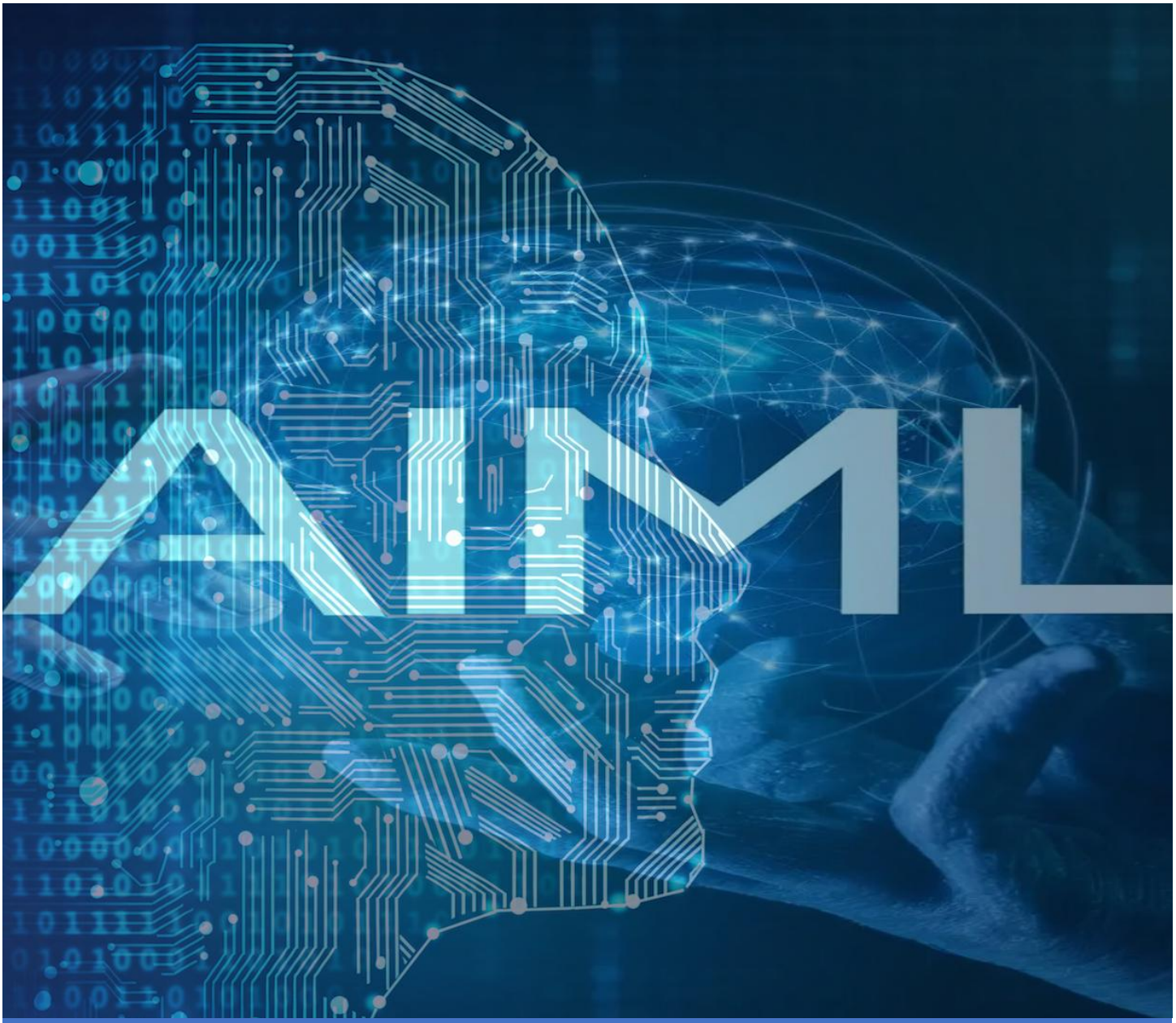


STOCK PREDICTION REPORT

AI-ML PROJECT REPORT

2384550032_CHARAN VENKAT

2/5/25

FROM BATCH : 01

STOCK-PREDECTION

Name: **GUTTULA CHARAN VENKAT.**

Roll Number**: 2384550032.**

Institution: **ADITYA SCHOOL OF ANIMATION 010141 (A UNIT OF ADITYA DGREE COLLEGE), ASILMETTA VISAKHAPATNAM-530002 .**

Course/Department Name: **B.sc ANIMATION.**

Submission Date: **05-02-2025.**

Batch: **01**

ABSTRACT

This project focuses on predicting stock prices using machine learning techniques. The dataset used for this project contains historical stock prices, including Open, High, Low, Close, Adjusted Close, and Volume data. The goal is to develop a predictive model to estimate future stock prices based on historical trends.

The project applies data preprocessing techniques, exploratory data analysis (EDA), and a machine learning model (Linear Regression) to make predictions. The results demonstrate the feasibility of using historical data for stock price forecasting, although more advanced models may yield better accuracy. Future improvements could incorporate deep learning techniques and additional features.

TABLE OF CONTENTS

1.INTRODUCTION

The ability to predict stock prices is valuable for investors and financial analysts. This project aims to develop a machine learning model to forecast stock prices based on historical data. By analyzing trends and patterns, the model provides insights into potential future price movements.

2. FRAMEWORKS

- **Programming Language**: Python
- **Libraries Used**: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn
- **Machine Learning Model**: Linear Regression

3. IMPLEMENTATION/EXECUTION

- **Step 1**: Data Collection from Yahoo Finance

```
# Load dataset
df = pd.read_csv("yahoo_stock.csv")
```

- **Step 2**: Data Preprocessing (handling missing values, converting date format, sorting data)

```
# Check for missing values
print(df.isnull().sum())

# Drop rows with missing values (if any)
df = df.dropna()

# Convert the 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Sort the dataset by date
df = df.sort_values('Date')
```

- **Step 3**: Exploratory Data Analysis (EDA)

```
# Display dataset info
print(df.info())
print(df.shape())
```

- **Step 4**: Feature Selection

```
Click to add a breakpoint & Splitting Data
X = df[['High', 'Low', 'Open', 'Volume', 'Adj Close']]
y = df['Close']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

- **Step 5**: Model Training and Evaluation

```
Click to add a breakpoint essionModel
model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression
LinearRegression()
```

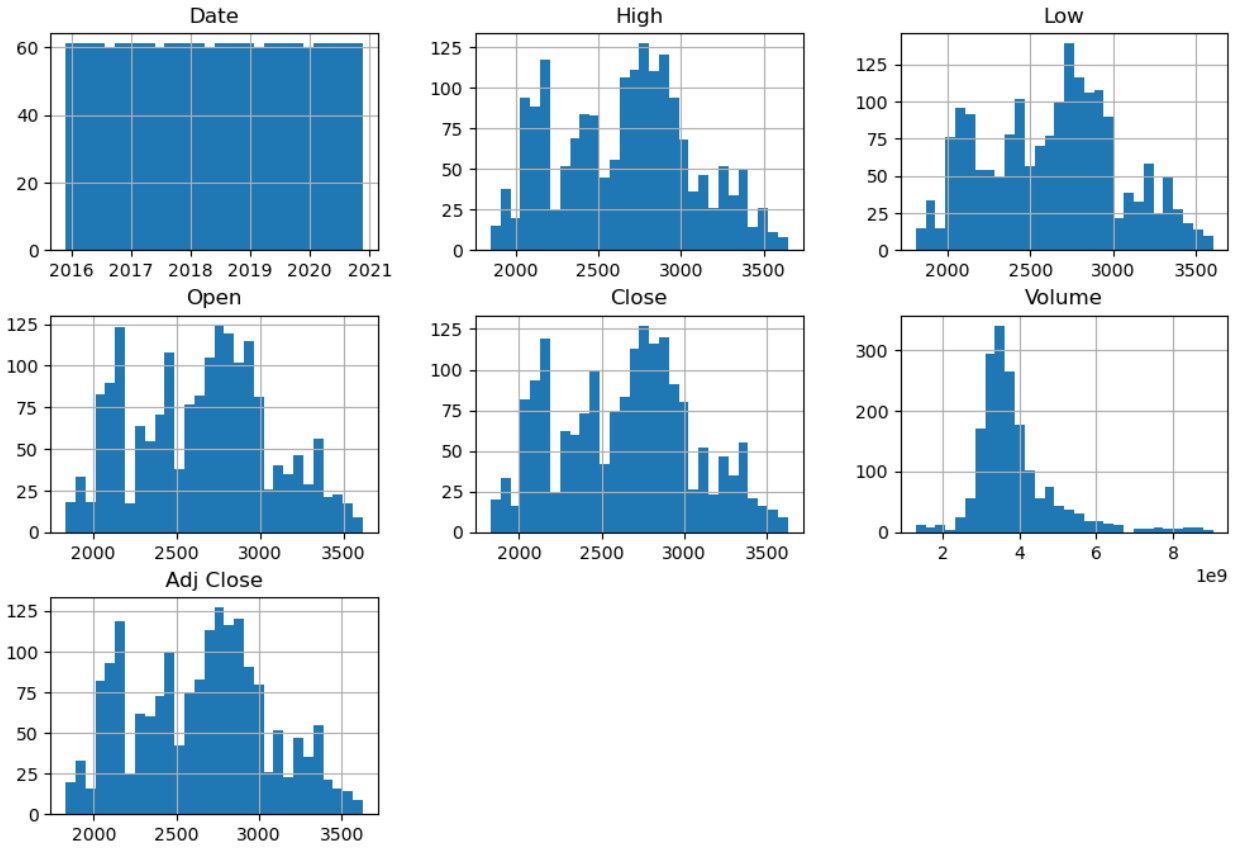
```
# Model Prediction
y_pred = model.predict(X_test)
# Model Evaluation
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Absolute Error: {mae}")
print(f"Mean Squared Error: {mse}")
print(f"R-squared Score: {r2}")
```

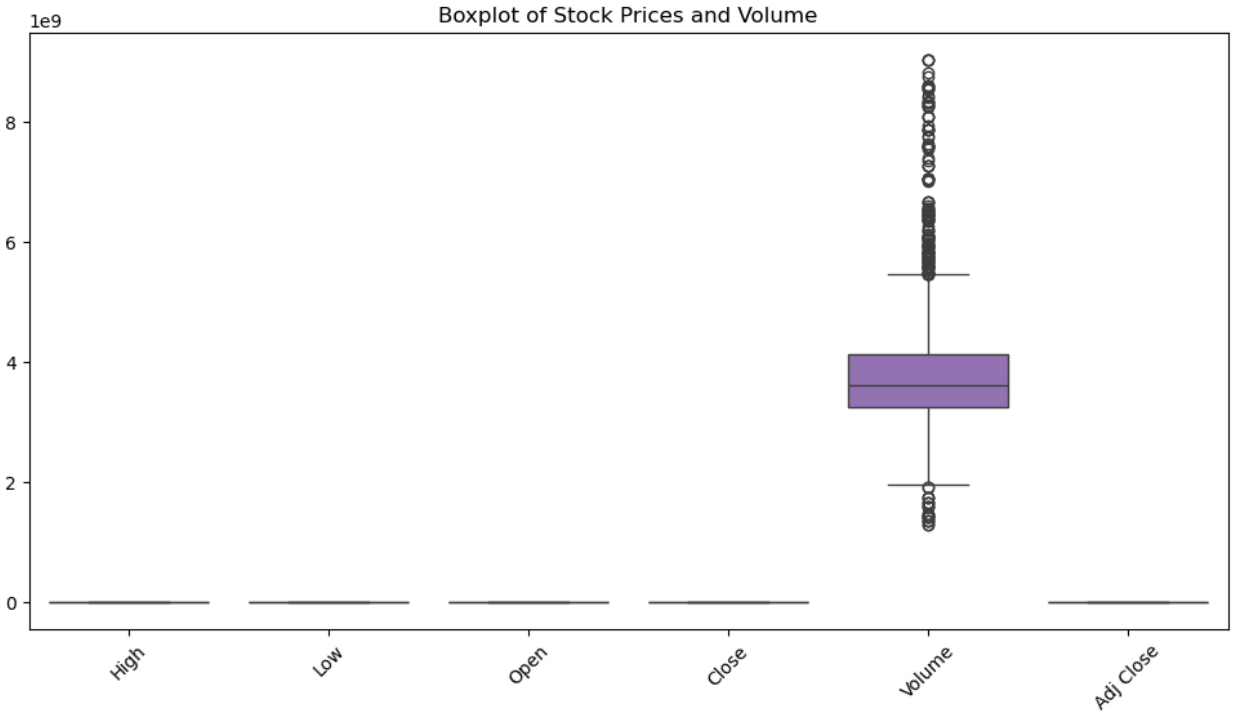
- **Step 6**: Performance Analysis

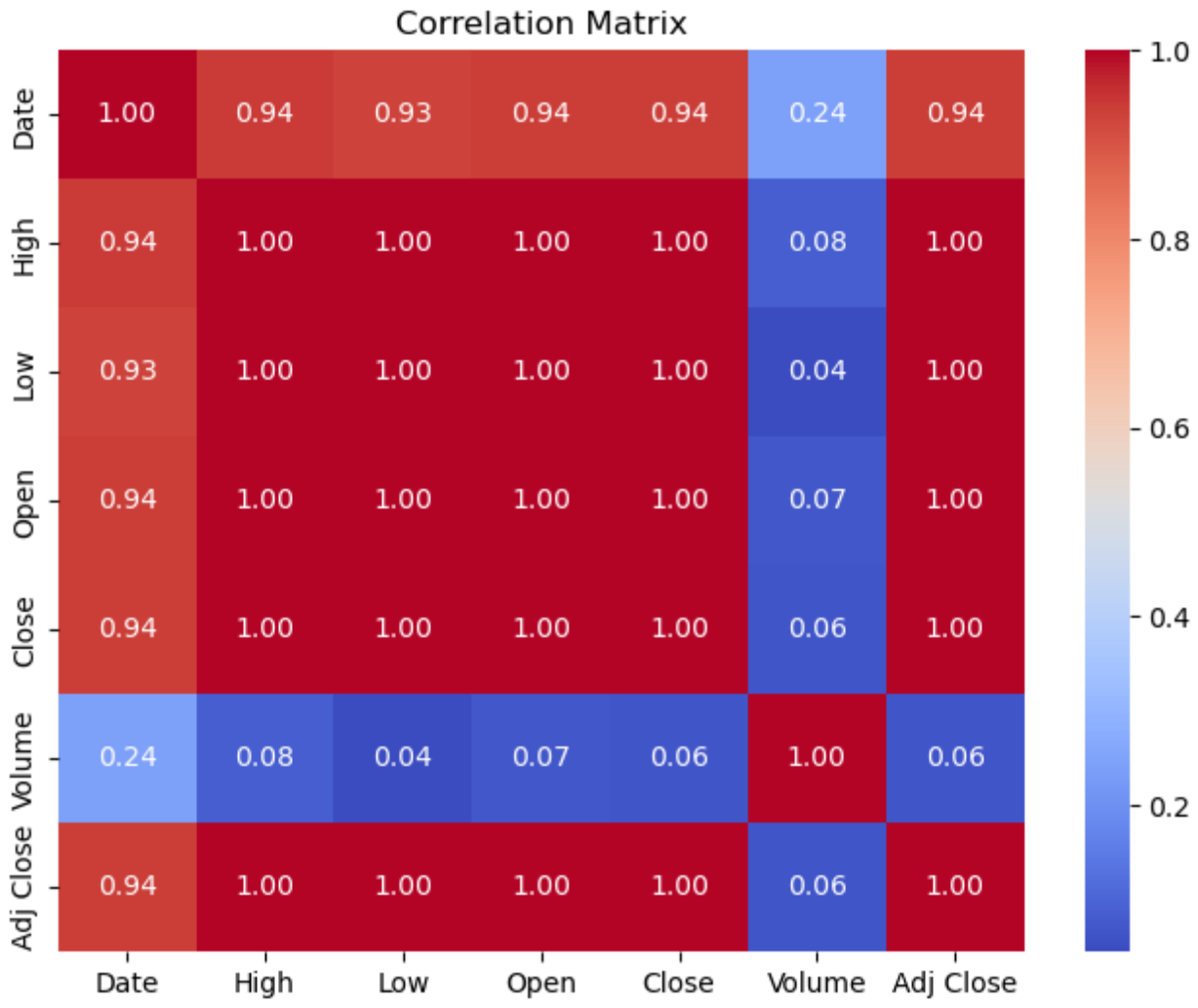
```
#model accuracy
accuracy = model.score(X_test, y_test) * 100
print(accuracy)
```

Execution Details

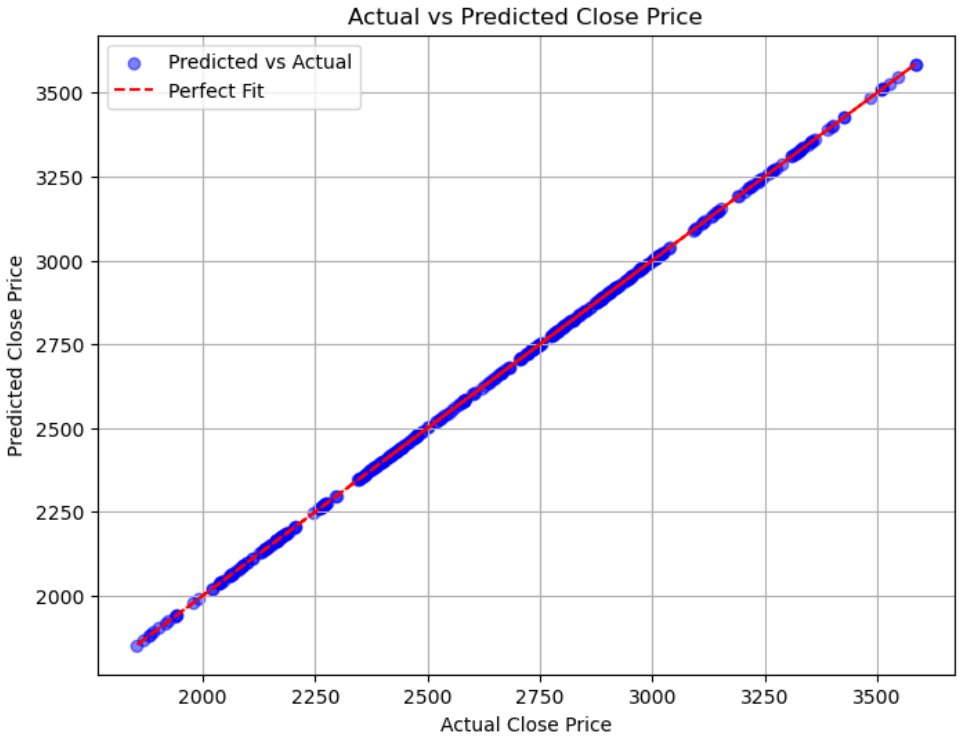
- **Dataset**: 1,825 stock market records
- **Training-Testing Split**: 80%-20%
- **Evaluation Metrics**: Mean Absolute Error (MAE), Mean Squared Error (MSE), R-Squared Score (R^2)
- **Visualization**: Plots of stock price trends, correlation analysis


Date
High
Low
Open
Close
Volume
Adj Close
1e9


Boxplot of Stock Prices and Volume
1e9
8
6
4
2
0
High
Low
Open
Close
Volume
Adj Close



4. Results and Discussion



The model provides a reasonable approximation of stock prices, but further enhancements are needed for improved accuracy. The R^2 score indicates that while the model captures general trends, additional advanced techniques like LSTMs or Random Forests could yield better results.

5. Conclusion

The Linear Regression model serves as a baseline for stock price prediction. Future work can explore deep learning techniques (LSTMs, GRUs) for time-series forecasting. Incorporating macroeconomic indicators and real-time data could further enhance predictions.

6. References

- Yahoo Finance for stock data
- Scikit-learn for model implementation