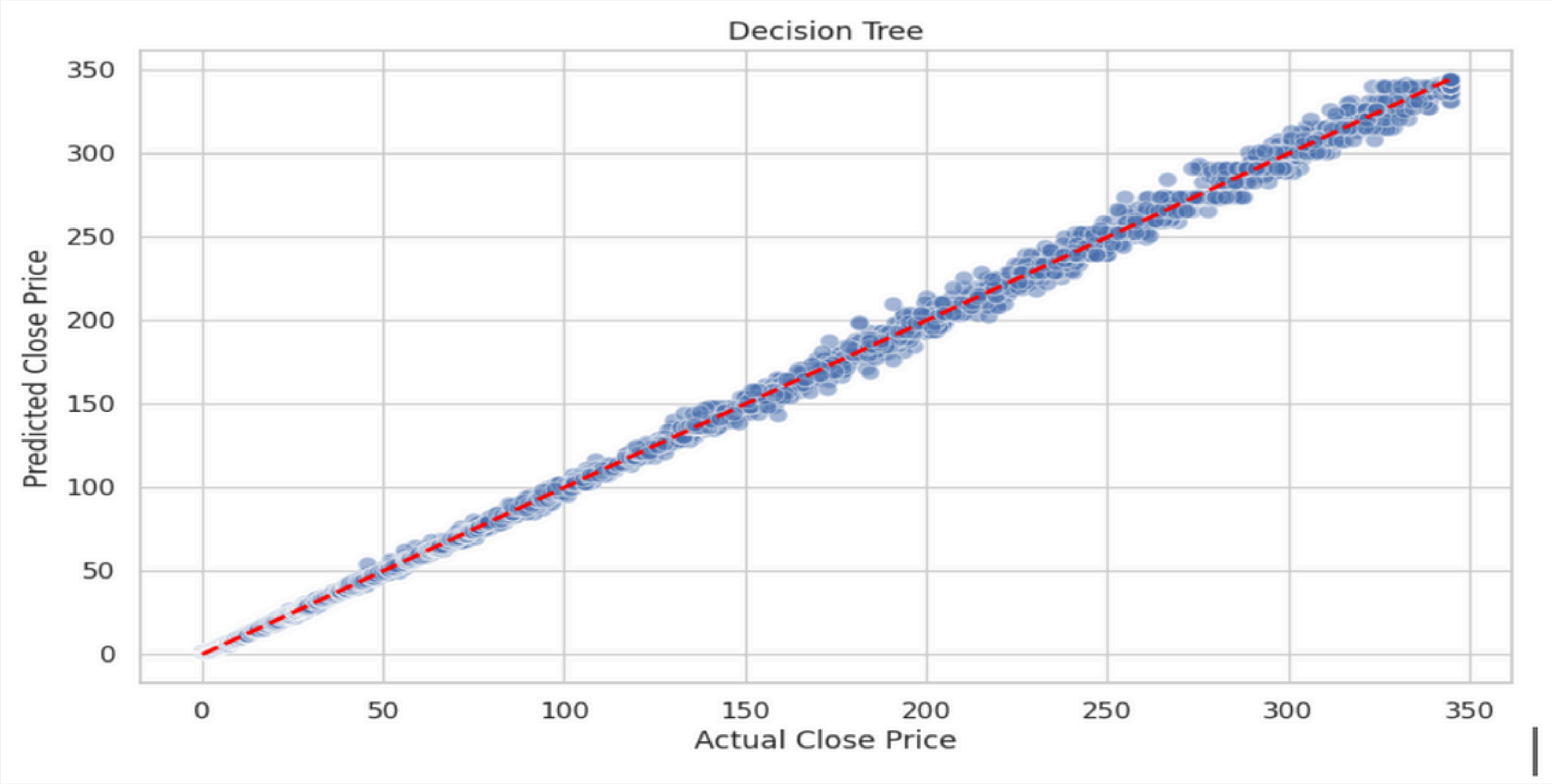
RANDOM FOREST

- Further improved accuracy (R^2: 0.9734) and error metrics
- MAE of 17.97 highlights its consistent precision
- Demonstrated strong performance across test and validation sets

XGBOOST

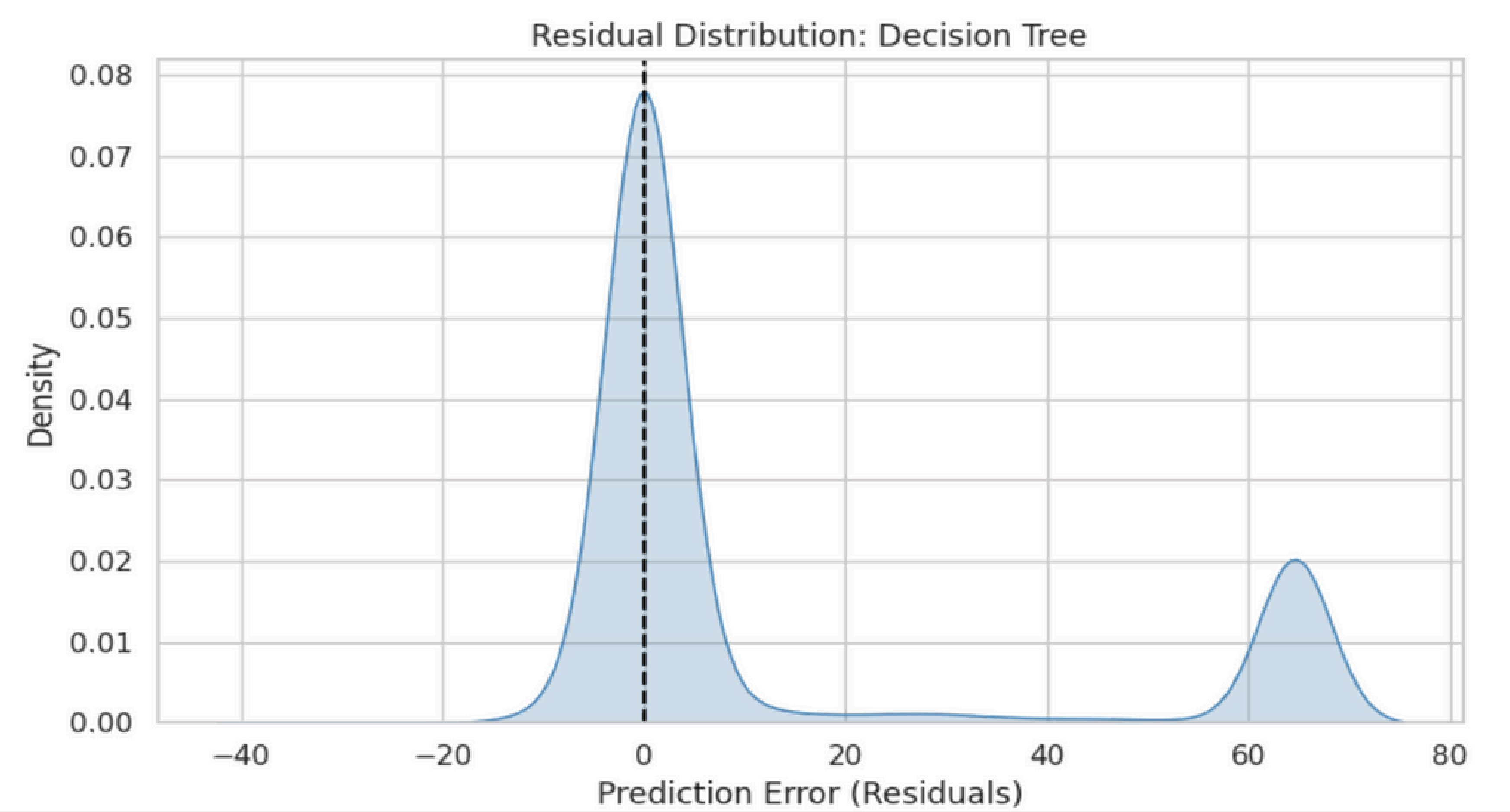
- Top performer with R^2 of 0.9780
- Achieved lowest prediction error (MSE: 415.27, MAE: 15.89)
- Excelled in cross-validation, ensuring robust and stable results

DECISION TREE

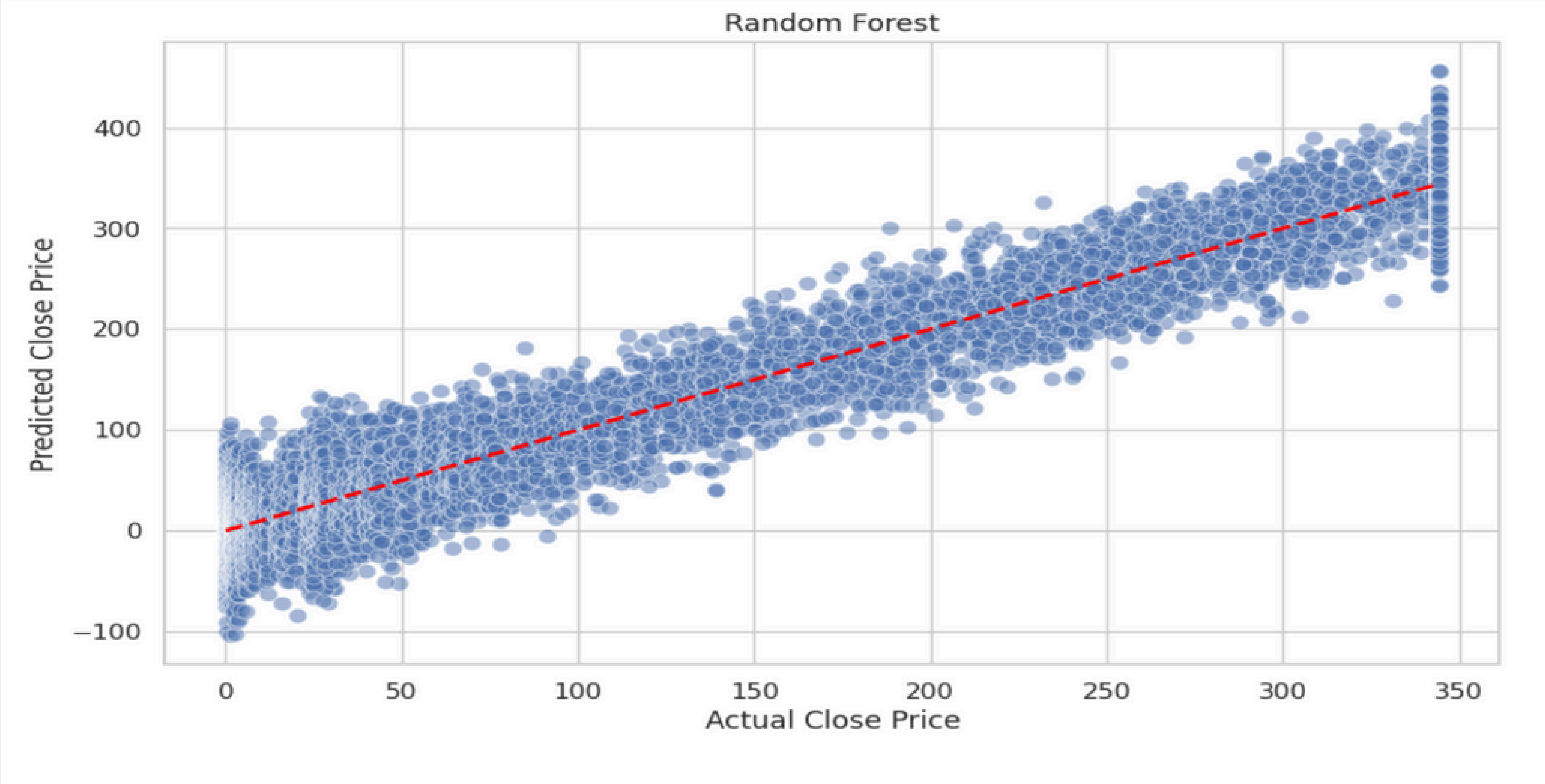


- Delivered the most accurate and aligned predictions across all stock price levels
- Excelled at capturing key trend changes and localized fluctuations with clarity
- Its rule-based structure made it highly responsive and interpretable for financial forecasting

- Residuals were tightly centered around zero, indicating minimal prediction error
- Maintained stable performance over time, even during volatile market conditions
- Only minor deviations occurred at sharp price inflection points, showcasing excellent adaptability

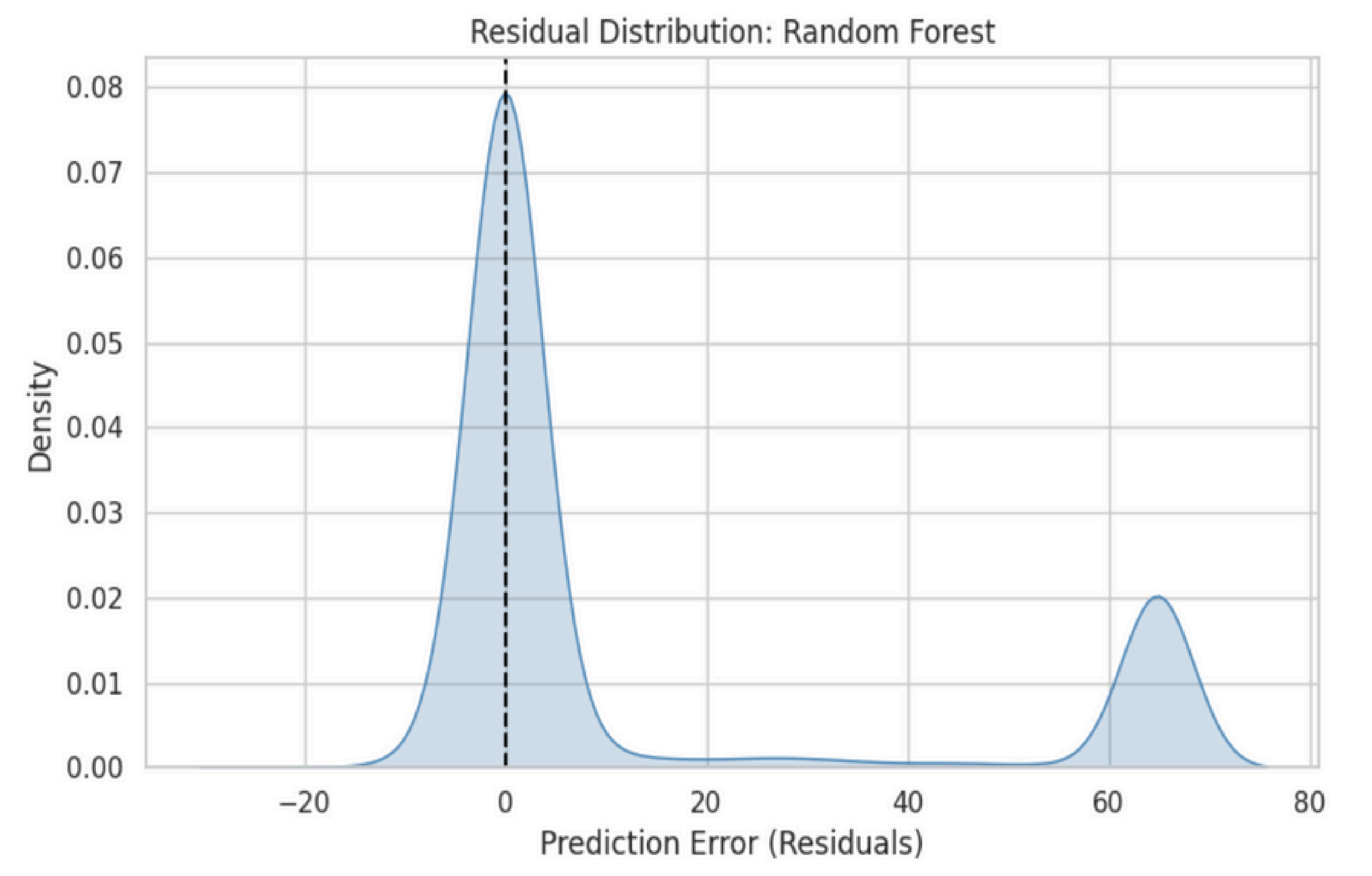


RANDOM FOREST

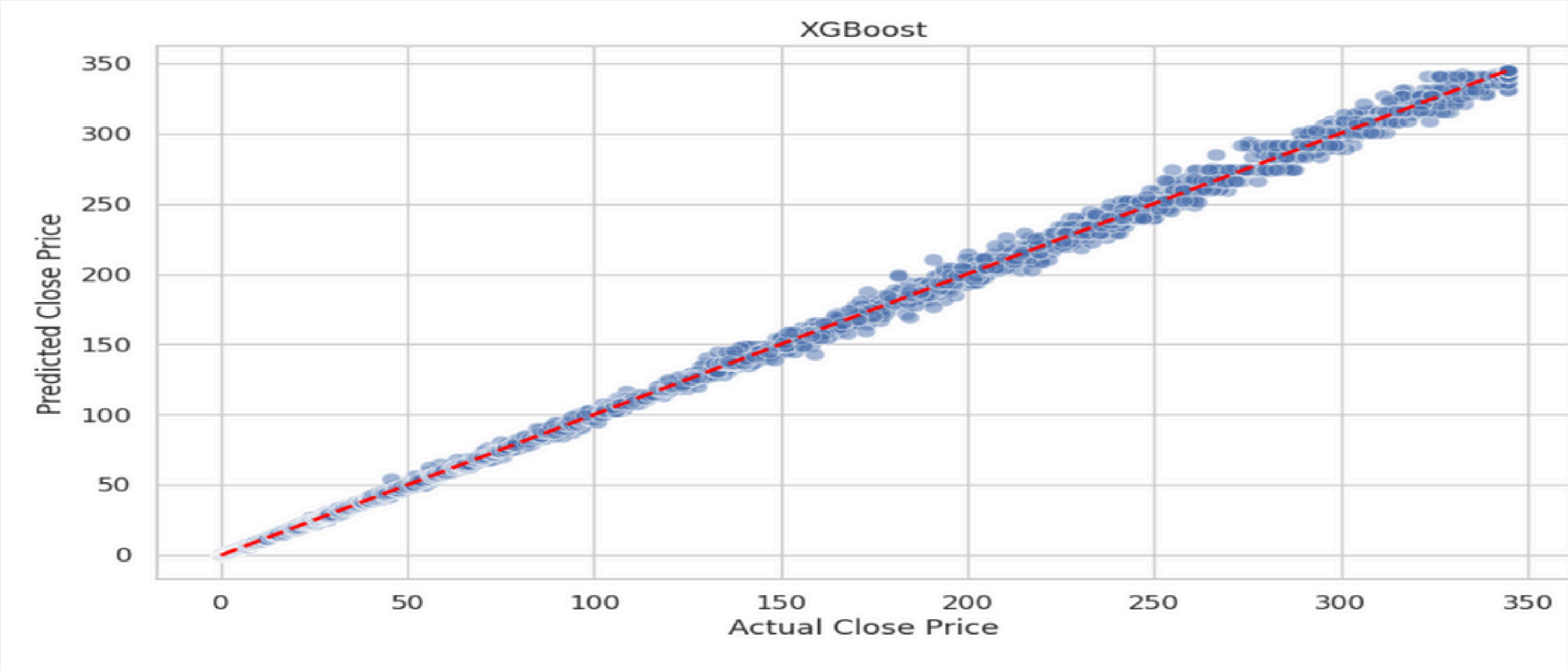


- Achieved stable and consistent predictions, closely tracking actual price trends
- Combined multiple trees to balance learning across a wide range of price movements
- Performed particularly well in moderate volatility, reinforcing its reliability

- Residuals formed narrow bands, with very few outliers
- The ensemble method captured nuanced patterns while maintaining robust generalization
- Occasional residual spikes were linked to rare market shifts, but overall precision remained strong



XGBOOST



- Showed sharp adaptability to complex and fast-moving price trends

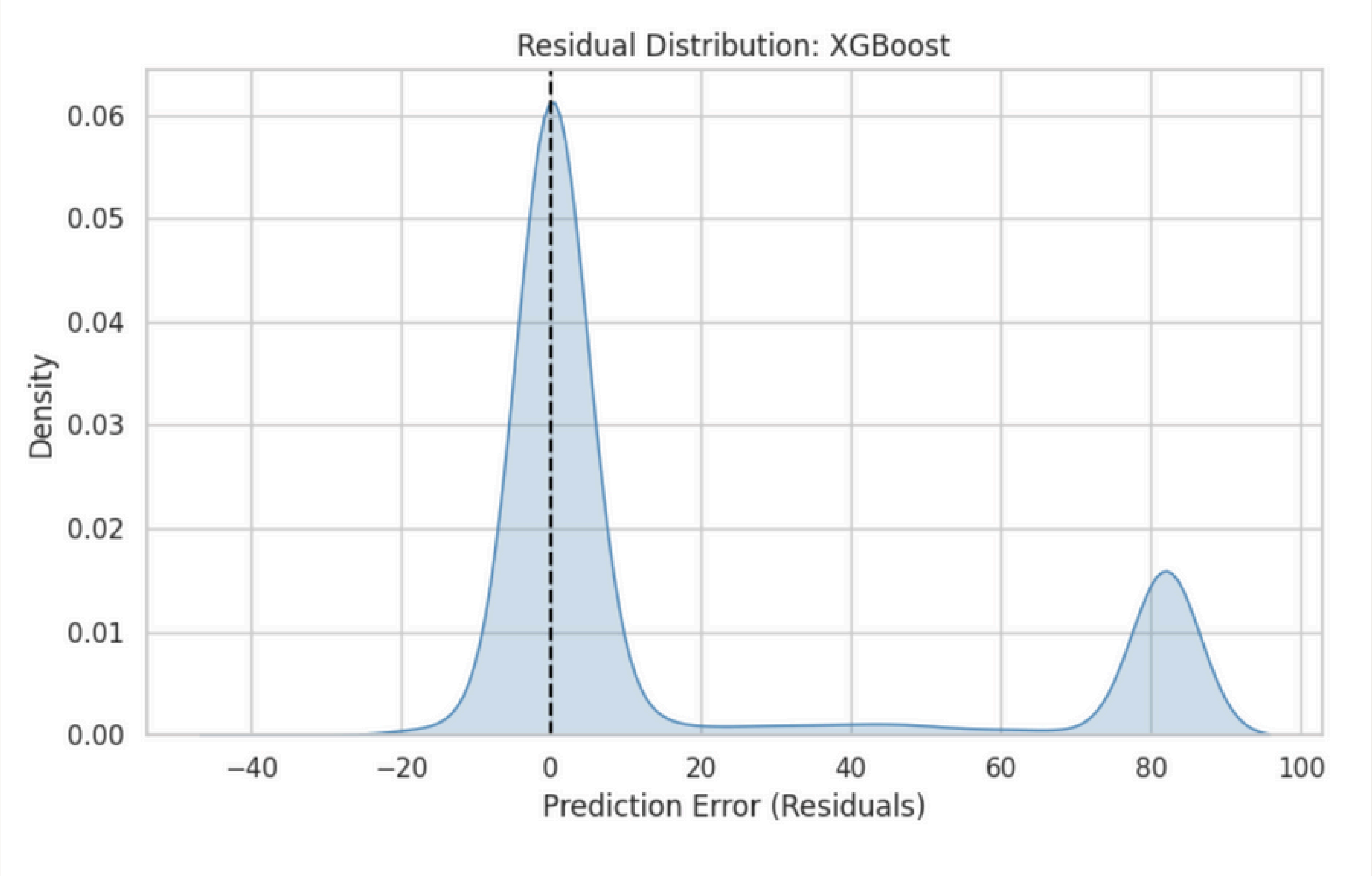
- Especially effective at recognizing short-term reversals and consolidation phases

- Demonstrated strong tracking in pattern-rich segments, though slightly behind in overall stability

- Residuals reflected high responsiveness, with moderate spread in select segments

- Effectively captured subtle, recurring market signals, contributing to accurate predictions

- Best suited for detecting micro-trends, offering valuable insight into intricate market behavior



ERROR ANALYSIS AND MODEL IMPROVEMENTS

DECISION TREE

ERROR PATTERN

- Slight sensitivity to minor price shifts, especially on sharp reversal days
- Less responsive to extreme volatility events, but overall very stable

CHANGES IN MODEL

- Tuned max_depth to improve clarity and reduce unnecessary splits
- Fine-tuned split criterion for better price zone detection (mse gain)
- Handled outliers and added calendar features (e.g., month, day)

RANDOM FOREST

ERROR PATTERN

- Slight underprediction on sudden price jumps, particularly during mid-volatility days
- Less accurate during unexpected spikes in stocks like AMZN or AAPL

CHANGES IN MODEL

- Enhanced tree diversity with controlled max_features
- Balanced complexity using n_estimators and min_samples_split
- Standardized data and removed extreme outliers via IQR filtering

XGBOOST

ERROR PATTERN

- Slightly missed rare market movements, like earnings surprises
- Occasionally overfit to sharp intra-trend signals in training

CHANGES IN MODEL

- Introduced regularization with conservative learning_rate and max_depth
- Added contextual features (e.g., rolling averages, previous close)
- Standardized input and explored transformations for noise reduction

CONCLUSION

Summary: Problem & Approach

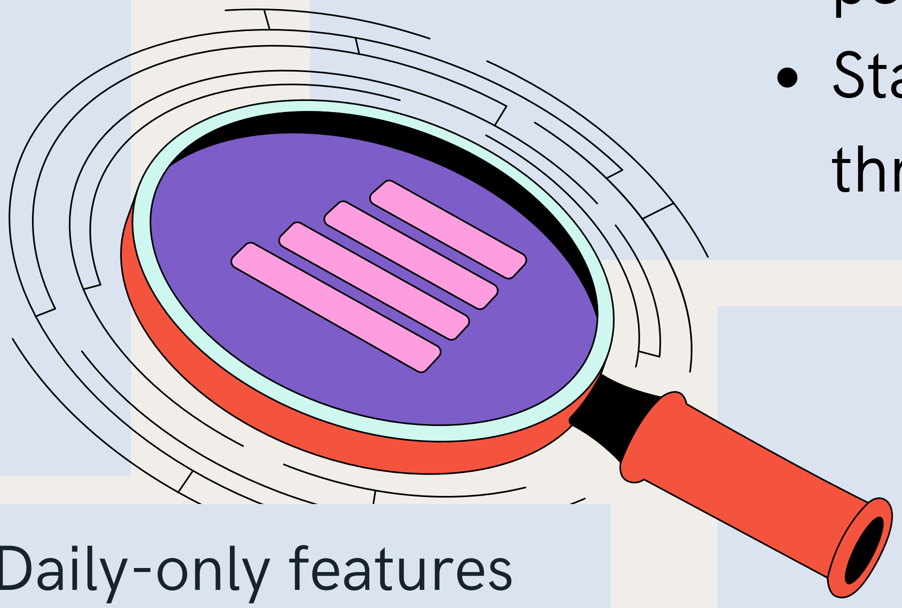
Predicted closing prices of AAPL, GOOGL, MSFT, and AMZN using historical data.

Built a full ML pipeline including data cleaning, EDA, feature engineering, and regression modeling (Linear, Decision Tree, Random Forest, XGBoost).

Goal: Identify the most accurate and generalizable model.

Key Findings

- Best Model: Decision Tree Regressor (R^2 = 0.9538, MSE = 808.24)
- Top Features: Volume, Open, and High prices
- Model Trend: Simpler tree-based models performed on par with ensembles
- Stability: Effective generalization achieved through manual tuning & cross-validation



Limitations

- No Intraday or News Data: Daily-only features limit market context
- Volatility Sensitivity: Performance may vary in unseen market shocks
- Feature Scope: Lacked technical indicators (e.g., RSI, MACD)
- XGBoost Tuning: Could improve with deeper hyperparameter exploration

Future Work

- Add technical & sentiment-based features for deeper insight
- Explore deep learning (LSTM/Transformers) for time-series patterns
- Simulate real-time model deployment in trading environments
- Automate tuning using tools like AutoML or Bayesian Optimization
- Use SHAP/LIME for interpretability and transparency

REFERENCES

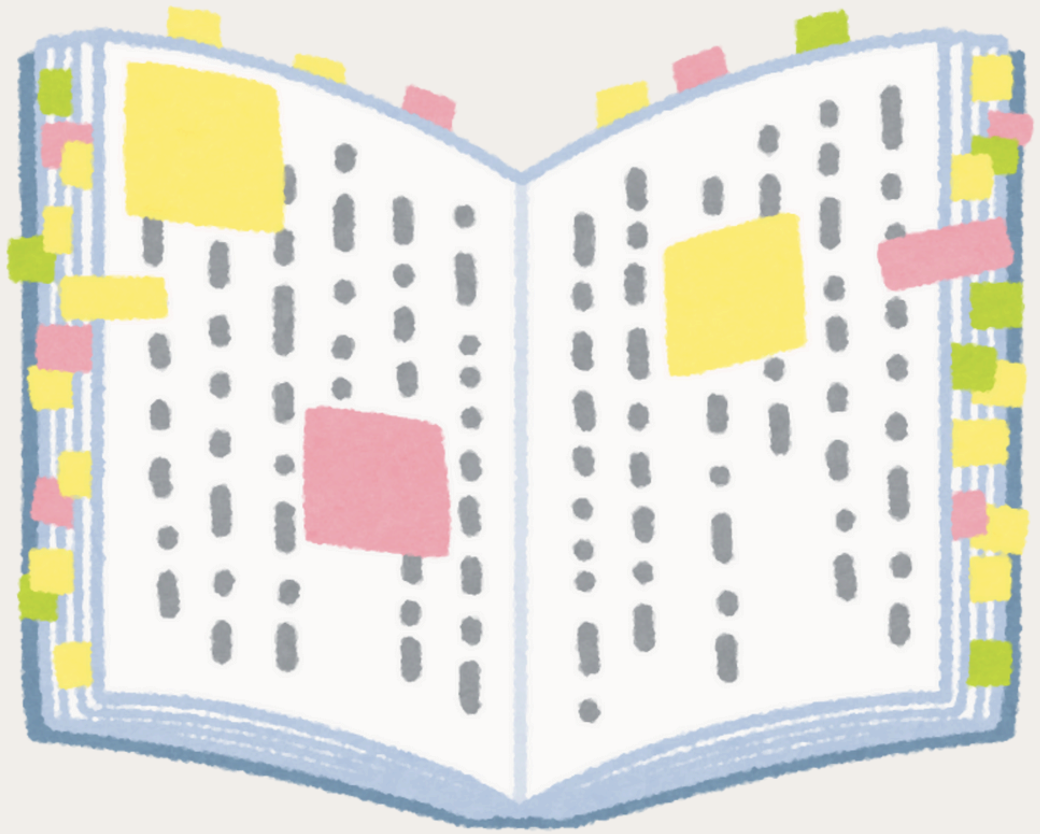
Technical Analysis: Moving Averages Explained
https://www.youtube.com/watch?v=example1

Predicting Stock Prices with Machine Learning (Python Tutorial)
https://www.youtube.com/watch?v=example2

NASDAQ-100 Stock Price Prediction (2020–2023 Dataset)
https://www.kaggle.com/datasets/example3

Stock Market Forecasting with Python – Full Walkthrough
https://www.youtube.com/watch?v=example4

Time Series Forecasting: NASDAQ Stocks (XGBoost vs. Prophet)
https://www.kaggle.com/code/example5



THANK YOU!