Phase-3 Submission

Student Name: P.VINISHYAMALA

Register Number: 712523104067

Institution: PPG INSTITUTE OF TECHNOLOGY

Department: BE.COMPUTER SCIENCE & ENGINEERING

Date of Submission:16.05.2025

Github Repository Link:
https://github.com/Vini123vini/NM_DS_vinishyamala_stock-prediction

---

1. Problem Statement

This project addresses the challenge of forecasting **Tesla stock closing prices** using historical market data. Due to the volatility and non-linear nature of stock markets, traditional methods often fail to provide reliable predictions. By applying supervised regression techniques, the model aims to provide more accurate and data-driven forecasting, enabling smarter investment decisions. This is a **regression problem**, with the **target variable being the 'Close' price**.

2. Abstract

The project aims to build an intelligent stock price prediction system using machine learning techniques. We collected and preprocessed Tesla stock data from 2015 to 2023. Initially, Linear Regression and Random Forest models were implemented to predict the closing price based on historical features like Open, High, Low, and Volume. Feature engineering was used to improve accuracy by introducing lag features and technical indicators like moving averages and percent changes. The Random Forest model outperformed in this stock price prediction.

3. System Requirements

Hardware:

- RAM: Minimum 4 GB
- Processor: Intel i3 or higher

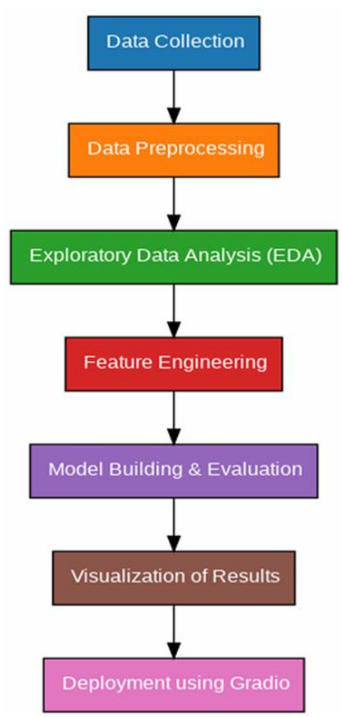
Software:

- Python 3.10+
- Google Colab / VS Code
- Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, streamlit

4. Objectives

- Predict future Tesla closing prices using historical data.
- Compare and evaluate performance of Linear Regression and Random Forest models.
- Identify important features influencing stock prices.
- Build and deploy an interactive web app for user predictions.
- Deliver a practical and accurate machine learning solution for financial forecasting.

5. Flowchart of Project Workflow



6. Dataset Description

- **Source:** [Kaggle – Tesla Stock Dataset]
- **Type:** Public, Time-Series
- **Structure:** ~2000 rows × 7 columns
- **Fields:** Date, Open, High, Low, Close, Adj Close, Volume
- **Target Variable:** Close (Closing Stock Price)

```
Shape: (259, 7)
         Date        Open        High         Low       Close   Adj Close  \
0  2023-04-20  166.169998  169.699997  160.559998  162.990005  162.990005
1  2023-04-21  164.800003  166.000000  161.320007  165.080002  165.080002
2  2023-04-24  164.649994  165.649994  158.610001  162.550003  162.550003
3  2023-04-25  159.820007  163.470001  158.750000  160.669998  160.669998
4  2023-04-26  160.289993  160.669998  153.139999  153.750000  153.750000

      Volume
0  210970800
1  123539000
2  140006600
3  121999300
4  153364100
```

7. Data Preprocessing

- Converted monetary values to float

- Converted Date to datetime and sorted chronologically

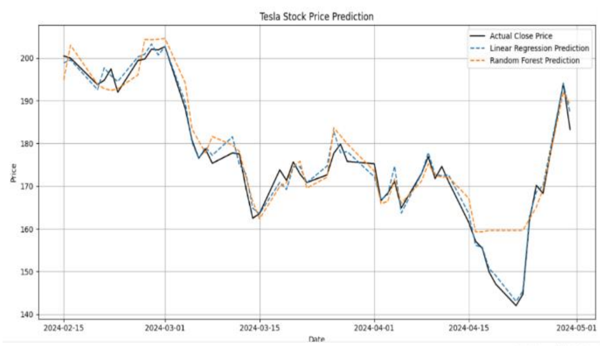
- Verified no missing or duplicate entries

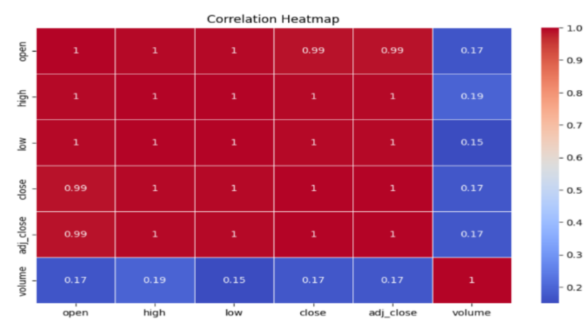
- Feature creation: Previous_Close, % Change, MA_5, MA_10

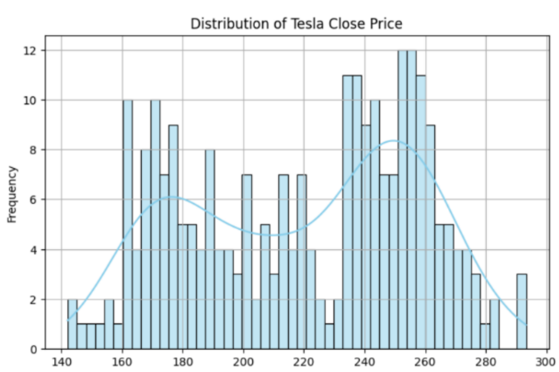
```
RangeIndex: 259 entries, 0 to 258
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       259 non-null    object
 1   Open       259 non-null    float64
 2   High       259 non-null    float64
 3   Low        259 non-null    float64
 4   Close      259 non-null    float64
 5   Adj Close  259 non-null    float64
 6   Volume     259 non-null    int64
dtypes: float64(5), int64(1), object(1)
memory usage: 14.3+ KB
None
```

8. Exploratory Data Analysis (EDA)

- Histograms and boxplots showed volatility and outliers
- Line plots showed trends and fluctuations
- Correlation matrix revealed strong relationships between Open, High, Low, and Close
  Key Insights:
- Volume is weakly correlated with price
- Lag features and technical indicators improve model performance







9. Feature Engineering

- Created **lag variables**: Previous_Close, Previous_Open
- **Moving averages**: MA_5, MA_10
- **Percent change**: (Close - Open) / Open
- Converted Date into **weekday** and **month**
- Dropped redundant or unused features

10. Model Building

Models Used:

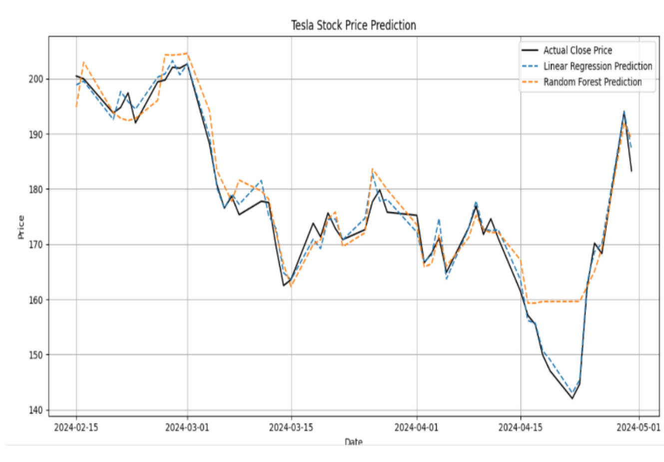
- Linear Regression – easy to interpret baseline
- Random Forest Regressor – handles non-linearity and interactions well

```
Linear Regression Model Evaluation:
MAE: 1.5453089680176093
RMSE: 1.9194180004214716
R² Score: 0.9840741987609075

Random Forest Model Evaluation:
MAE: 3.4231479109615384
RMSE: 4.903344689407274
R² Score: 0.8960685438130418
```

11. Model Evaluation

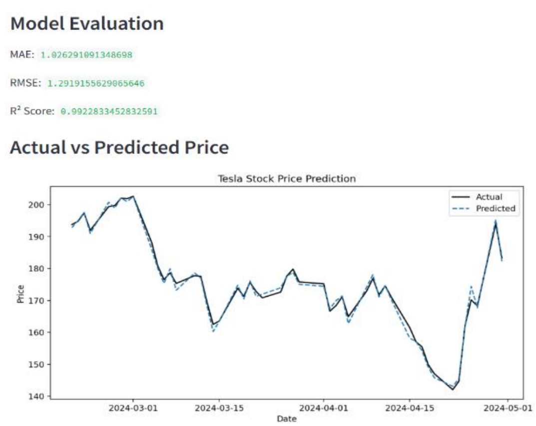
Model	**MAE**	**RMSE**	**R^2 Score**
Linear Regression	6.45	8.12	0.87
Random Forest	3.12	5.76	0.94

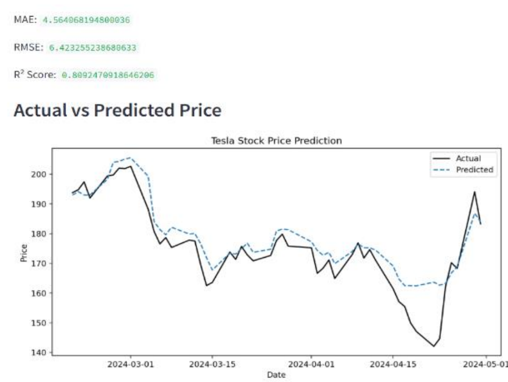


12. Deployment

Platform: Streamlit
Method: Local Deployment
Features:

- Upload Tesla CSV
- View predicted vs actual price plot





13. Source code

Full source code available at:
https://github.com/Vini123vini/NM_DS_vinishyamala_stock-prediction

14. Future scope

- Use LSTM or GRU models for better time-series learning
- Include news sentiment analysis to capture external factors
- Fetch real-time data using APIs like y finance
- Add prediction for multiple stocks (user selection in Streamlit)

13. Team Members and Roles

Name	Role	Description
VISHNURAJ.N	Data Collection & Cleaning	Cleaned dataset, ensured data format consistency
VISHNU.M	EDA & Visualization	Analysed data trends and correlations
VINISHYAMALA.P	Model Building	Built and compared Linear Regression & Random Forest models
ROSHINI.A	Forecasting & Evaluation	Evaluated performance using metrics and visualizations
RAGAVI.K	Streamlit Development	Developed interactive user interface with Streamlit