

MINOR PROJECT

**STOCK PRICE
PREDICTION**



TABLE OF CONTENT

- Problem Statement
- Background Study
- Methodology
- Framework
- Timeline
- Block Diagram
- Gantt Chart



PROBLEM STATEMENT

Stock price prediction involves developing accurate forecasting models that can reliably predict future stock prices based on historical data and relevant market factors. This entails addressing challenges such as data preprocessing, feature selection, model training, and evaluation. The primary objective is to create predictive models that can capture complex patterns and trends in stock price movements, enabling investors and financial analysts to make informed decisions and optimize their investment strategies. Ultimately, the goal is to predict that stock moves up or down stock market environment.



BACKGROUND STUDY

1. Introduction: Highlight the significance of stock market prediction and its challenges.
2. Previous Research: Summarize key studies in stock market prediction for context.
3. Background Study: Provide context on machine learning and neural networks.
4. Data Preprocessing: Describe data collection, cleaning, and normalization steps.
5. Model Development: Explain the process of model selection, training, and tuning.
6. Prediction and Evaluation: Discuss how models are used for prediction and evaluation.



METHODOLOGY

- **Data Acquisition and Preprocessing**

Retrieve stock data from Yahoo Finance API, preprocess it with cleaning, normalization, and feature engineering.

- **Visualization**

Visualize predicted prices alongside actual data using graphs and indicators.

- **Model Development and Training**

Develop machine learning models, including feature extraction, architecture design, and hyperparameter tuning.

- **Evaluation**

Assess model performance using metrics like MSE, RMSE, and MAE on unseen data.

- **Prediction**

Utilize trained models to forecast future stock prices based on new data.

- **Deployment**

Deploy models as a web app for interactive stock price prediction.

FRAMEWORK

Pandas and NumPy

Essential libraries for data manipulation and numerical computing, facilitating preprocessing and analysis of financial data.

Keras

Deep learning frameworks for developing and training neural networks for complex modeling tasks.

Streamlit

An open-source framework for building interactive web applications in Python, simplifying the creation of data-driven web interfaces.

Matplotlib

Visualization libraries for creating static, interactive, and publication-quality plots and charts to explore data patterns and present model results.

Yahoo Finance API

Provides access to historical stock price data, company fundamentals, and market indices for training predictive models and generating forecasts.

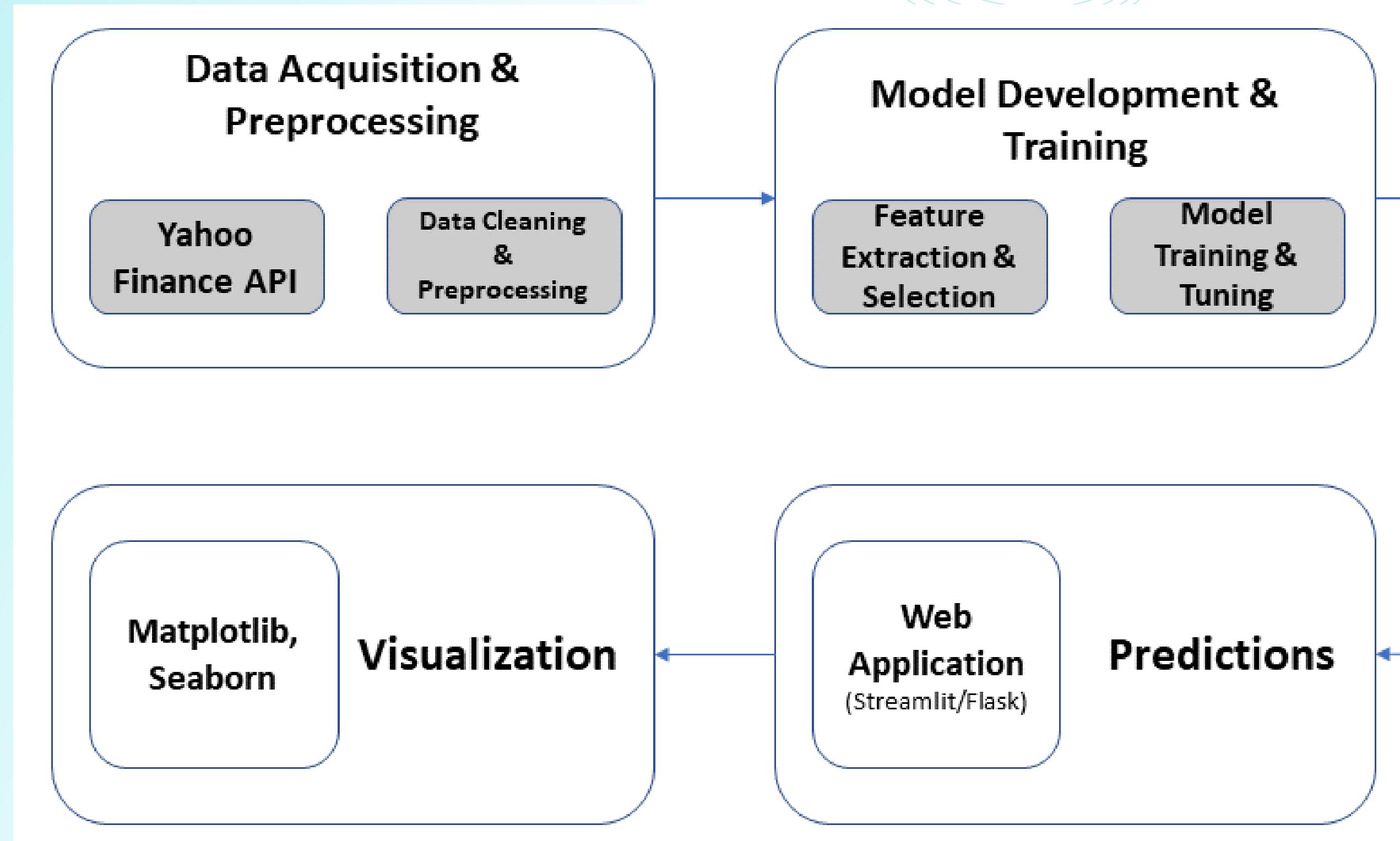
Python Programming

Used as the primary language for developing the predictive modeling framework and supporting infrastructure.

TIME LINE

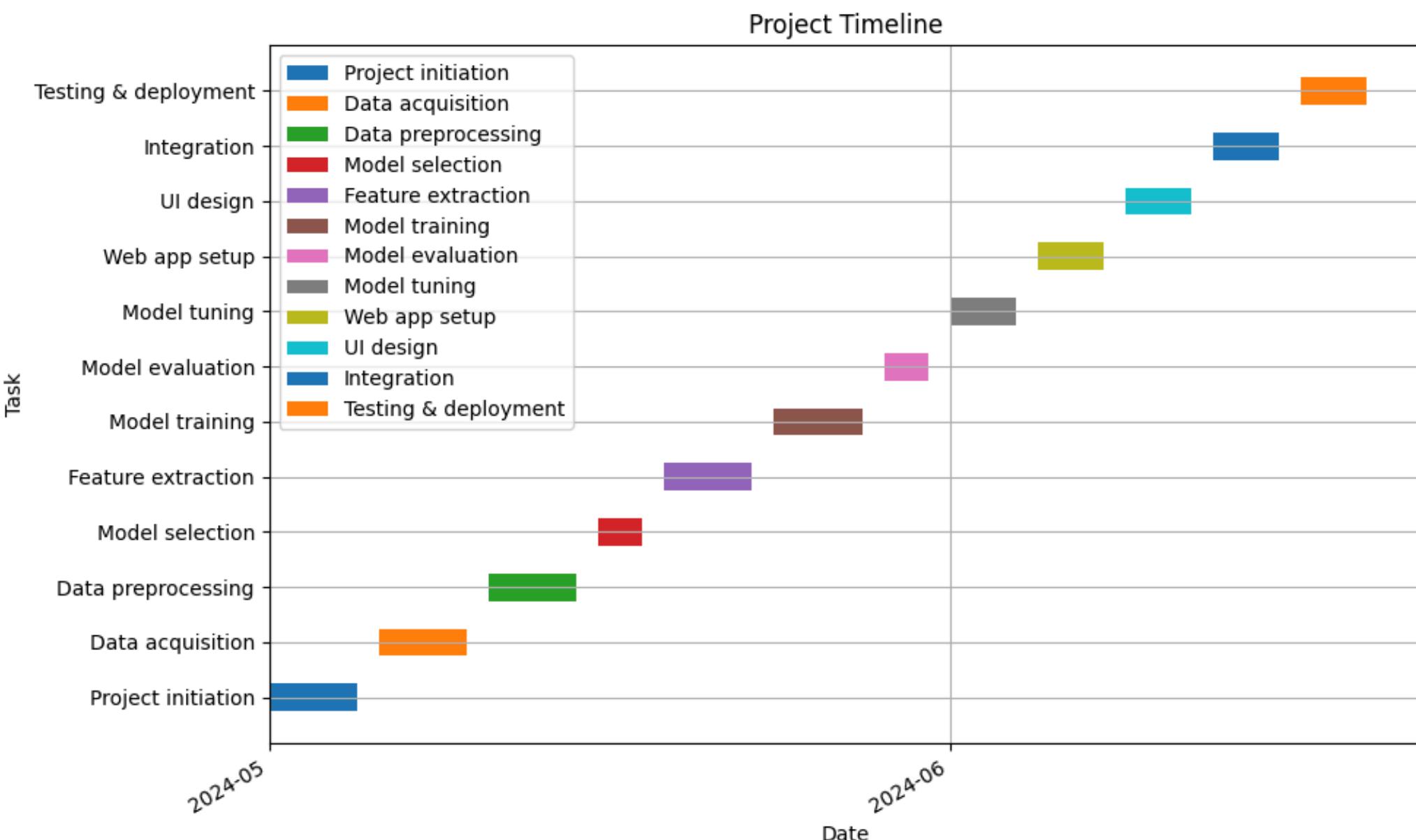
Weekly Task Done	Description
1. Project initiation	Define project scope, objectives, and timeline. Set up development environment.
2. Data acquisition	Retrieve historical stock price data using Yahoo Finance API.
3. Data preprocessing	Clean and preprocess the acquired data, handle missing values, and feature engineering.
4. Model selection	Choose appropriate machine learning or deep learning models for prediction.
5. Feature extraction	Extract relevant features from preprocessed data for input into the models.
6. Model training	Train selected models using prepared dataset.
7. Model evaluation	Evaluate model performance using appropriate metrics and validation techniques.
8. Model tuning	Fine-tune model hyperparameters to optimize performance.
9. Web app setup	Set up the web application framework for deploying the predictor.
10. UI design	Design user interface for the web application, including input forms and result display.
11. Integration	Integrate the trained model into the web application for real-time predictions.
12. Testing & deployment	Test the integrated system thoroughly and deploy it to a production environment.

BLOCK DIAGRAM



GANTT CHART

The Gantt chart outlines the project's timeline, divided into distinct tasks over a 12-week period. It begins with two weeks dedicated to Data Acquisition & Preprocessing, followed by four weeks for Model Development & Training, one week for Prediction Module implementation, three weeks for Web Application Development, and one week for Visualization Integration. Each task is allocated a specific duration, ensuring a structured approach to project management and facilitating efficient progress monitoring.



THANK YOU

