An Intelligent Forecasting Engine for Algorithmic Trading via Agent Interface

Problem Statement & Motivation

- Stock market forecasting is inherently **noisy**, **nonlinear**, and affected by countless factors.
- Retail and institutional traders seek predictive tools to **identify profitable opportunities** ahead of time.
- Traditional methods often rely on **manual analysis or single-model predictions**, which can be narrow or biased.
- **<u>There is a growing need for intelligent, automated systems that can:</u>**
  - Analyze market data efficiently
  - Generate forecasts with multiple models
  - Provide insights in a user-friendly and customizable way

Our goal:

Build a smart, end-to-end pipeline that automates data collection, feature extraction, and multi-model forecasting—**wrapped in an intuitive agent interface**.

Our Motivation:

Empower traders and analysts with **accessible algotrading tools** that are adaptable, scalable, and transparent.

Project Overview

- **Built an end-to-end algotrading system for forecasting stock prices.**
- **Automatically: Fetches data from “Yahoo Finance” & enriches with features and correlations.**
- **Generates forecasts using multiple ML models.**
- **Wrapped in a chat-like agent interface for easy user interaction.**
- **Supports custom input: stock symbol, forecast horizon, and model choice.**
- **Designed for modularity, scalability, and real-world trading use.**

Data Acquisition and Preprocessing

1 Download Data

Use yfinance to download closing prices for selected ticker from 2015 to current date of Today and calculating measurements: 'Close', 'HLP', 'GAP', 'HC', 'LC', 'VolumeChange','RSI', 'MACD', 'MACD_signal', 'Volatility'.

2 Calculate Returns

Compute percentage changes to obtain return data for analysis.

3 Data Cleaning

Remove NaN values and outliers using interquartile range (IQR) method to ensure data quality.

4 Handle Multicollinearity

removed highly correlated features by dropping one feature from each pair with a correlation above 0.8, based on the upper triangle of the correlation matrix.

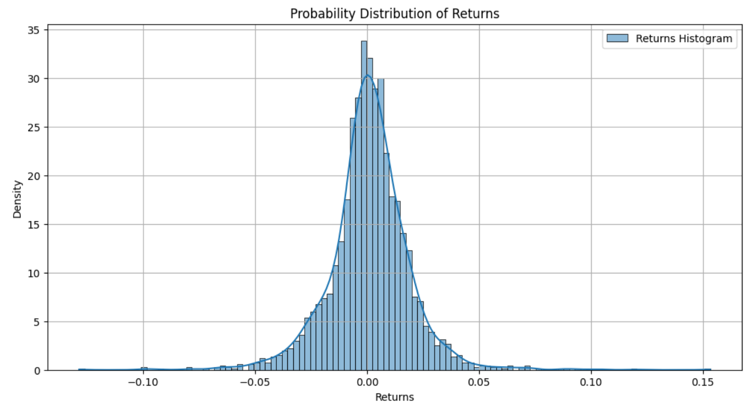
Statistical Analysis

Descriptive Statistics

Calculate and display summary statistics for the data, including mean, standard deviation, and quartiles.

Probability Distribution

Plot the probability distribution of returns using a histogram.

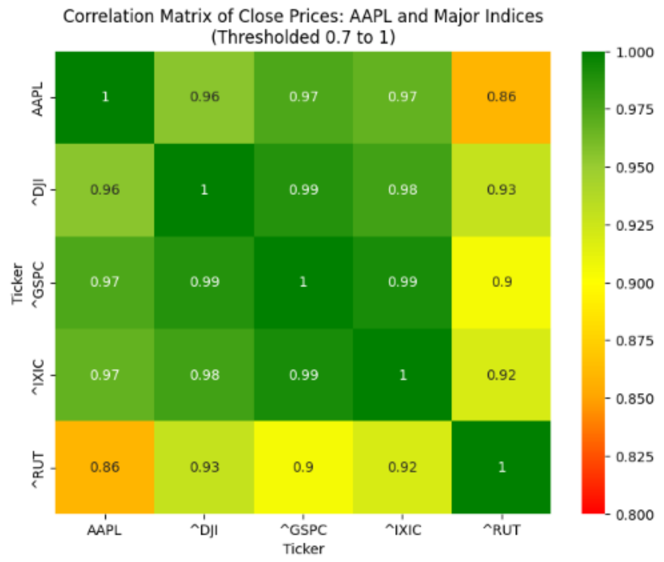


Feature Correlation Analysis

Generate a correlation matrix heatmap to visualize relationships between different assets in the portfolio.

Ticker To Major Indices Correlation

measure the correlation between the selected stock and market indices (e.g., S&P 500, NASDAQ) based on historical closing prices



ARIMA Modeling

Data Preparation

Split time series into **train and test sets** (e.g., 95% train, 5% test).

Model Training & Prediction

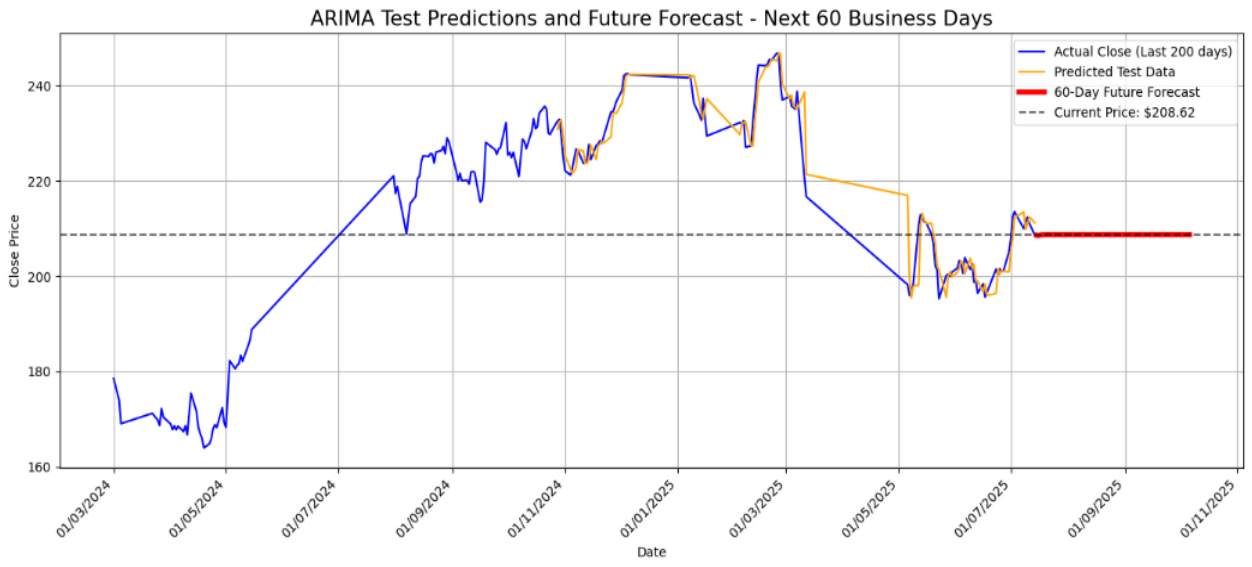
- Use **ARIMA(p,d,q)** to iteratively train on the dataset.
- Predict one step ahead at each test point, simulating real-time inference.

Evaluation Metrics

- Calculate **RMSE and R²** to evaluate prediction accuracy.
- Visualize actual vs. predicted trends over time.

Future Forecasting

- Fit ARIMA to the full historical series (train + test).
- **Forecast X business days into the future** and display results with current price and forecast trendline.



SARIMAX Modeling

SARIMAX- Enhance ARIMA by modeling both trends and seasonality in stock prices

Data Preparation

Split time series into **train and test sets** (e.g., 95% train, 5% test).

Model Training & Prediction

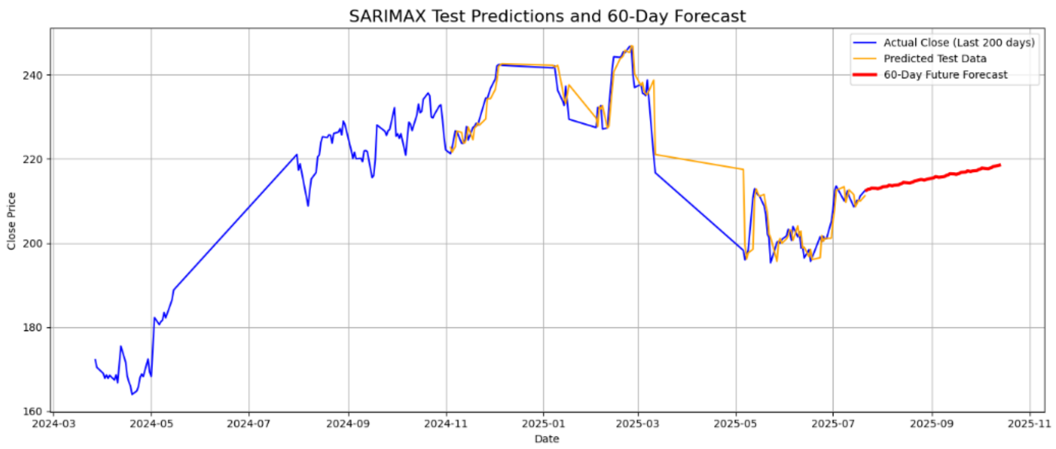
- Fit a **SARIMAX(p,d,q)(P,D,Q,s)** model to capture both short-term patterns and seasonal cycles.
- Predict test points step-by-step using rolling forecast.

Evaluation Metrics

- Calculate **RMSE and R²** to evaluate prediction accuracy.
- Visualize predicted vs actual prices over the test period.

Future Forecasting

- Refit **SARIMAX** on the **entire historical dataset**.
- Forecast future price movement for a set number of business days.



Xgboost Modeling

Feature Engineering

Lagged features, moving averages, RSI, MACD, volatility, and ratio-based indicators. Derived signals like RSI overbought/oversold and MACD crossover.

Model Training & Prediction

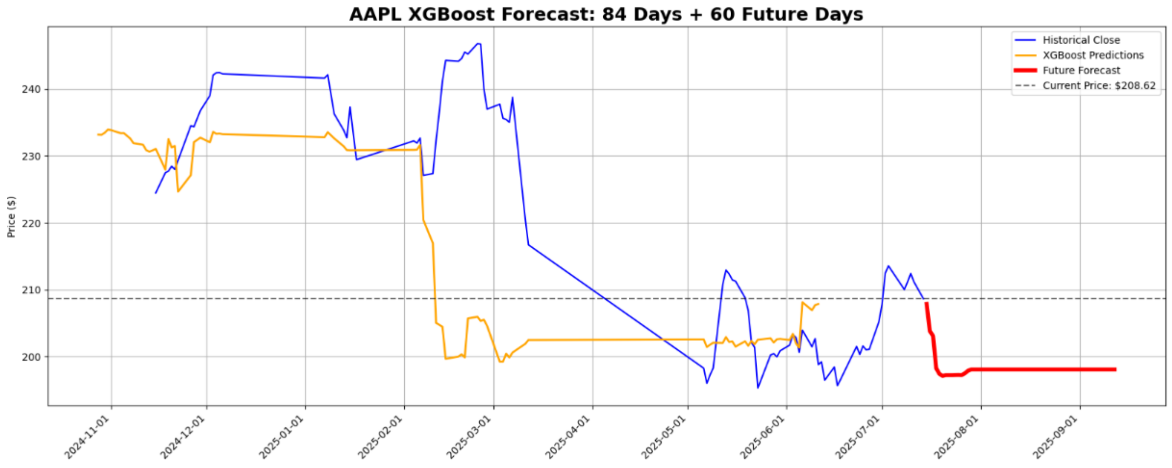
- Train XGBRegressor using scaled features.
- Predict test set using rolling window strategy and evaluate with RMSE and R^2 metrics.

Future Forecasting

- Iteratively predict next 20 trading days using last known data.
- Automatically updates input with each predicted value.

Explainability

- Feature importance plot reveals which technical indicators most strongly influence the model’s stock price predictions



LSTM Modeling

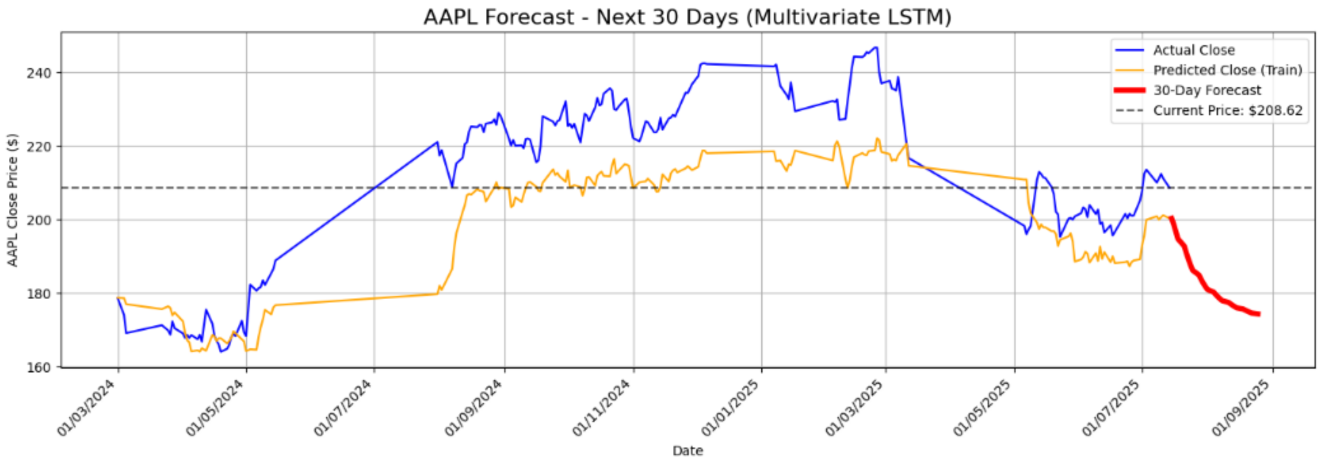
Data Scaling
All features normalized using StandardScaler to ensure stable gradient flow.

Sequence Creation:
Time series split into rolling sequences of 60-time steps for multivariate modeling.

Feature Combination
Combined technical indicators as input features to capture market dynamics.

Model Architecture
Two stacked LSTM layers (100 units each) → **Dropout regularization** → **Dense layer for final prediction**

Model Compilation
Compiled with Adam optimizer and Mean Squared Error loss.



Prediction
Generates predicted closing prices on the training set.

Training
Trained on historical sequences with a validation split (e.g., 90/10).

Future Forecasting
Predicts future n days by feeding model outputs into new sequences recursively.

Visualization
Plots true values, in-sample predictions, and future forecast to evaluate performance.

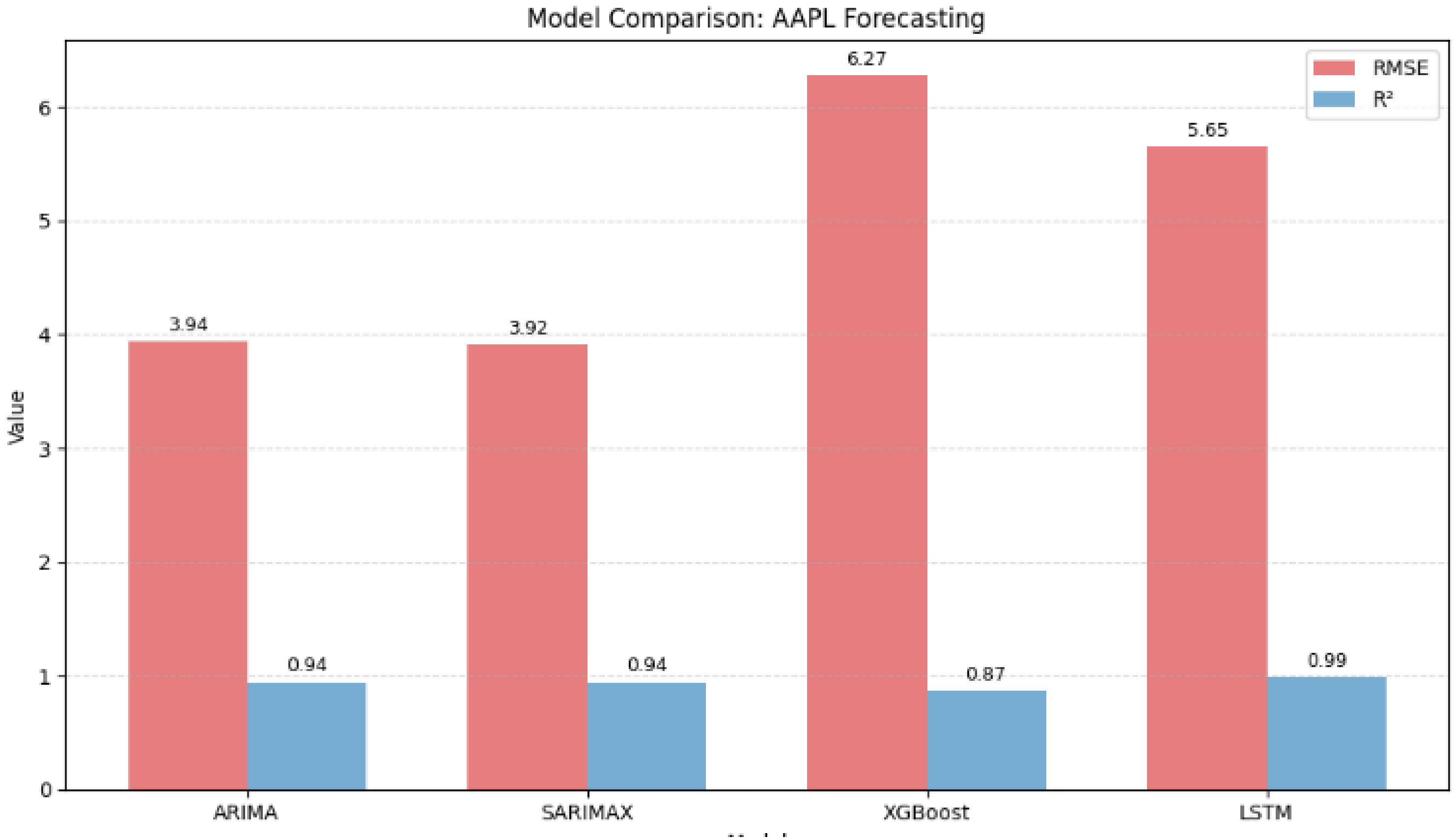
Model Comparison

Use case - predicting the next **30 days** of stock prices of **AAPL stock** by using a **60-day window** of past data to predict the next day.

Model	RSME	R^2
ARIMA	**3.9397**	0.9417
SARIMAX	3.9185	0.9419
XGBoost	6.2733	0.8747
LSTM	5.6473	**0.9930**

RMSE: Measures the average difference between predicted and actual values. **Lower is better** because it means the model's predictions are closer to the real values.

R^2: Measures how well the model explains the variance in the data. **Closer to 1 is better.**



All models are effective, with SARIMAX and LSTM standing out:
one for interpretability, the other for modeling flexibility and long-term forecasting power

Agent Interface

Purpose: Enables interactive, customizable stock forecasting with minimal user input.

User Flow:

1. Prompt user to input a stock ticker (e.g., AAPL, MSFT)
2. Ask how many days to forecast (1–60)
3. Let user select a model (ARIMA, SARIMAX, XGBoost, LSTM, or all)
4. Optionally adjust hyperparameters (epochs, sequence length)
5. Automatically run preprocessing, training, and prediction
6. Show results + ask whether to run another forecast

Key Features:

- Auto timeout handling (ends session after inactivity)
- Ticker validation using Yahoo Finance API
- Automatic data download, feature cleaning, and correlation filtering
- Supports 4 model types + evaluation metrics (RMSE, R^2)
- Looping interface for back-to-back forecasts

Outputs:

- Model evaluation table (RMSE, R^2)
- Visual plots (optional)

Conclusion and Future Work

- Our system supports **all tickers available in Yahoo Finance**, enabling wide-scale forecasting coverage.
- **All models demonstrated strong performance** in forecasting stock prices.
- **LSTM achieved the highest R^2** (0.9930), excelling at learning sequential patterns.
- Classical models like **ARIMA and SARIMAX provided reliable results with low RMSE**.

Future Work

- **Expand feature set:** incorporate technical indicators, volume, and macroeconomic signals to enhance prediction accuracy.
- **Enhance the agent:** integrate an LLM-based interface for natural language querying and forecasting.
- **Implement RAG integration:** combine real-time news and articles to make context-aware predictions.