Stock Price Prediction – Project Documentation

1. Introduction

Stock markets are highly volatile and influenced by various economic, political, and social factors. Predicting stock prices has always been a challenging problem in finance. This project aims to build a stock price prediction system using Machine Learning, Time Series Analysis, and Technical Indicators.

The system allows users to input a stock ticker symbol, fetches the latest stock data, calculates Relative Strength Index (RSI), and applies different prediction models to forecast future prices.

2. Objectives

- To collect real-time stock data using Yahoo Finance API (yfinance).
- To preprocess the data and compute technical indicators (RSI) for better feature engineering.
- To implement and compare multiple models:
    - **Linear Regression**
    - **Random Forest Regressor**
    - **Decision Tree Regressor**
    - **ARIMA (Auto-Regressive Integrated Moving Average)**
- To evaluate model performance using the R^2 Score.
- To visualize RSI alongside stock prices for decision-making.
- To deploy the model in a simple Flask web application with charts

3. Tools & Technologies

- **Programming Language:** Python
- **Libraries:**
    - Data Processing → pandas, numpy
    - Machine Learning → scikit-learn, statsmodels
    - Technical Analysis → ta (Technical Analysis Library) or custom RSI function

    - Visualization → matplotlib, chart.js (frontend)
    - Data Source → yfinance
- **Web Framework:** Flask
- **Database:** SQLite (via Flask-SQLAlchemy)

4. System Architecture

1. **User Input** – Enter stock ticker (e.g., AAPL, TSLA).
2. **Data Collection** – Fetch stock data using yfinance.
3. **Feature Engineering** – Compute RSI and include it as an input feature.
4. **Preprocessing** – Handle missing values, split into training/testing sets.
5. **Model Training** – Train ML/ARIMA models with stock + RSI data.
6. **Prediction** – Forecast future prices.
7. **Visualization** – Display predicted values and RSI graphs on web UI.

5. Methodology

5.1 Data Collection

- Stock data is collected from Yahoo Finance with daily frequency.
- Features: Open, High, Low, Close, Volume, Adjusted Close.

5.2 Relative Strength Index (RSI)

- RSI is a momentum oscillator ranging from 0 to 100.
- Formula:

RSI=100−1/RS+100

where

- RS=Average Loss (n days)Average Gain (n days)
- **Interpretation:**
    - RSI > 70 → Overbought (possible price correction).
    - RSI < 30 → Oversold (possible upward trend).
- In this project, RSI (14-day) is calculated and used as an additional feature to improve predictions.

5.3 Data Preprocessing

- Convert date to datetime format.
- Normalize numerical features.
- Split into train (80%) and test (20%).

5.4 Model Implementation

- **Linear Regression** – baseline model.
- **Random Forest Regressor** – handles non-linearity, used with RSI.
- **Decision Tree Regressor** – interpretable but prone to overfitting.
- **ARIMA** – time-series model (does not use RSI but captures trends).

5.5 Model Evaluation

- Metric: **R² Score**.
- Example results:
    - Linear Regression: -33.42
    - Random Forest: -2.85
    - Decision Tree: -2.84
    - ARIMA: -2.83
- RSI helped in feature enrichment, but traditional ML models still struggled due to stock volatility.

6. Web Application

- **Frontend**: HTML + CSS + Chart.js for interactive visualization.
- **Backend**: Flask for data fetching, model predictions, RSI computation.
- **Features**:
    - Enter stock ticker.
    - Display stock price chart.
    - Show RSI line graph (0–100 scale).
    - Predict next-day price.

7. Results & Findings

- RSI successfully identifies overbought/oversold regions, which helps in decision-making.
- Among ML models, Random Forest performed relatively better but accuracy remained low.
- RSI is useful as a trading indicator rather than a pure predictor of stock price.

8. Future Enhancements

- Combine RSI with other technical indicators (MACD, Bollinger Bands).
- Use LSTM/GRU deep learning models that capture sequential patterns
- Deploy on cloud for real-time predictions.

9. Conclusion

This project demonstrates stock price prediction using machine learning, time series analysis, and RSI as a technical indicator. While ML models had limited predictive power, RSI provided valuable insights into market momentum. The system lays the foundation for more advanced hybrid models combining technical indicators with deep learning.

10. References

- Yahoo Finance API – https://pypi.org/project/yfinance
- Technical Analysis Library (ta) – https://technical-analysis-library-in-python.readthedocs.io
- Scikit-learn Documentation – https://scikit-learn.org
- Flask Documentation – https://flask.palletsprojects.com