AN MINI PROJECT REPORT

on

ASIAN PAINT Stock Performance

As a part of

Data Science (CE0630)

Submitted by

Student Name

YASH JAIN (IU2241230235)

In fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNOLOGY AND ENGINEERING INDUS

UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ

AHMEDABAD-382115, GUJARAT, INDIA,

WEB: www.indusuni.ac.in

APRIL 2025

PROBLEM STATEMENT : "To analyze and model the historical stock data of ASIAN PAINT in order to uncover trading patterns, assess performance metrics, and explore predictive insights for informed decision-making**.**

1. Introduction:

In the dynamic world of financial markets, understanding the behavior of stock prices is essential for investors, analysts, and policymakers. ASIAN PAINT, being one of the leading companies in the Indian stock market, presents a valuable opportunity to study real-time financial data and draw meaningful insights.

This project aims to perform a comprehensive analysis of ASIAN PAINTs stock data using various statistical and machine learning techniques. By leveraging descriptive statistics, we explore historical trends in price movements, trading volume, and market volatility. Inferential statistics help us test hypotheses and make generalizations from sample data. Furthermore, a predictive model is built using machine learning algorithms to forecast closing prices based on other trading indicators.

2. DataSet Overview :

Dataset Name: ASIANPAINT.csv

Source: National Stock Exchange (NSE) / Bombay Stock Exchange (BSE)

Time Frame: Multiple years (exact range based on data available)

Total Records: 2001 rows

Total Features: 15 columns

3. Key Features:

1. Price-Related Metrics

Open: The price at which the stock started trading for the day.

High: The maximum price reached during the trading session.

Low: The lowest price reached during the trading session.

Close: The final price at which the stock traded before market close.

Last: The last price at which the stock was traded (not necessarily the closing price).

Prev Close: The previous day's closing price, used to analyze price gaps and trends.

VWAP (Volume Weighted Average Price): Average price weighted by volume, used for trade benchmarking.

2. Volume & Value Indicators

Volume: Total number of shares traded. High volume often indicates strong investor interest or volatility.

Turnover: Total trading value (in INR), representing the total money exchanged for the stock.

3. Delivery Metrics

Deliverable Volume: The number of shares that were actually transferred to a buyer's demat account.

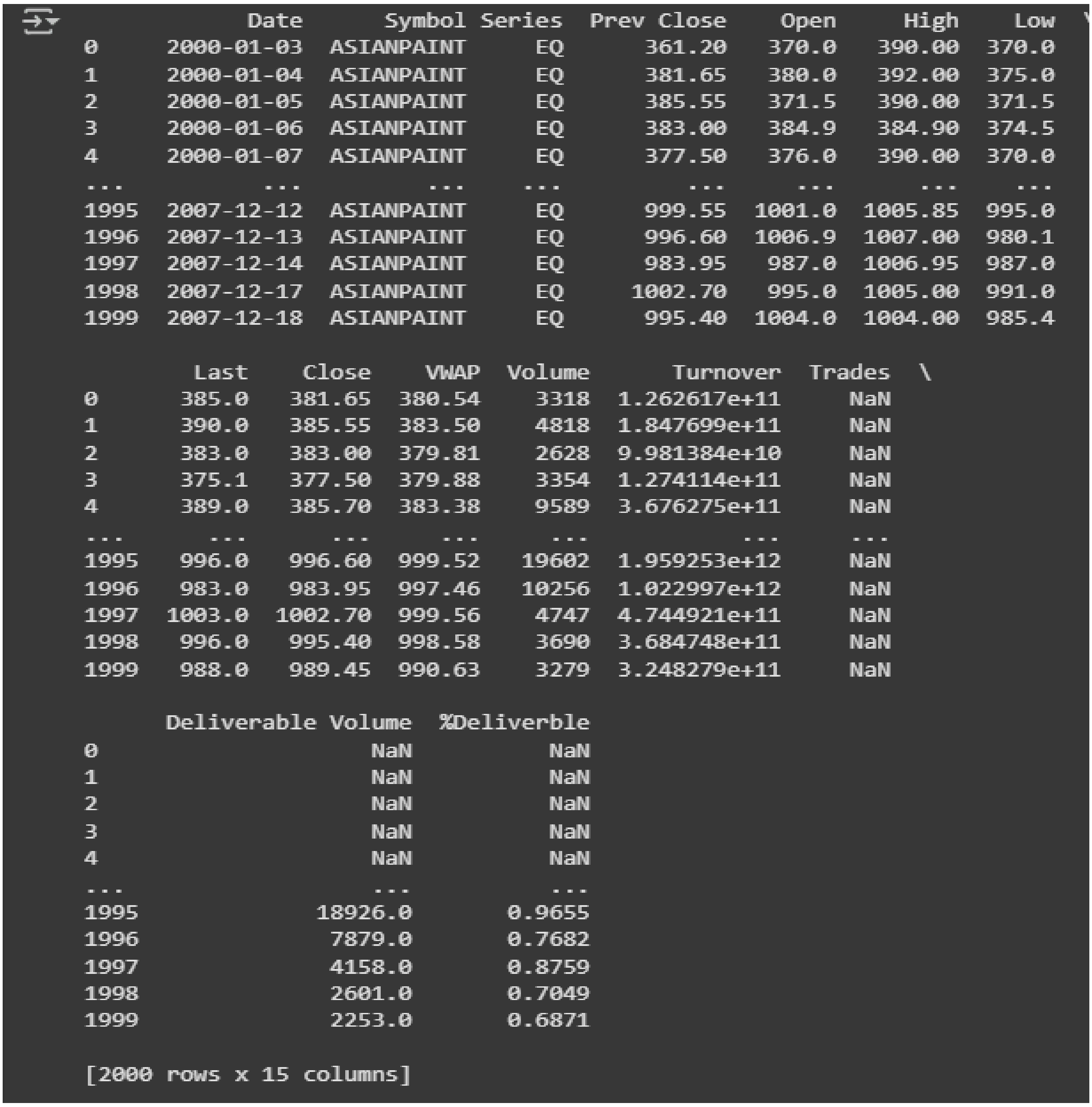
%Deliverable: The percentage of total traded volume that was actually delivered, which indicates long-term investment interest versus intraday trading.

4. OBJECTIVE:

The objective of this project is to perform a comprehensive analysis of ASIAN PAINTs historical stock market data using descriptive and inferential statistics, and to apply machine learning models to predict stock price behavior.Machine learning is used in this project to predict the Close price of ASIAN PAINT stock based on other available trading parameters.

CODE:

```
FROM GOOGLE.COLAB IMPORT FILES

import pandas as pd

df = pd.read_csv("ASIANPAINT.csv")

print(df.head(2000))
```

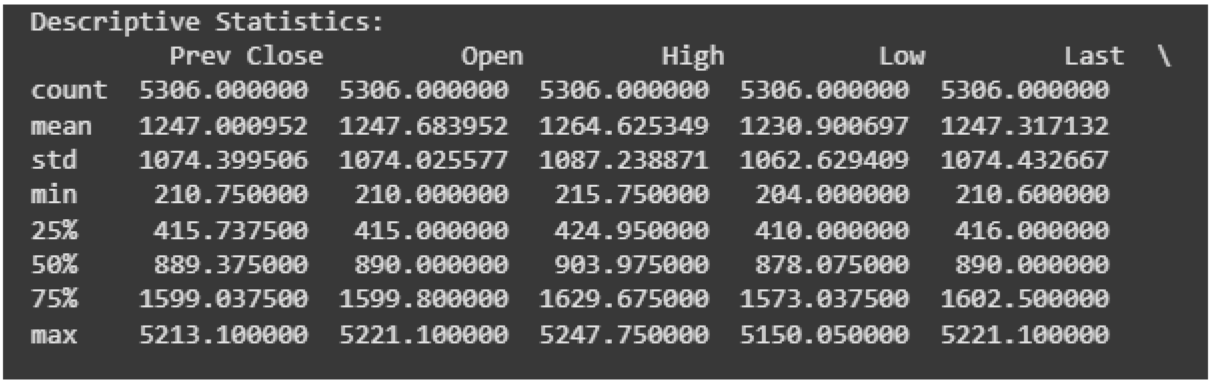


```
            Date      Symbol Series  Prev Close    Open     High    Low
0     2000-01-03  ASIANPAINT     EQ      361.20   370.0   390.00  370.0
1     2000-01-04  ASIANPAINT     EQ      381.65   380.0   392.00  375.0
2     2000-01-05  ASIANPAINT     EQ      385.55   371.5   390.00  371.5
3     2000-01-06  ASIANPAINT     EQ      383.00   384.9   384.90  374.5
4     2000-01-07  ASIANPAINT     EQ      377.50   376.0   390.00  370.0
...          ...         ...    ...         ...     ...      ...    ...
1995  2007-12-12  ASIANPAINT     EQ      999.55  1001.0  1005.85  995.0
1996  2007-12-13  ASIANPAINT     EQ      996.60  1006.9  1007.00  980.1
1997  2007-12-14  ASIANPAINT     EQ      983.95   987.0  1006.95  987.0
1998  2007-12-17  ASIANPAINT     EQ     1002.70   995.0  1005.00  991.0
1999  2007-12-18  ASIANPAINT     EQ      995.40  1004.0  1004.00  985.4

        Last    Close    VWAP  Volume      Turnover  Trades  \
0      385.0   381.65  380.54    3318  1.262617e+11     NaN
1      390.0   385.55  383.50    4818  1.847699e+11     NaN
2      383.0   383.00  379.81    2628  9.981384e+10     NaN
3      375.1   377.50  379.88    3354  1.274114e+11     NaN
4      389.0   385.70  383.38    9589  3.676275e+11     NaN
...      ...      ...     ...     ...           ...     ...
1995   996.0   996.60  999.52   19602  1.959253e+12     NaN
1996   983.0   983.95  997.46   10256  1.022997e+12     NaN
1997  1003.0  1002.70  999.56    4747  4.744921e+11     NaN
1998   996.0   995.40  998.58    3690  3.684748e+11     NaN
1999   988.0   989.45  990.63    3279  3.248279e+11     NaN

      Deliverable Volume  %Deliverble
0                    NaN          NaN
1                    NaN          NaN
2                    NaN          NaN
3                    NaN          NaN
4                    NaN          NaN
...                  ...          ...
1995             18926.0       0.9655
1996              7879.0       0.7682
1997              4158.0       0.8759
1998              2601.0       0.7049
1999              2253.0       0.6871

[2000 rows x 15 columns]
```

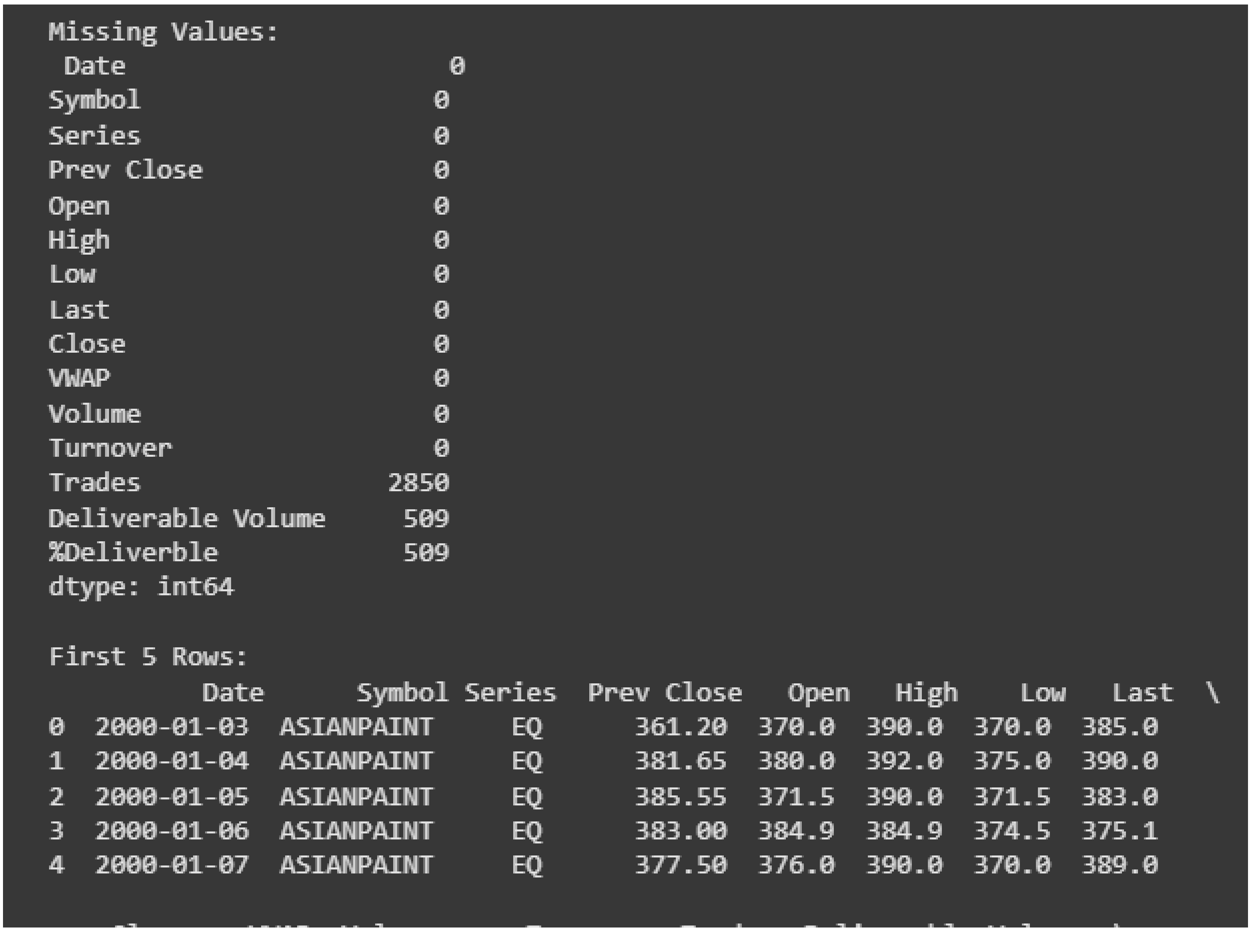
5. Statistical Analysis:

5.1 Descriptive Statistics:

```
import pandas as pd

df = pd.read_csv("ASIANPAINT.csv")

print("Shape:", df.shape)

print("\nColumn Data Types:\n", df.dtypes)

print("\nDescriptive Statistics:\n", df.describe())

print("\nMissing Values:\n", df.isnull().sum())

print("\nFirst 5 Rows:\n", df.head())
```



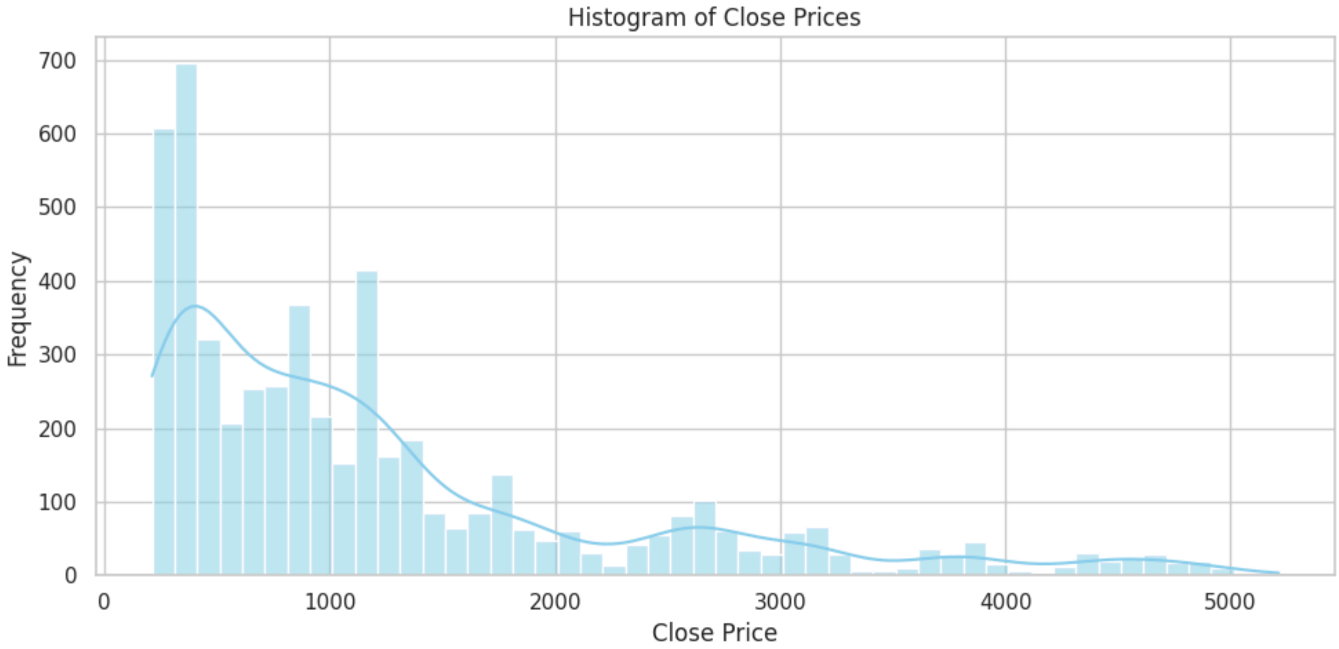
```
Descriptive Statistics:
         Prev Close         Open         High          Low         Last  \
count  5306.000000  5306.000000  5306.000000  5306.000000  5306.000000   
mean   1247.000952  1247.683952  1264.625349  1230.900697  1247.317132   
std    1074.399506  1074.025577  1087.238871  1062.629409  1074.432667   
min     210.750000   210.000000   215.750000   204.000000   210.600000   
25%     415.737500   415.000000   424.950000   410.000000   416.000000   
50%     889.375000   890.000000   903.975000   878.075000   890.000000   
75%    1599.037500  1599.800000  1629.675000  1573.037500  1602.500000   
max    5213.100000  5221.100000  5247.750000  5150.050000  5221.100000   
```

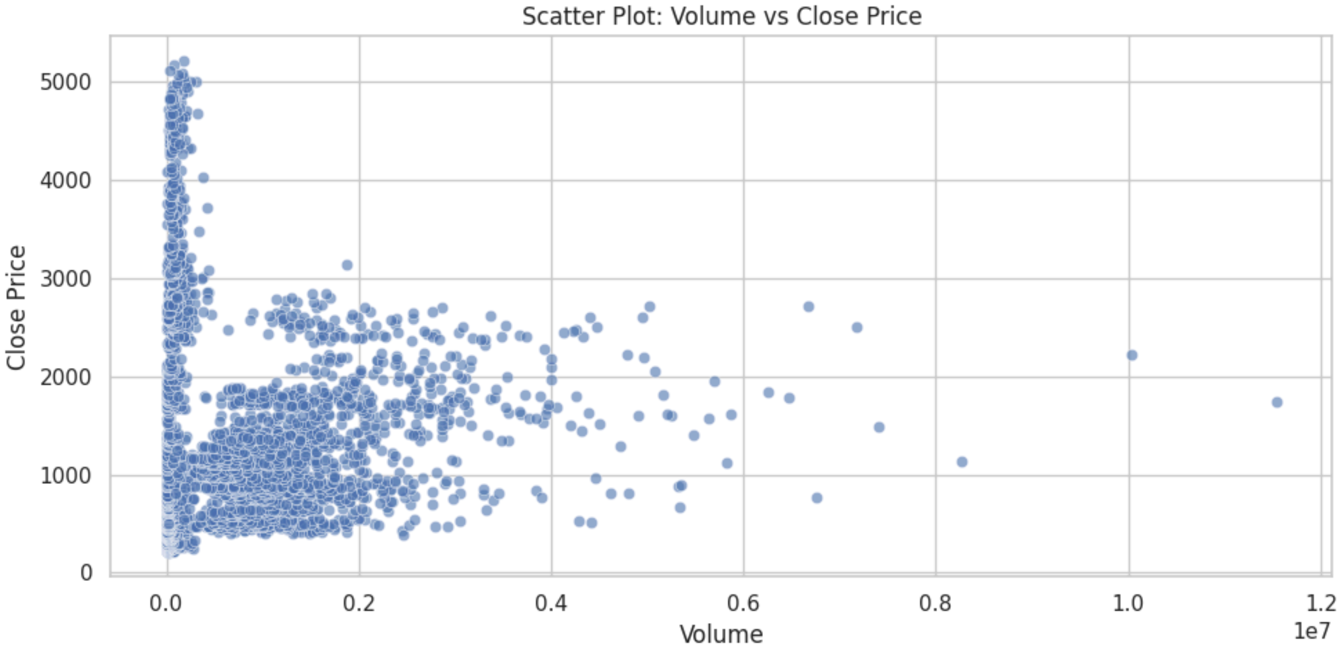


```
Missing Values:
 Date                     0
Symbol                   0
Series                   0
Prev Close               0
Open                     0
High                     0
Low                      0
Last                     0
Close                    0
VWAP                     0
Volume                   0
Turnover                 0
Trades                2850
Deliverable Volume     509
%Deliverble            509
dtype: int64

First 5 Rows:
         Date      Symbol Series  Prev Close   Open   High    Low   Last  \
0  2000-01-03  ASIANPAINT     EQ      361.20  370.0  390.0  370.0  385.0   
1  2000-01-04  ASIANPAINT     EQ      381.65  380.0  392.0  375.0  390.0   
2  2000-01-05  ASIANPAINT     EQ      385.55  371.5  390.0  371.5  383.0   
3  2000-01-06  ASIANPAINT     EQ      383.00  384.9  384.9  374.5  375.1   
4  2000-01-07  ASIANPAINT     EQ      377.50  376.0  390.0  370.0  389.0   
```

5.2 Inferential Statistics

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read_csv("ASIANPAINT.csv")

sns.set(style="whitegrid")

plt.figure(figsize=(10, 5))

sns.histplot(df['Close'], bins=50, kde=True, color='skyblue')

plt.title('Histogram of Close Prices')

plt.xlabel('Close Price')

plt.ylabel('Frequency')

plt.tight_layout()

plt.show()

plt.figure(figsize=(10, 5))

sns.scatterplot(data=df, x='Volume', y='Close', alpha=0.6)

plt.title('Scatter Plot: Volume vs Close Price')

plt.xlabel('Volume')

plt.ylabel('Close Price')

plt.tight_layout()

plt.show()
```

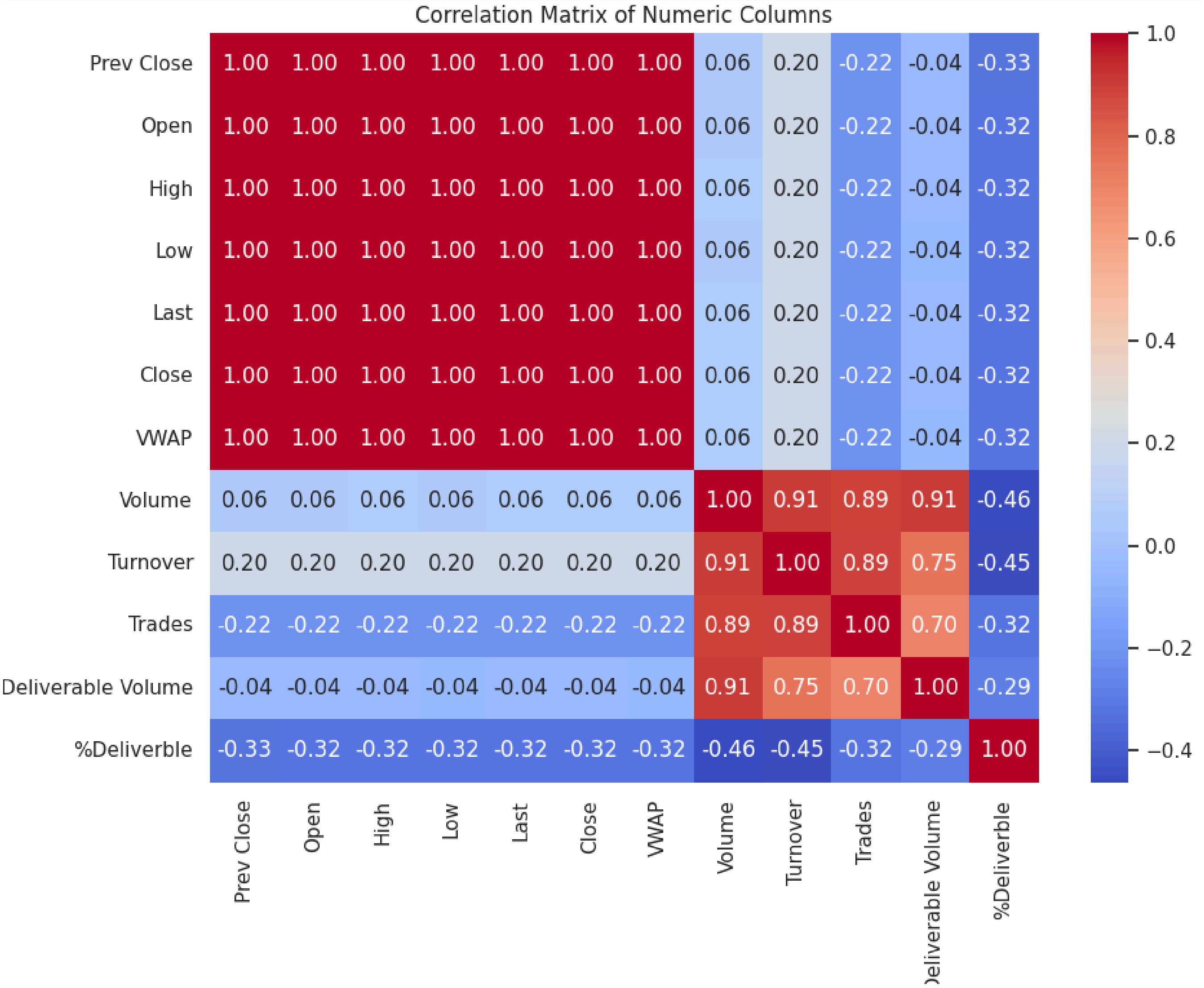


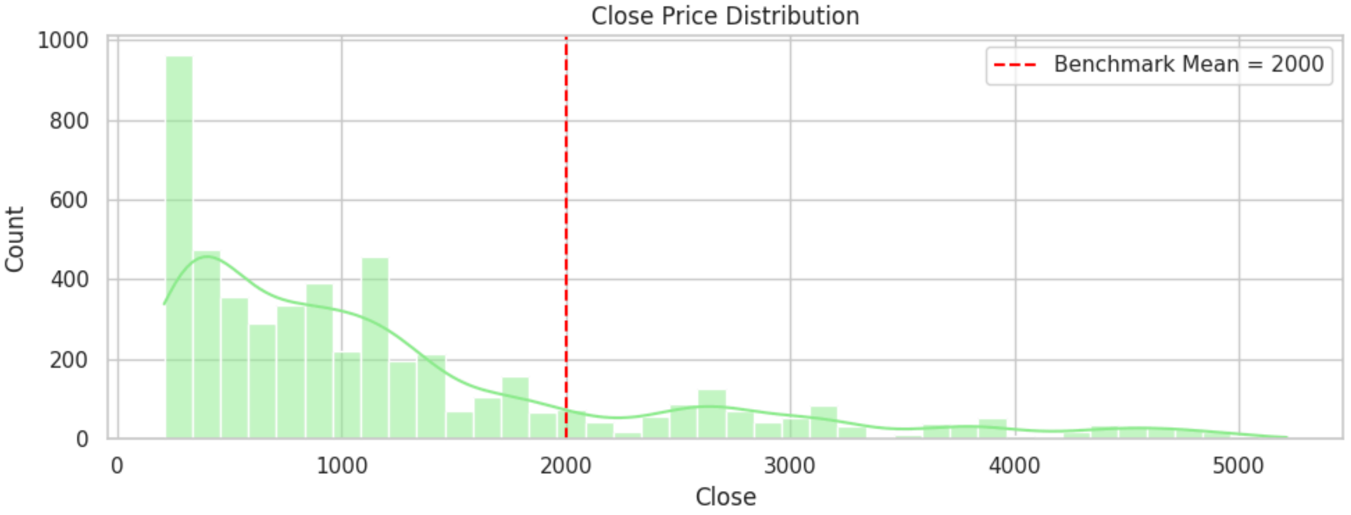


5.2 Inferential Statistics

```
import pandas as pd

import numpy as np

from scipy import stats
```

```
import seaborn as sns

import matplotlib.pyplot as plt

df = pd.read_csv("ASIANPAINT.csv")

numeric_df = df.select_dtypes(include=['float64', 'int64'])

conf_intervals = {}

for col in numeric_df.columns:

    data = numeric_df[col].dropna()

    mean = np.mean(data)

    sem = stats.sem(data)  # standard error of the mean

    ci = stats.t.interval(0.95, len(data)-1, loc=mean, scale=sem)

    conf_intervals[col] = {'Mean': mean, '95% CI': ci}

t_stat, p_val = stats.ttest_1samp(df['Close'].dropna(), 2000)

plt.figure(figsize=(10, 8))

corr = numeric_df.corr()

sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")

plt.title("Correlation Matrix of Numeric Columns")

plt.tight_layout()

plt.show()
plt.figure(figsize=(10, 4))

sns.histplot(df['Close'], bins=40, kde=True, color='lightgreen')

plt.axvline(x=2000, color='red', linestyle='--', label='Benchmark Mean = 2000')

plt.title('Close Price Distribution')

plt.legend()

plt.tight_layout()

plt.show()

conf_intervals, {'T-statistic': t_stat, 'P-value': p_val}
```


Correlation Matrix of Numeric Columns
Prev Close 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.06 0.20 -0.22 -0.04 -0.33
Open 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.06 0.20 -0.22 -0.04 -0.32
High 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.06 0.20 -0.22 -0.04 -0.32
Low 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.06 0.20 -0.22 -0.04 -0.32
Last 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.06 0.20 -0.22 -0.04 -0.32
Close 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.06 0.20 -0.22 -0.04 -0.32
VWAP 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.06 0.20 -0.22 -0.04 -0.32
Volume 0.06 0.06 0.06 0.06 0.06 0.06 0.06 1.00 0.91 0.89 0.91 -0.46
Turnover 0.20 0.20 0.20 0.20 0.20 0.20 0.20 0.91 1.00 0.89 0.75 -0.45
Trades -0.22 -0.22 -0.22 -0.22 -0.22 -0.22 -0.22 0.89 0.89 1.00 0.70 -0.32
Deliverable Volume -0.04 -0.04 -0.04 -0.04 -0.04 -0.04 -0.04 0.91 0.75 0.70 1.00 -0.29
%Deliverble -0.33 -0.32 -0.32 -0.32 -0.32 -0.32 -0.32 -0.46 -0.45 -0.32 -0.29 1.00
Prev Close
Open
High
Low
Last
Close
VWAP
Volume
Turnover
Trades
)eliverable Volume
%Deliverble
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4


Close Price Distribution
Benchmark Mean = 2000
Count
1000
800
600
400
200
0
0
1000
2000
3000
4000
5000
Close

6. Machine Learning Algorithm

6.1 Model Chosen: Logistic Regression

Reason:

Logistic Regression is a fundamental machine learning algorithm used for binary classification problems. In the context of stock analysis, it can be used to predict categorical outcomes such as:

- Whether the stock price will increase or decrease compared to the previous day.
- Whether a stock is being actively traded (based on volume or delivery percentages).
- Market behavior classifications (e.g., high-risk vs low-risk days).

Code:

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import (

    accuracy_score, classification_report, confusion_matrix, roc_curve, auc

)


# Load and preprocess the dataset

df = pd.read_csv("ASIANPAINT.csv")

df = df.dropna(subset=['Open', 'Close', 'Prev Close', 'High', 'Low', 'Last', 'VWAP', 'Volume'])
```

```
df['Target'] = (df['Close'] > df['Open']).astype(int)

features = ['Prev Close', 'High', 'Low', 'Last', 'VWAP', 'Volume']
X = df[features]
y = df['Target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:, 1]  # Probabilities for ROC curve

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))


cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=["Down", "Up"], yticklabels=["Down", "Up"])
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
```

plt.ylabel("Actual")

plt.show()

fpr, tpr, thresholds = roc_curve(y_test, y_prob)

roc_auc = auc(fpr, tpr)

plt.figure(figsize=(6, 5))

plt.plot(fpr, tpr, color='darkorange', label=f"ROC curve (AUC = {roc_auc:.2f})")

plt.plot([0, 1], [0, 1], color='navy', linestyle='--')

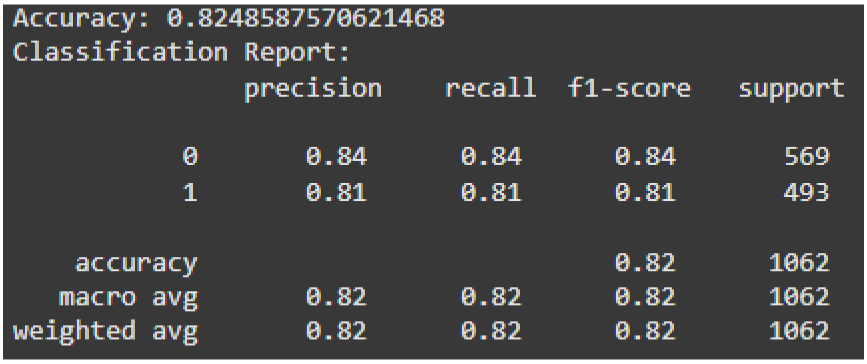
plt.xlabel("False Positive Rate")

plt.ylabel("True Positive Rate")

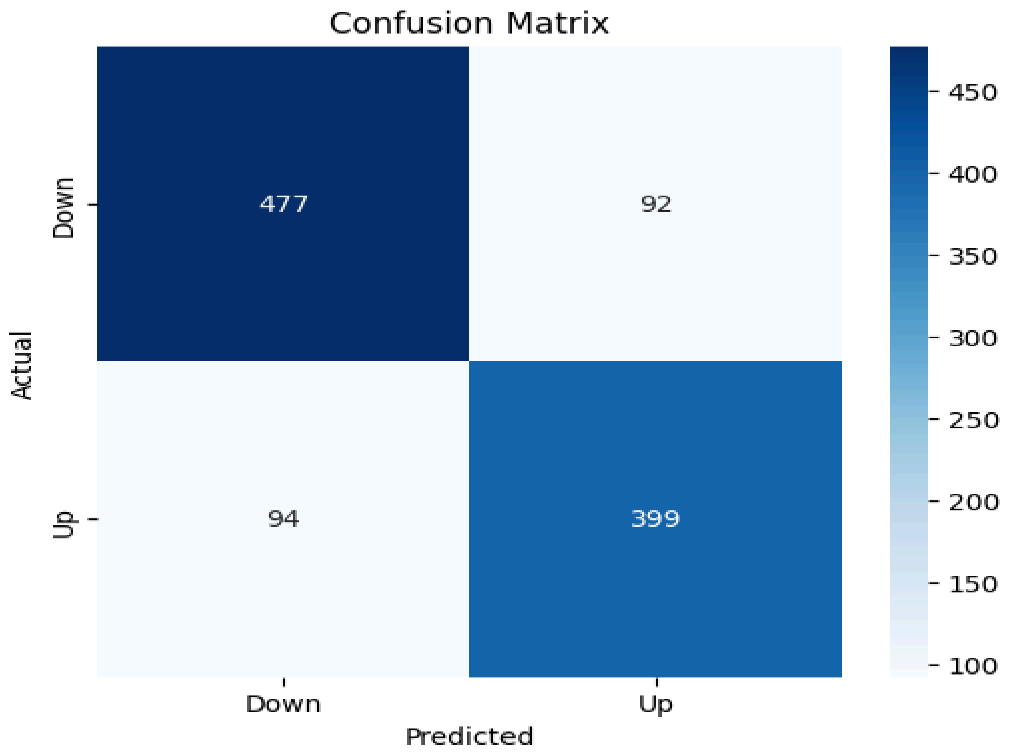
plt.title("Receiver Operating Characteristic (ROC) Curve")

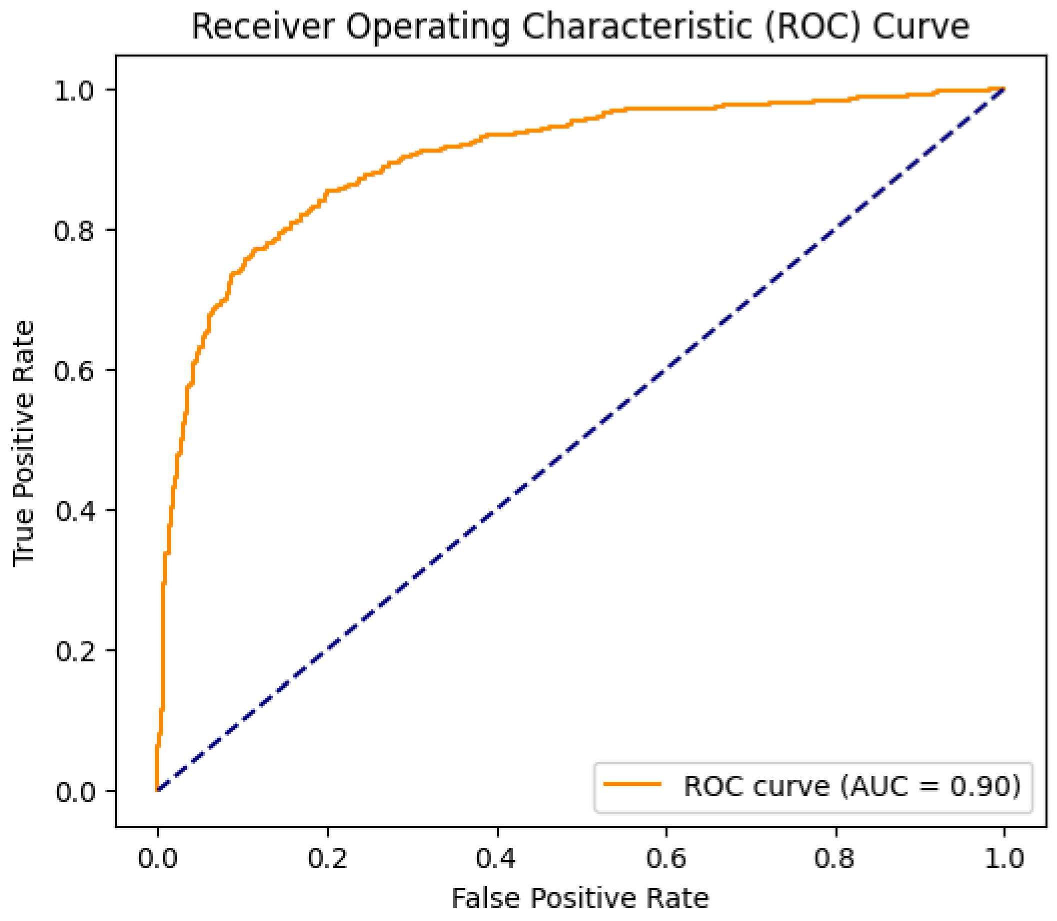
plt.legend(loc="lower right")

plt.show()



```
Accuracy: 0.8248587570621468
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.84      0.84       569
           1       0.81      0.81      0.81       493

    accuracy                           0.82      1062
   macro avg       0.82      0.82      0.82      1062
weighted avg       0.82      0.82      0.82      1062
```





7. Conclusion:

This project successfully applied descriptive and inferential statistics along with machine learning models to analyze the historical stock data of ASIAN PAINT. Through statistical analysis, key insights were uncovered regarding price trends, volatility, and investor behavior. The inferential methods, such as confidence intervals and hypothesis testing, validated significant differences in market movement.

This analysis demonstrates how data science can effectively support market analysis, aid in decision-making, and identify patterns that may not be visible through manual observation alone.

8. Future Scope

- Use of LSTM and Deep Learning Models-Implement advanced time series models (like LSTM or GRU) to better capture sequential dependencies in stock price behavior.
- Real-Time Stock Prediction-Develop a live dashboard using APIs to pull real-time stock data and apply trained models to make predictions instantly.
- Sentiment Analysis Integration-Combine stock data with news or social media sentiment to improve prediction accuracy.
- Alert System for Traders-Build a rule-based or AI-powered alert system to notify traders of high-risk or high-opportunity trading days based on model output.