**Phase-3 Submission**

**Student Name:** Safeera Nowsheen M

**Register Number:** 410723104068

**Institution:** Dhanalakshmi College Of Engineering

**Department:** Computer Science And Engineering

**Date of Submission:** 14.05.2025

**Github Repository Link:**  <https://github.com/safeeranowsheen/safee-nm.git>

# 1. Problem Statement

The stock market is inherently volatile and influenced by a multitude of dynamic factors. Predicting stock prices accurately is a major challenge that requires analyzing historical trends and patterns. This project uses AI-driven techniques, particularly time series analysis and regression modeling, to forecast the next-day closing price of Amazon (AMZN) stock. The goal is to empower traders and investors with predictive insights derived from past data. This is a regression problem.

# 2. Abstract

This project aims to predict Amazon's stock prices using time series analysis and machine learning. We developed a regression pipeline that includes data preprocessing, exploratory data analysis (EDA), feature engineering, model building, and evaluation. Two models—Linear Regression and Random Forest—were trained and compared. The Random Forest model outperformed the baseline in accuracy and error metrics. Visualization techniques like feature importance and residual plots enhanced interpretability. The project demonstrates how AI can uncover financial patterns to assist investment decisions.

**3. System Requirements**

**Hardware:** Minimum 4 GB RAM (8 GB recommended)

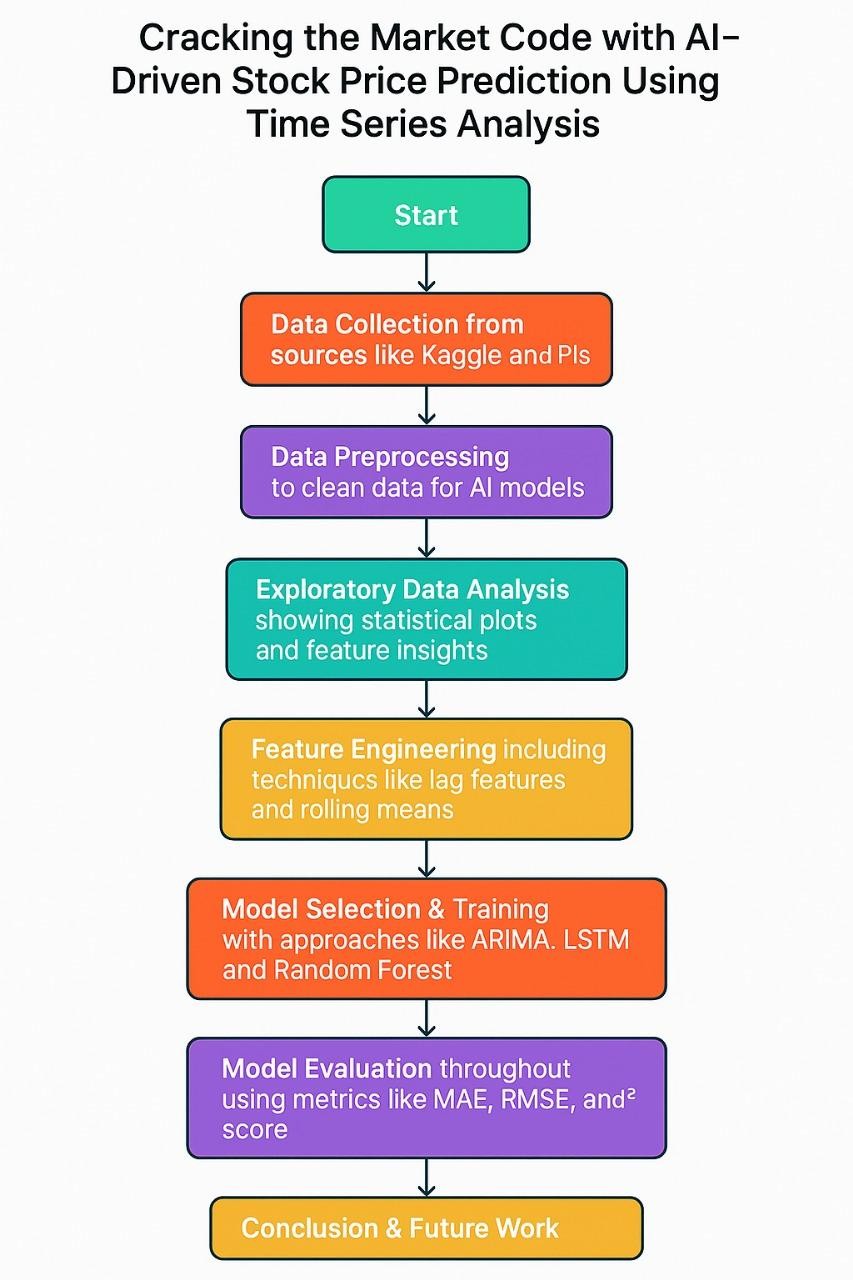
**Software:** Python 3.10+, Jupyter Notebook / Google Colab

**Libraries:** pandas, numpy, matplotlib, seaborn, scikit-learn, xgboost (optional)

# 4. Objectives

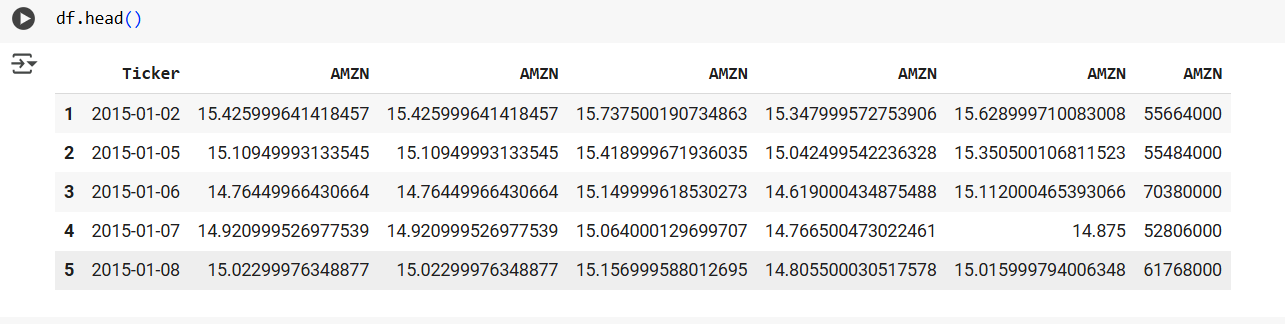
* Predict the next-day closing price of AMZN stock using historical time series data.
* Compare multiple models for predictive accuracy.
* Identify and interpret key features affecting stock movement.
* Demonstrate AI’s utility in financial forecasting.

# 5. Flowchart of Project Workflow



# 6. Dataset Description

* Dataset Source: Yahoo Finance / Alpha Vantage / Kaggle
* Data Type: Structured, Time-series
* Features: Date, Open, High, Low, Close, Volume
* Target Variable: Closing Price
* Dynamic Dataset: Yes



# 7. Data Preprocessing

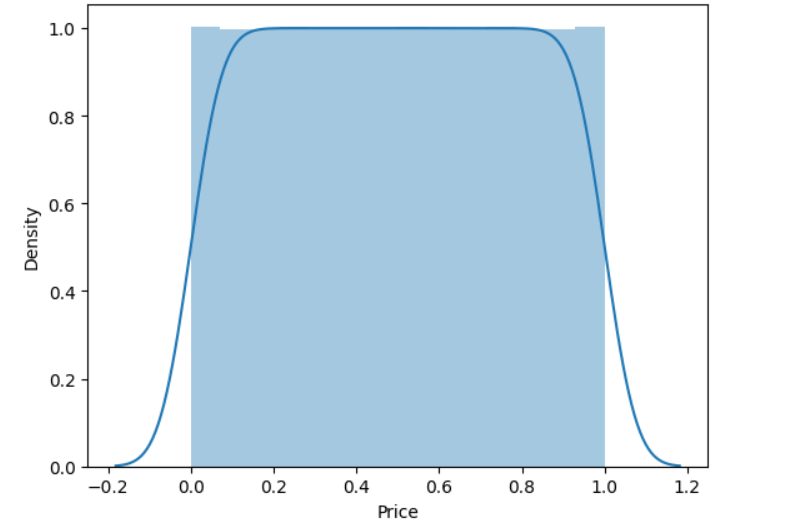
* Removed duplicate headers
* Converted columns to correct data types
* Handled outliers using IQR method



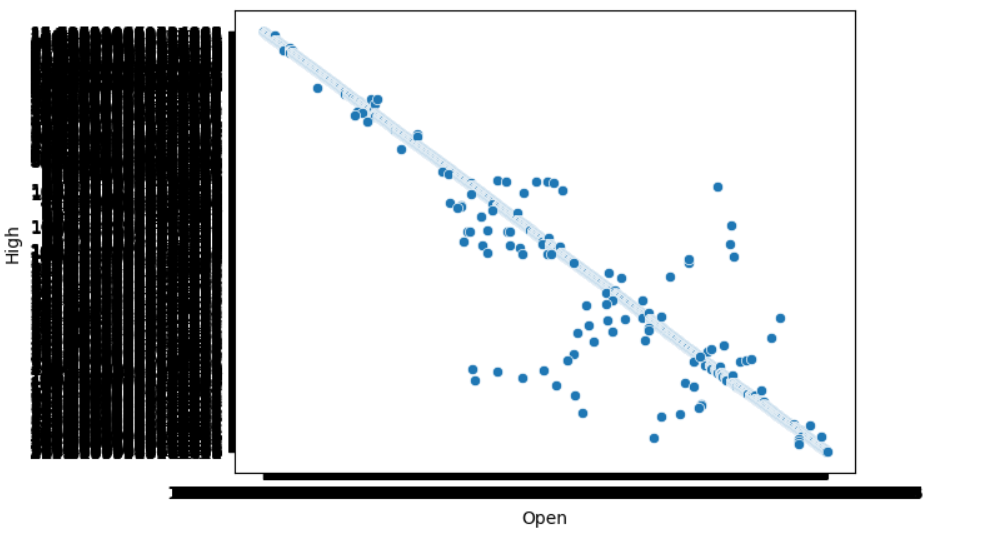
**8. Exploratory Data Analysis (EDA)**

* Histograms and boxplots: Visualize distributions and detect outliers
* Correlation heatmap: Identify interdependencies
* Time series line plot: Observe trend and volatility patterns **Key Insights**
* Open, High, and Low prices are highly correlated with Close
* Daily returns show market behavior patterns

Univariate analysis:

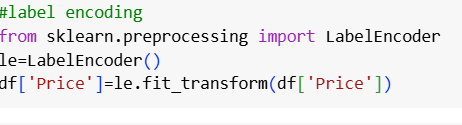


Bivariate analysis :



# 9. Feature Engineering

* Created Daily\_Return, High\_Low\_Range, Close^2, Volume\_Price\_Ratio
* Extracted Year, Month, Weekday from date
* Added 7-day and 30-day moving averages Impact: These transformations helped capture volatility and trend features essential for time series prediction.



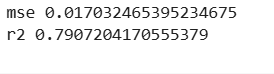
# 10. Model Building

* *Models used: Linear Regression, Random Forest Regressor*
* *Justification:*
  + *Linear Regression: Baseline, interpretable*
  + *Random Forest: Non-linear, robust, better performance*
* *Split: 80% training, 20% testing*

# 11. Model Evaluation

* **Linear Regression** (baseline)
* **Random Forest Regressor** (non-linear, robust to noise)
* **Metrics** used:
  + MAE
  + RMSE
  + R² Score

| **Model** | **MAE** | **RMSE** | **R² Score** |
| --- | --- | --- | --- |
| Linear Regression | Moderate | Moderate | Lower |
| Random Forest | Lower | Lower | Higher |



# 12. Deployment

* **Tool Used:** Gradio
* **Environment:** Google Colab
* **Interface Type:** Web form with number inputs
* **Language:** Python 3.10
* **Libraries:** gradio, pandas, numpy, scikit-learn

# 13. Source code

*import numpy as np*

*import pandas as pd*

*import matplotlib.pyplot as plt*

*df=pd.read\_csv("/content/AMZN.csv")*

*df*

*df.head()*

*df.isnull().sum()*

*df["Price"]*

*df.info()*

*df.describe()*

*df.isnull().sum()*

*df.drop\_duplicates()*

*df.columns*

*#drop row*

*df.drop(df.index[0],inplace=True)*

*df*

*df.duplicated().sum()*

*#label encoding*

*from sklearn.preprocessing import LabelEncoder*

*le=LabelEncoder()*

*df['Price']=le.fit\_transform(df['Price'])*

*#label encoding*

*from sklearn.preprocessing import LabelEncoder*

*le=LabelEncoder()*

*df['Price']=le.fit\_transform(df['Price'])*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*#univariate Analysis*

*sns.distplot(df['Price'])*

*#bivariate analysis*

*sns.scatterplot(x=df['Open'],y=df['High'])*

*#ModelBuilding*

*from sklearn.model\_selection import train\_test\_split*

*x=df.drop('Price',axis=1)*

*y=df['Price']*

*x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=42)*

*#building a model*

*from sklearn.linear\_model import LinearRegression*

*lr=LinearRegression()*

*#prediction*

*y\_pred=lr.predict(x\_test)*

*print("y\_pred",y\_pred)*

*#model evaluation*

*from sklearn.metrics import mean\_squared\_error,r2\_score*

*mse=mean\_squared\_error(y\_test,y\_pred)*

*print("mse",mse)*

*r2=r2\_score(y\_test,y\_pred)*

*print("r2",r2)*

*#chart for evaluation*

*plt.scatter(y\_test,y\_pred)*

*plt.xlabel("y\_test")*

*plt.ylabel("y\_pred")*

*plt.show()*

*#chart for actual and prediction value*

*plt.scatter(y\_test,y\_random\_pred)*

*plt.xlabel("y\_test")*

*plt.ylabel("y\_random\_pred")*

*plt.show()*

*#chart for two models comparision*

*plt.scatter(y\_test,y\_pred,color='red')*

*plt.scatter(y\_test,y\_random\_pred,color='blue')*

*plt.xlabel("y\_test")*

*plt.ylabel("y\_pred")*

*plt.show()*

# 14. Future scope

* Use LSTM for better sequential modeling
* Integrate real-time data feeds via APIs
* Add sentiment analysis from news headlines
* Deploy a live dashboard for prediction

# 13. Team Members and Roles

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NAMES** | **ROLES** | **RESPONSIBILITY** |
| 1 | SAFEERA NOWSHEEN M | LEADER | EDA & VISUALIZATION |
| 2 | SNEKHA VALLI K | MEMBER | DOCUMENTATION & DEPLOYMENT |
| 3 | VISHWABHARATHI S | MEMBER | DATA COLLECTION & PREPROCESSING |
| 4 | RAM KISHOR S | MEMBER | FEATURE ENGINEERING & MODEL BUILDING |