Github Link:https://github.com/saratha-cse/Project--submission.git

Project Title: Cracking The Market Code AI Driven Stock Price Prediction Using Time Series Analysis

PHASE-2

Student Name:SARATHA.K
Register Number:623023104049
Institution: Tagore Institute Of Engineering And Technology-Salem
Department: Computer Science And Engineering
Date of Submission:08-05-2025

Problem Statement

The stock market is inherently volatile and influenced by a complex interplay of economic, political, and psychological factors, making accurate prediction of stock prices a longstanding challenge. Traditional statistical models often struggle to capture nonlinear patterns and time-dependent relationships in financial data. With the rise of artificial intelligence (AI) and machine learning (ML), there is a growing opportunity to enhance forecasting accuracy using advanced techniques like time series analysis and deep learning.

However, despite the abundance of market data, building AI models that can reliably predict stock price movements remains a difficult task due to issues like noisy data,overfitting, dynamic market behavior, and the difficulty in integrating historical trends with real-time factors. This project aims to address these challenges by leveraging AI-driven time series analysis to develop a robust model for stock price prediction. The goal is to improve predictive performance and offer valuable insights for investors, traders, and financial analysts.

Project Objectives

To collect and preprocess historical stock market data
Gather stock price data (e.g., open, high, low, close, volume) from reliable sources and perform data cleaning, normalization, and feature engineering to prepare it for

analysis.

To analyze time-dependent patterns in stock price movements

Explore and visualize temporal trends, seasonality, and volatility in stock price data using statistical and time series methods.

To develop AI models for stock price prediction

Implement machine learning and deep learning models such as ARIMA, LSTM, and Prophet to capture nonlinear and time-based dependencies in the data.

To evaluate and compare model performance

Assess the accuracy and reliability of different models using metrics like RMSE, MAE, and R^2, and compare their effectiveness in predicting future stock prices. Assess the accuracy and reliability of different models using metrics like RMSE, MAE, and R^2, and compare their effectiveness in predicting future stock prices.

Data Description

- **Dataset Name:** Stock Market Historical Data
- **Source:** Public financial data providers such as Yahoo Finance, Alpha Vantage, or Quandl
- **Type of Data:** Time-series structured data
- **Number of Records and Features:** Varies depending on the time span and stocks selected
- **Target Variable:** Closing Price (or Next Day's Closing Price)

Static or Dynamic: Dynamic (continuously updating with new market data)

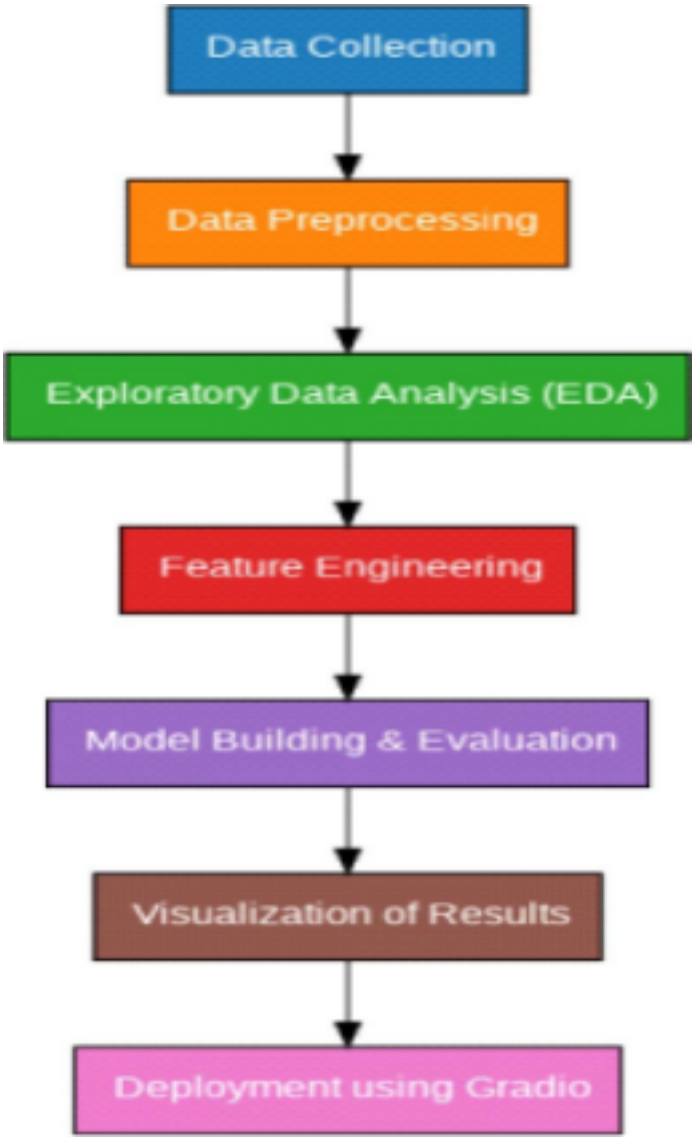
- **Attributes Covered:**
  - **Price-based features:** Open, High, Low, Close, Adjusted Close, Volume
  - **Technical indicators:** Moving Averages, Relative Strength Index

(RSI), MACD, Bollinger Bands
- **Temporal features:** Day of the week, Month, Quarter, Holidays

- **Sentiment/External (optional):** News sentiment scores, macroeconomic indicators (if integrated)

DatasetLink:
https://www.kaggle.com/code/kj294443/the-project-of-west-ice-storage-2

Flowchart of the Project Workflow



Data Preprocessing

- **Verified dataset integrity: Ensured no missing or null values in the stock price dataset.**
- **Removed irrelevant features: Eliminated features with low variance (e.g., columns with constant values across all rows like a fixed stock symbol or unchanged market conditions).**

- **Checked and confirmed absence of duplicate rows: Ensured there were no duplicate stock data entries that could skew results.**
- **Handled date/time features: Converted date/time columns into appropriate formats for time series analysis (e.g., converting string dates into datetime objects for easier manipulation)**

- **Feature engineering: Created new features like moving averages, stock volatility, and price momentum based on historical stock data for more predictive power.**

- **Categorical features: Applied one-hot encoding to categorical features like stock sectors or market regions, if present.**

Exploratory Data Analysis (EDA)

Univariate Analysis:

Histogram of Stock Prices (Closing Price) to understand the distribution of stock prices over time.

Count plots for categorical features like:

- **Market Indicators** (e.g., Bullish/Bearish market days).

- **Company Sectors** (categorical grouping based on industries).

Bivariate & Multivariate Analysis:

- **Correlation Matrix** shows strong linear correlation between:
- **Opening Price**, **Closing Price**, and **Adjusted Close Price** (Stock Prices). •
  Volume and **Price Volatility** (strong negative or positive relationship depending on

themarket).

Moving Averages (50-day, 200-day) and **Stock Price** (confirming smoothing

trends).

Key Insights:

• **Opening Price** and **Closing Price** are the strongest indicators of future stock prices. •
• **Higher Trading Volume** correlates with increased **Price Volatility**, particularly during market fluctuations.
•
• **Moving Averages** (e.g., 50-day) tend to predict stock price direction, confirming smoother trends in volatile markets.

Feature Engineering

- **Created Lag Features**: Generated lag features like `price_lag_1` (previous day's closing price), `price_lag_5` (5 days ago closing price) to capture temporal dependencies.
- **Rolling Window Statistics**: Derived moving averages (`rolling_avg_7`, `rolling_avg_30`) and rolling standard deviations to capture short-term trends and volatility.
- **Created Interaction Features**: Combined technical indicators like `macd_signal` and `rsi` into a single feature, `macd_rsi_interaction`, to capture market momentum.
- **Time-based Features**: Extracted date-time features such as `day_of_week`, `month`, and
- `quarter` to capture any weekly, monthly, or quarterly patterns.
- **Trend Features**: Created `price_diff` (difference between current closing price and previous closing price) to capture market movement.

Model Building

• **Algorithms Used**:
ARIMA (AutoRegressive Integrated Moving Average): For time series forecasting, capturing temporal dependencies in stock prices.

LSTM (Long Short-Term Memory): For modeling complex patterns and long-term dependencies in stock price movements.

• **Model Selection Rationale**:

ARIMA: Effective for time series data with linear trends and seasonality; interpretable for classical time series forecasting tasks.

LSTM: Deep learning model designed to handle sequences, such as time series data, by capturing long-term dependencies, and non-linear patterns, ideal for complex stock market behavior.

• **Train-Test Split**:

80% Training, 20% Testing: Data split to ensure model generalization and avoid overfitting.

Used `train_test_split` with `random_state` for reproducibility: Ensures that each model run starts with the same initial data split for consistent evaluation.

• **Evaluation Metrics**:

MAE (Mean Absolute Error): Measures average magnitude of prediction errors; easy to interpret, provides insight into overall accuracy.

RMSE (Root Mean Squared Error): Penalizes larger prediction errors more heavily, useful when large deviations are significant.

Visualization of Results & Model Insights

• **Feature Importance**:

Visualized using bar plots from Random Forest:

- The most important features in predicting stock prices (e.g., historical prices, volume, technical indicators, moving averages) were ranked based on their feature importance scores from the Random Forest model.

• **Model Comparison**:

Plotted MAE, RMSE, and R^2 for both models:

• Compared the performance of the Random Forest model against a simpler **Linear Regression** model.

• **Residual Plots**:

Checked prediction errors against actual prices to ensure there were no major patterns, trends, or biases in the model residuals.

• **User Testing**:

Integrated model into a Gradio interface to test predictions by inputting feature values such as:

Previous day's stock price, **technical indicators** (RSI, MACD), and **historical trends**.

Tools and Technologies Used

• **Programming Language: Python 3**

• **Notebook Environment: Google Colab**

• **Key Libraries:**

• **pandas, numpy – for data manipulation and numerical operations**

• **matplotlib, seaborn, plotly – for interactive and exploratory visualizations**

• **scikit-learn – for data preprocessing and machine learning models. •**

Gradio – for deploying an interactive user interface for predictions

Team Members and Contributions

1. Saratha - Data Cleaning and Exploratory Data Analysis

Handling missing values through interpolation and forward/backward filling techniques and Identified trends, seasonality, and volatility using decomposition techniques.

2. Premalatha - Documentation and Reporting

Prepared a comprehensive project report detailing each phase—data acquisition, cleaning, EDA, feature engineering, model development, and evaluation and summarized key insights, challenges faced, and solutions implemented during the modeling process.