
# Stock Price Prediction Using Machine Learning and Real-Time Data Visualization

Stock markets are complex systems characterized by volatile and dynamic behavior influenced by numerous factors. Predicting stock prices accurately, particularly short-term trends, remains a significant challenge. However, the advancement of machine learning (ML) algorithms combined with access to real-time financial data has enabled innovative approaches to forecasting stock trends.

This project focuses on leveraging historical stock data through Linear Regression models to predict the next day’s closing price. Furthermore, a graphical user interface (GUI) is developed to visualize both historical data and predicted values seamlessly.

# Project Objectives and Operational Definitions Definitions

## Objective 1

Develop a user-friendly GUI enabling input of stock ticker symbols and displaying predicted closing prices.

## Objective 2

Apply Linear Regression techniques on normalized historical stock data for predictive modeling of price trends.

## Objective 3

Visualize historical and forecasted stock prices using embedded graphical plots within the application.

## Key Terms

- Stock Ticker: Unique symbol identifying publicly traded stock.
- Closing Price: Final price of a stock at market close each day.
- Prediction Accuracy: Proximity of predicted price to actual closing price.
- Linear Regression: Supervised ML algorithm modeling relationships for prediction.


# Research Hypothesis and Experimental Design

## Hypothesis

Linear Regression applied to normalized historical stock data can predict next-day closing prices within an acceptable margin of error.

## Study Design

Experimental software-based research conducted virtually with Python implementations and real-time data acquisition.

## Setting and Duration

The research is executed on a local machine with internet access for stock data retrieval over one month (April 1–30, 2025).

# Data Collection: Sample & Sampling Technique

## Sample Selection Criteria

- Stocks with at least one year of complete historical closing price data.
- Excluding stocks lacking sufficient or consistent historical data.

## Sampling Technique

Utilizes time-series data sampling with a sliding window of 60 days used as features to predict the next day's closing price.

Approximately 252 trading days per stock from the previous year form the dataset.

# Data Collection Procedure and Tools

1

## User Input

User enters a valid stock ticker via the Python tkinter GUI interface.

2

## Data Retrieval

Real-time stock data fetched from Yahoo Finance API using yfinance library.

3

## Data Preprocessing

Closing prices normalized using MinMaxScaler to scale data between 0 and 1 for modeling.

4

## Dataset Construction

Sliding window approach constructs feature-target pairs from 60 days' sequence to predict the next price.

5

## Model Training

Linear Regression model trained on 80% of data, tested on 20%.

6

## Prediction and Visualization

Predicted closing price displayed in GUI alongside historical and actual prices via line plots.

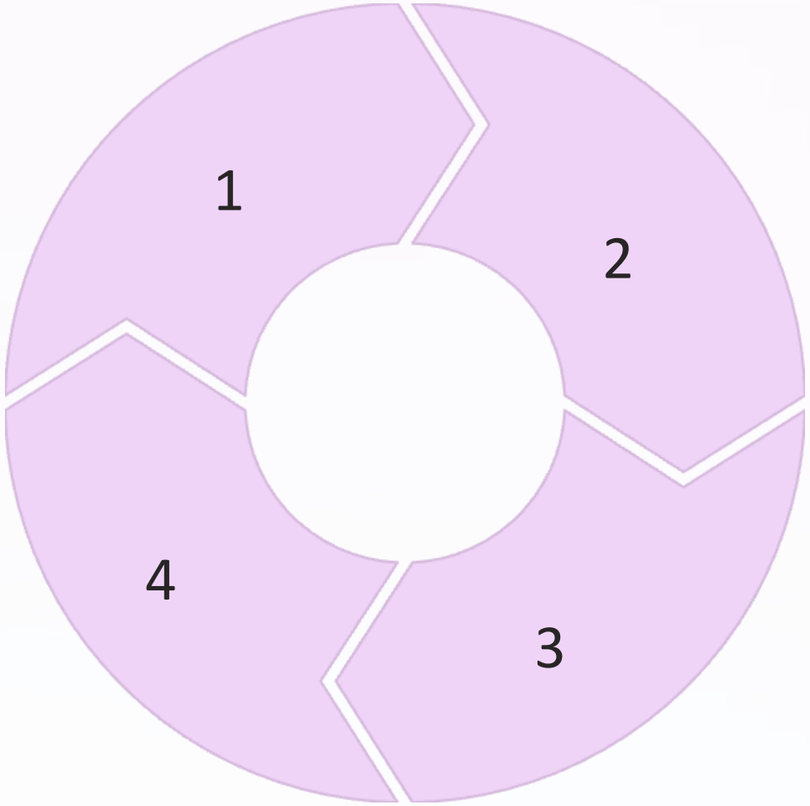
# Data Analysis Methods and Visualization

### Descriptive Statistics

Statistical measures including min, max, mean, and standard deviation analyze stock price distribution.

### Python Libraries

Use of scikit-learn, numpy, matplotlib, yfinance, and tkinter enables robust data handling, modeling, and UI presentation.

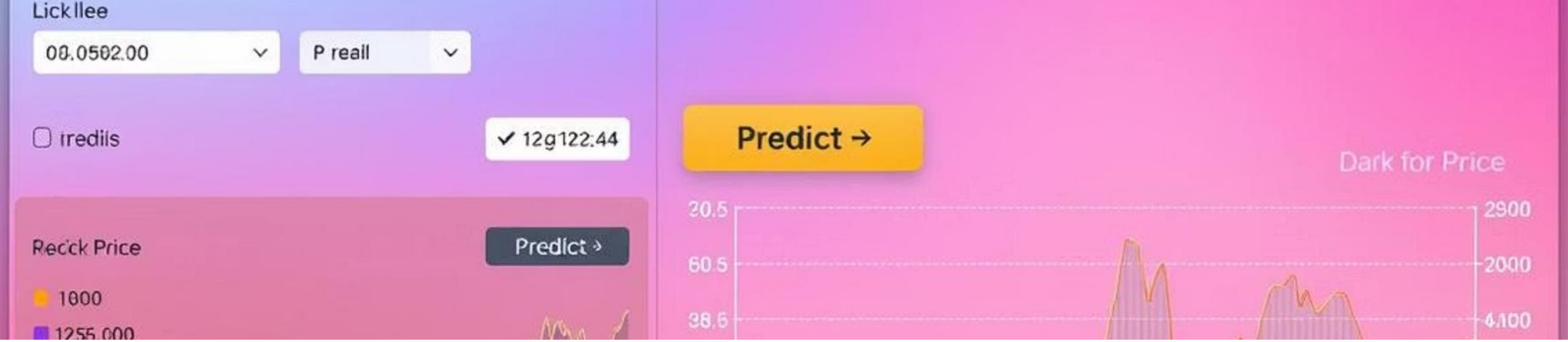



### Machine Learning Model

Linear Regression exploits linear relationships in normalized data for forecasting next-day prices.

### Visualization

Matplotlib plots illustrate historic trend lines juxtaposed with actual and predicted closing prices for intuitive interpretation.




# Software Architecture and GUI Design

### User Input Module

Allows seamless entry of stock ticker symbols and triggers data retrieval and prediction.

### Data Fetching & Processing

Automates fetching of historical and real-time data, normalization, and dataset formation.

### Prediction Engine

Executes Linear Regression model training and generates forecasts.

### Visualization Panel

Displays historical and predicted price trends using dynamic, embedded plots for user insight.

# References and Further Reading


- **Brownlee J.** Machine Learning Mastery with Python. 2016.
- **Geron A.** Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2019.
- **Yahoo Finance API Documentation.** https://pypi.org/project/yfinance/
- **Pedregosa F., Varoquaux G., et al.** Scikit-learn: Machine Learning in Python. J Mach Learn Res. 2011;12:2825–2830.


These key references provide foundational knowledge in machine learning techniques, Python-based implementation, and real-time financial data acquisition critical for replicating and extending this research.

# Thank You

Thank you for your attention and interest in this presentation.

We hope that the insights shared here will be valuable to your future projects.

1 **Open for Questions**

I'm now happy to answer any questions you may have.

2 **Further Inquiry**

Feel free to reach out after the presentation too.

3 **Future Research**

I hope this stimulates ideas for future research.