P-439

CRUDE OIL PRICE PREDICTION

GROUP MEMBERS:

- GULAM MOHIUDDIN KHAN
- MOHDGOUS SHAKEEL KURNE
- NAMITHA KUAMRI
- SHREEYA NITTURKAR
- SOORA SHIRISHA
- SURENDRA KUMAR ABALUR

OBJECTIVE

To forecast oil prices, which are often influenced by external factors rather than real-time data, making predictions challenging. Our model aims to identify patterns in oil price movements to assist customers and businesses in making informed decisions, ultimately mitigating the economic impact of fluctuating oil prices

DATA SET DESCRIPTION

The dataset named Crude oil Prices Daily.xlsx contains data collected to analyze and predict crude oil prices. The dataset includes various market factors and external influences that affect the fluctuations in oil prices, providing insights into the complex dynamics of global oil markets.

- **Number of Instances (Rows): 8223**
- **Number of Variables (Columns): 2**

	Date	Closing Value
0	1986-01-02	25.56
1	1986-01-03	26.00
2	1986-01-06	26.53
3	1986-01-07	25.85
4	1986-01-08	25.87
...	...	...
8218	2018-07-03	74.19
8219	2018-07-04	NaN
8220	2018-07-05	73.05
8221	2018-07-06	73.78

- **Data Source:** The dataset, named Crude Oil Prices Daily.xlsx, contains daily records of crude oil prices.
- **Time Period:** The dataset spans a significant period, capturing fluctuations in oil prices over time, helping to understand both short-term and long-term trends.
- **Key Variables:** It includes variables like date, daily closing price, opening price, highs, and lows.
- **Market Factors:** The dataset reflects the impact of various global events and economic indicators on crude oil prices.
- **Usage**: This data is used to analyze price trends, identify key drivers of price changes, and forecast future price movements.
- **Environmental and Economic Impact:** The data provides insights into the economic and environmental factors influencing crude oil prices.
- **products.** While they've seen success, we acknowledge the evolving market dynamics and the need to adapt our sales strategies to maintain and increase our competitive edge.

Exploratory Data Analysis

SUMMARY STATISTICS

```
[ ]  df.describe()
```

	Date	Closing Value
count	8223	8216.000000
mean	2002-04-05 22:11:15.082086912	43.492139
min	1986-01-02 00:00:00	10.250000
25%	1994-01-25 12:00:00	19.577500
50%	2002-04-02 00:00:00	29.610000
75%	2010-06-12 12:00:00	63.402500
max	2018-07-09 00:00:00	145.310000
std	NaN	29.616804

Missing Values:

DATE	0
CLOSING VALUE	7

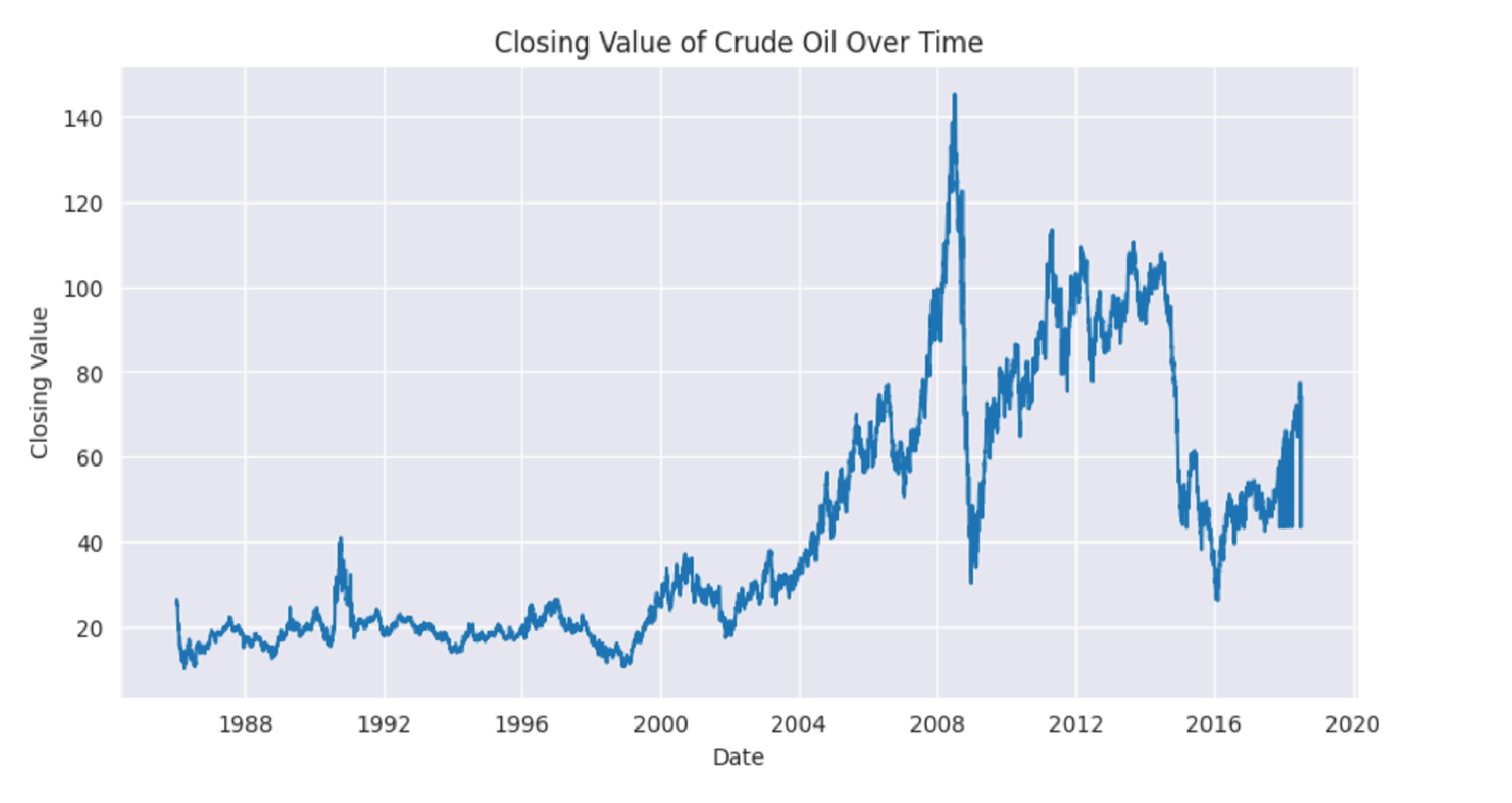
Replaced by Mean :

Number of Outliers: 34

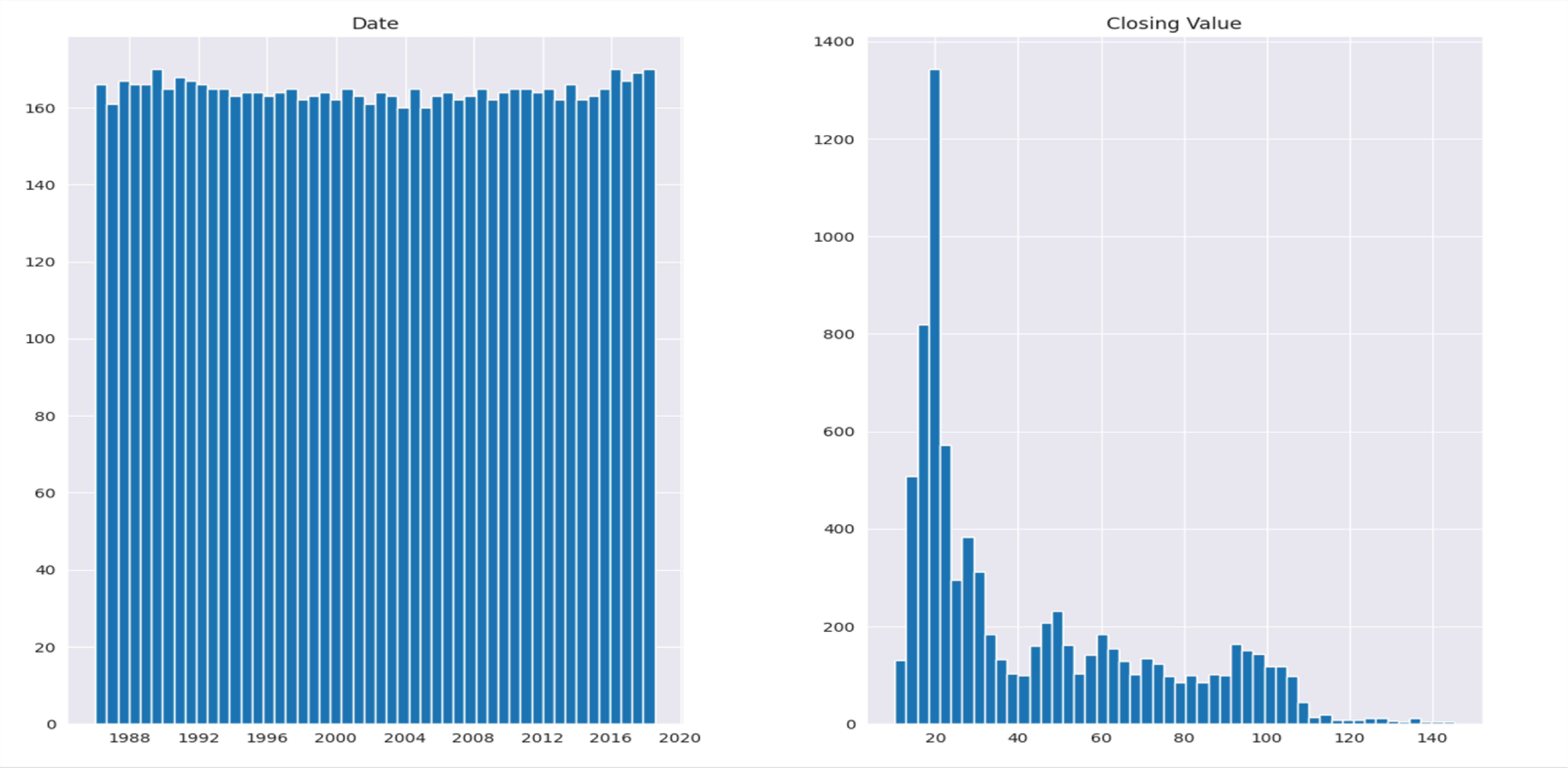
```
df['Closing Value'].fillna(df['Closing Value'].mean(), inplace = True)
```

VISUALIZATIONS

Line plot

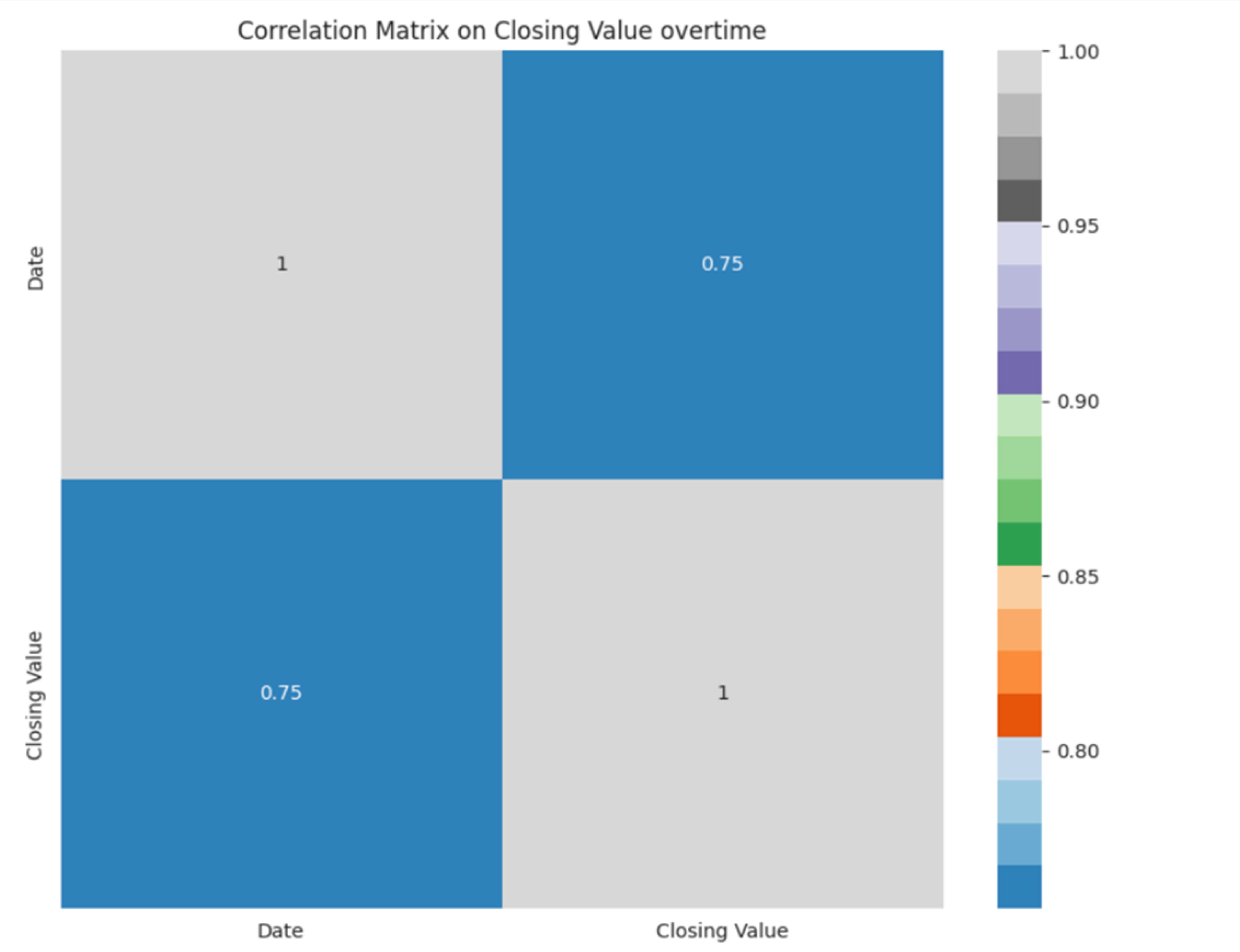


Histogram

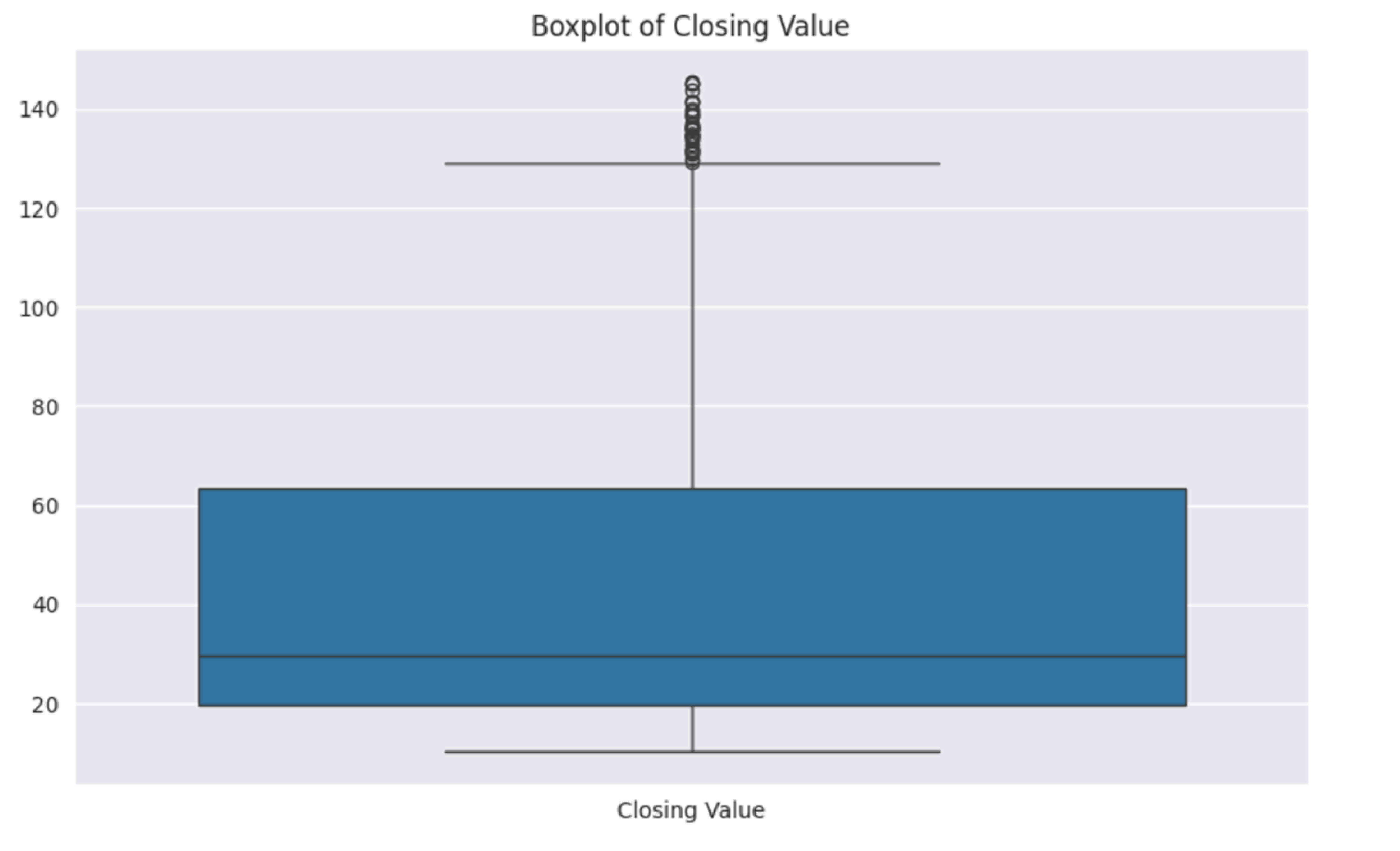


Date
0
20
40
60
80
100
120
140
160
1988
1992
1996
2000
2004
2008
2012
2016
2020
Closing Value
0
200
400
600
800
1000
1200
1400
20
40
60
80
100
120
140

Correlation HeatMap



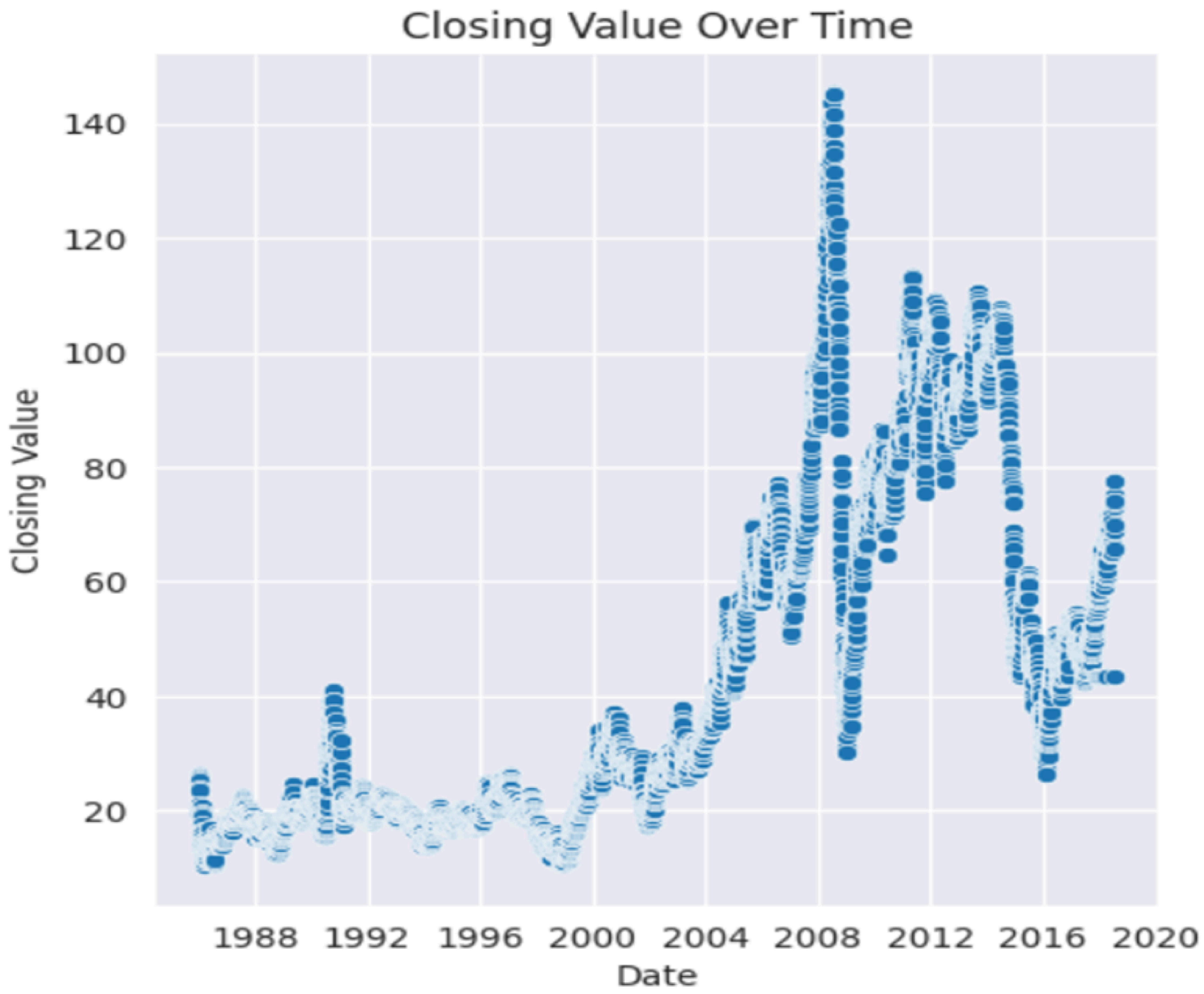
Box Plot



Boxplot of Closing Value
140
120
100
80
60
40
20
Closing Value

Relational Plot

<Figure size 1000x600 with 0 Axes>



Model Building

We Implemented

ARIMA Model:

- Utilized for time-series analysis.
- Captures trends and seasonality effectively.

FBProphet Model:

- Handles missing data and outliers well.
- Suitable for business forecasting.

Linear Regression Model:

- Models relationships between variables.
- Simple and effective for predictive tasks.

LSTM (Long Short-Term Memory):

- Deep learning model for sequential data.
- Captures long-term dependencies for accurate forecasting.

Arima

```
last_date = df['Date'].iloc[-1]
forecast_dates = pd.date_range(start=last_date + pd.DateOffset(days=1), periods=30)

forecast_df = pd.DataFrame({'Date': forecast_dates, 'Closing Value': forecast_arima})
print(forecast_df)
```

```
           Date  Closing Value
8223 2018-07-10      73.500150
8224 2018-07-11      73.777619
8225 2018-07-12      73.649127
8226 2018-07-13      73.662335
8227 2018-07-14      73.662736
8228 2018-07-15      73.665288
8229 2018-07-16      73.663272
8230 2018-07-17      73.664012
8231 2018-07-18      73.663908
8232 2018-07-19      73.663911
8233 2018-07-20      73.663895
8234 2018-07-21      73.663909
8235 2018-07-22      73.663904
8236 2018-07-23      73.663905
8237 2018-07-24      73.663905
8238 2018-07-25      73.663905
8239 2018-07-26      73.663905
8240 2018-07-27      73.663905
8241 2018-07-28      73.663905
8242 2018-07-29      73.663905
8243 2018-07-30      73.663905
8244 2018-07-31      73.663905
8245 2018-08-01      73.663905
8246 2018-08-02      73.663905
8247 2018-08-03      73.663905
8248 2018-08-04      73.663905
8249 2018-08-05      73.663905
8250 2018-08-06      73.663905
8251 2018-08-07      73.663905
8252 2018-08-08      73.663905
```

We implemented an ARIMA model to forecast the next 30 days of closing values. The model was fitted on historical data, and forecasted values were stored in a new DataFrame. And Plotted original closing values and ARIMA forecast for comparison.

Data Collection:We used the Prophet model for forecasting by first preparing the data with the required column names. After fitting the model, we created a future dataframe for the next 30 days and generated predictions. Finally, we plotted the forecast results to visualize the expected closing values.

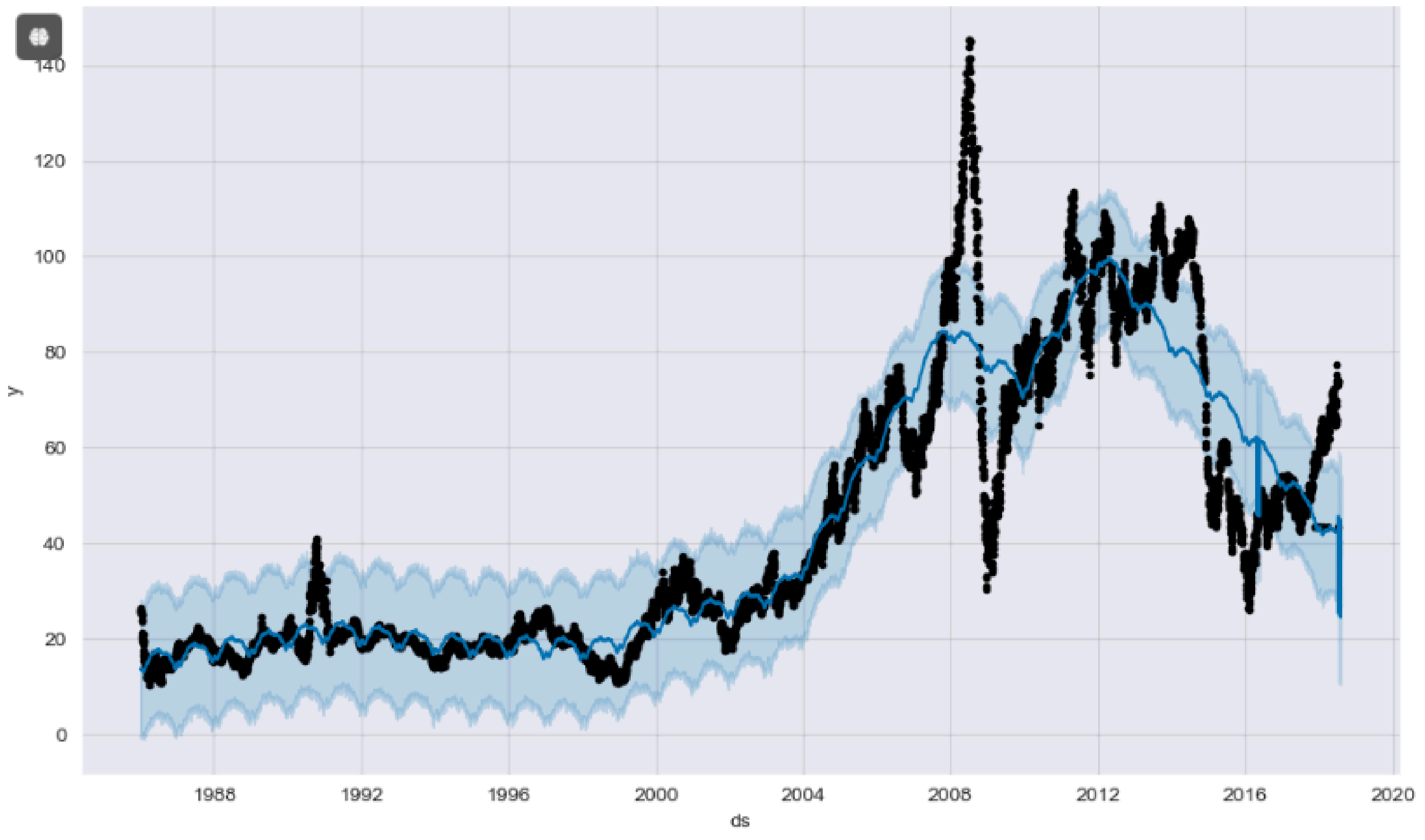
```
[31]: forecast_prophet = model_prophet.predict(future_dataframe)
      forecast_prophet
```

[31]:

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	weekly	weekly_lower	week
0	1986-01-02	13.547237	-0.812076	26.896852	13.547237	13.547237	0.044834	0.044834	0.044834	1.737094	1.737094	
1	1986-01-03	13.550835	-0.105520	28.257352	13.550835	13.550835	0.075047	0.075047	0.075047	1.715310	1.715310	
2	1986-01-06	13.561631	0.286505	26.861136	13.561631	13.561631	0.151355	0.151355	0.151355	1.676326	1.676326	
3	1986-01-07	13.565229	0.440669	27.035142	13.565229	13.565229	0.096666	0.096666	0.096666	1.597896	1.597896	
4	1986-01-08	13.568828	-0.309276	27.093065	13.568828	13.568828	0.121138	0.121138	0.121138	1.606217	1.606217	
...	...	...	...	...	...	...	...	...	...	...	...	
8248	2018-08-04	37.771996	30.964056	57.642273	37.771996	37.771996	7.021247	7.021247	7.021247	5.944766	5.944766	
8249	2018-08-05	37.746341	10.509279	37.130297	37.746341	37.746341	-13.206400	-13.206400	-13.206400	-14.277608	-14.277608	-1
8250	2018-08-06	37.720686	27.279881	54.412939	37.720686	37.720686	2.740785	2.740785	2.740785	1.676326	1.676326	
8251	2018-08-07	37.695032	26.908909	53.507895	37.695032	37.695032	2.654032	2.654032	2.654032	1.597896	1.597896	
8252	2018-08-08	37.669377	26.953830	53.602828	37.669377	37.669377	2.652430	2.652430	2.652430	1.606217	1.606217	

```
model_prophet.plot(forecast_prophet)
plt.show()
```

```
C:\Users\nittu\anaconda3\Lib\site-packages\prophet\plot.py:72: FutureWarning: The behavior of DatetimeProperties.to_pydatetime is deprecated, :
e version this will return a Series containing python datetime objects instead of an ndarray. To retain the old behavior, call `np.array` on tl
  fcst_t = fcst['ds'].dt.to_pydatetime()
C:\Users\nittu\anaconda3\Lib\site-packages\prophet\plot.py:73: FutureWarning: The behavior of DatetimeProperties.to_pydatetime is deprecated, :
e version this will return a Series containing python datetime objects instead of an ndarray. To retain the old behavior, call `np.array` on tl
  ax.plot(m.history['ds'].dt.to_pydatetime(), m.history['y'], 'k.',
```

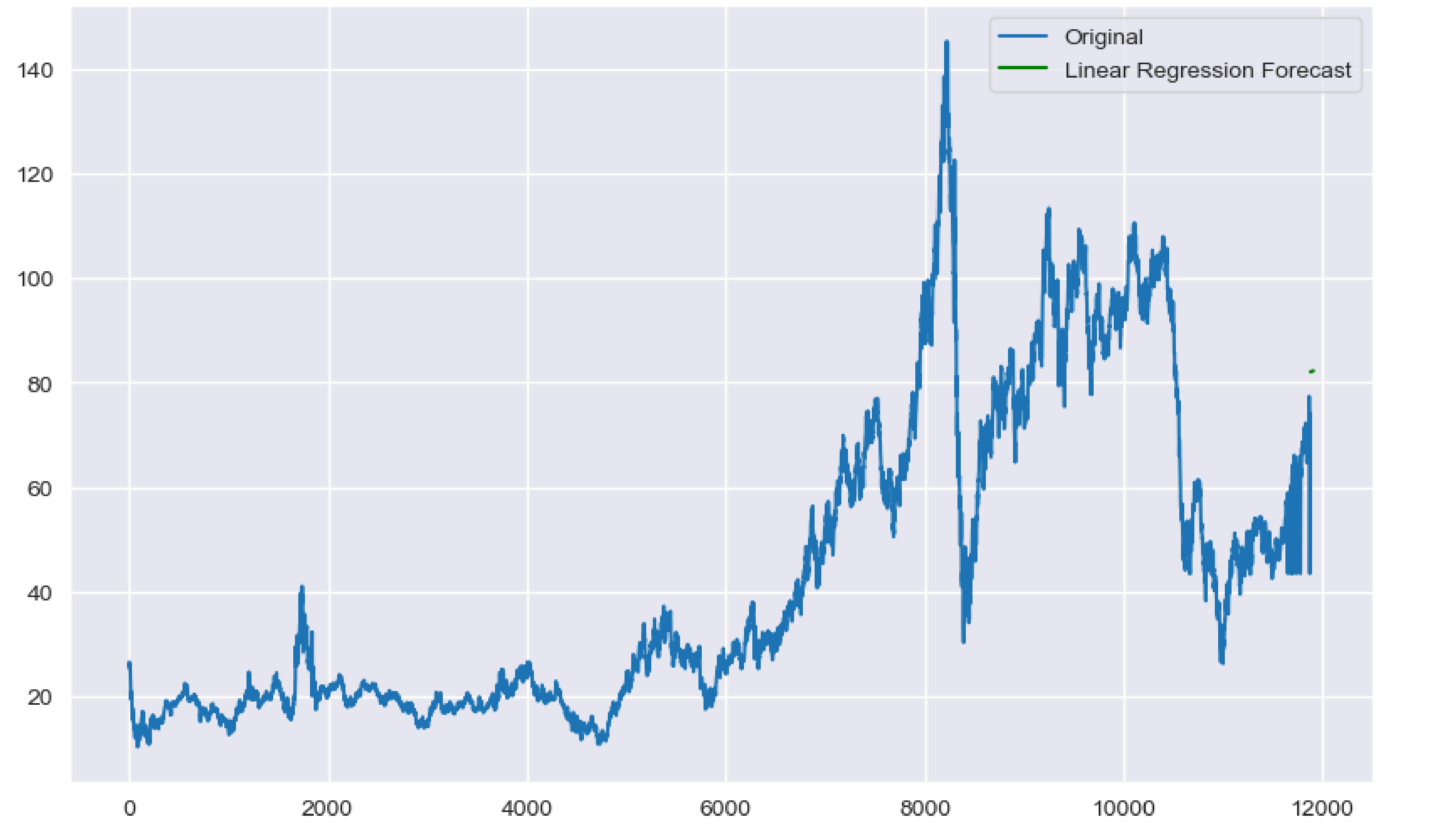


prophet

Linear Regression

We implemented a Linear Regression model by converting the date into a numerical sequence representing days. After fitting the model using historical closing values, we predicted values for the next 30 days and visualized the results by plotting both the original data and the forecast.

```
plt.figure(figsize=(10, 6))
plt.plot(df['Days'], df['Closing Value'], label='Original')
plt.plot(future_days, forecast_lr, label='Linear Regression Forecast', color='green')
plt.legend()
plt.show()
```




```
LinearRegression
LinearRegression()
```

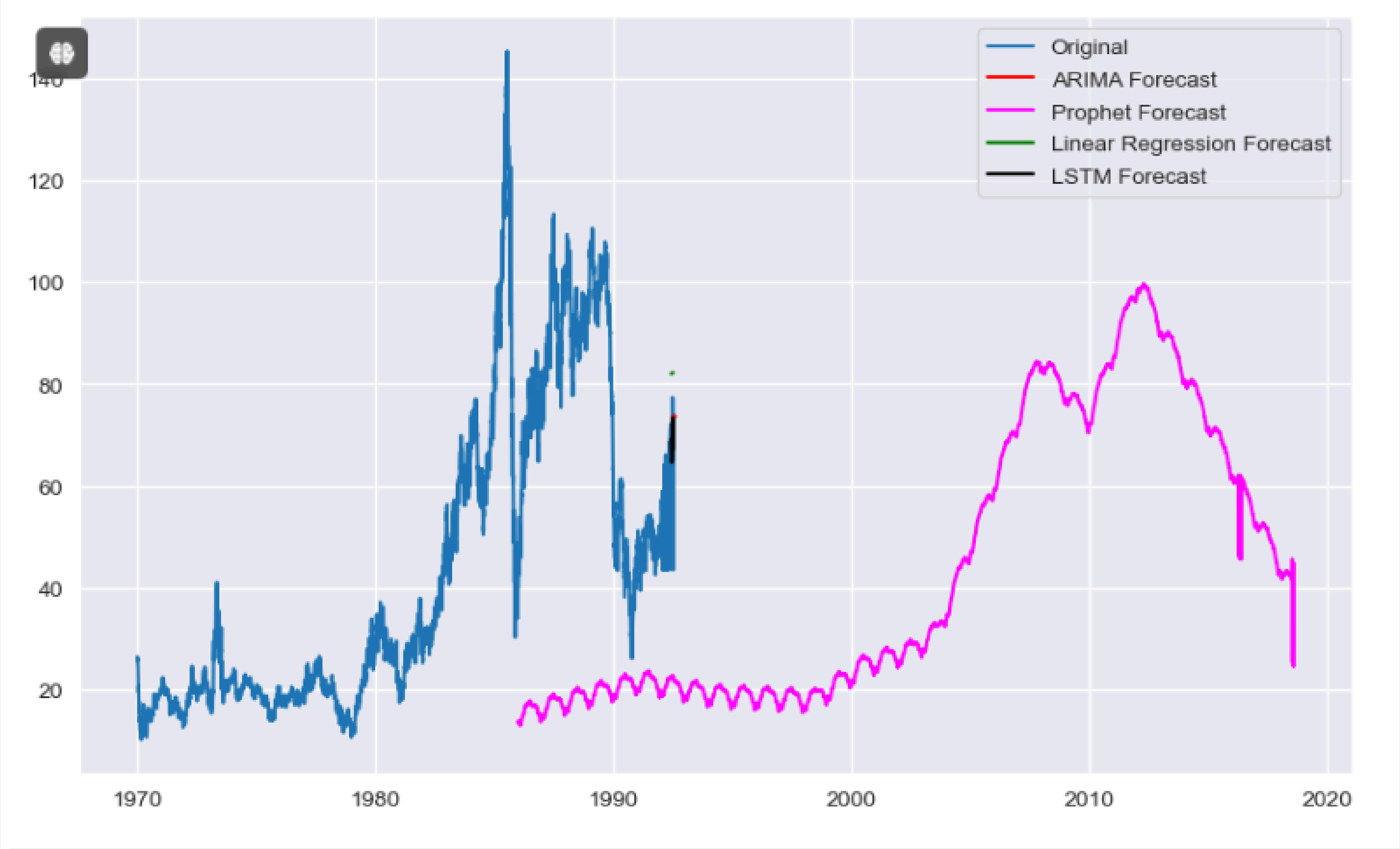
```
forecast_lr = model_lr.predict(future_days)
forecast_lr
```

```
C:\Users\nittu\anaconda3\Lib\site-packages\sklearn\base.py:493: User
ature names
  warnings.warn(
```

```
array([82.0481361 , 82.05462802, 82.06111994, 82.06761186, 82.074103
       82.0805957 , 82.08708761, 82.09357953, 82.10007145, 82.106563
       82.11305529, 82.11954721, 82.12603913, 82.13253105, 82.139022
       82.14551489, 82.15200681, 82.15849872, 82.16499064, 82.171482
       82.17797448, 82.1844664 , 82.19095832, 82.19745024, 82.203942
       82.21043408, 82.216926  , 82.22341792, 82.22990983, 82.236401
```

LSTM

We implemented an LSTM model for forecasting by first scaling the closing values using MinMaxScaler. We prepared the data by creating sequences of 60 time steps. The LSTM model was then built, trained, and used to predict future values. Finally, we visualized the forecasts from ARIMA, Prophet, Linear Regression, and LSTM models to compare their performances.



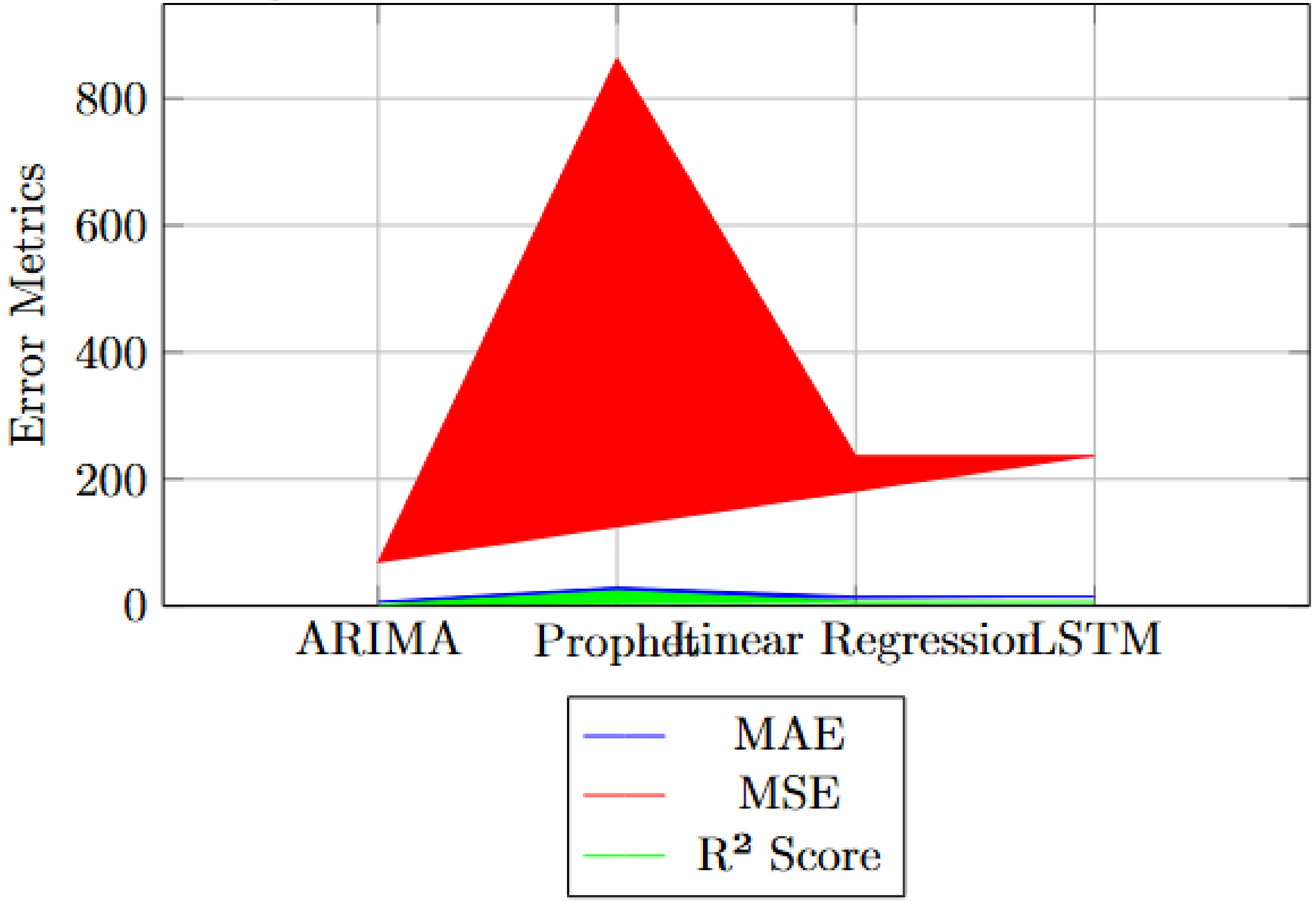

```
Epoch 1/10
256/256 ━━━━━━━━━━━━━━━━━━━━ 25s 62ms/step - loss: 0.0110
Epoch 2/10
256/256 ━━━━━━━━━━━━━━━━━━━━ 16s 62ms/step - loss: 0.0011
Epoch 3/10
256/256 ━━━━━━━━━━━━━━━━━━━━ 17s 65ms/step - loss: 0.0011
Epoch 4/10
256/256 ━━━━━━━━━━━━━━━━━━━━ 17s 67ms/step - loss: 0.0012
Epoch 5/10
256/256 ━━━━━━━━━━━━━━━━━━━━ 16s 62ms/step - loss: 8.7179e-04
Epoch 6/10
256/256 ━━━━━━━━━━━━━━━━━━━━ 16s 62ms/step - loss: 8.1532e-04
Epoch 7/10
256/256 ━━━━━━━━━━━━━━━━━━━━ 16s 64ms/step - loss: 6.9748e-04
Epoch 8/10
256/256 ━━━━━━━━━━━━━━━━━━━━ 16s 63ms/step - loss: 7.1644e-04
Epoch 9/10
256/256 ━━━━━━━━━━━━━━━━━━━━ 16s 62ms/step - loss: 5.8990e-04
Epoch 10/10
256/256 ━━━━━━━━━━━━━━━━━━━━ 15s 60ms/step - loss: 5.3796e-04
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 778ms/step
```

Comparision between Models

This visualization presents a compact comparison of four forecasting models: ARIMA, Prophet, Linear Regression, and LSTM. Each model is evaluated using three key metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score. The grouped bar chart allows for quick visual assessment of each model's performance, highlighting their strengths and weaknesses.

Model	MAE	MSE	R^2 Score
ARIMA	6.13	67.67	-0.91
Prophet	28.64	862.75	-23.30
Linear Regression	14.15	235.64	-5.64
LSTM	14.15	235.64	-5.64

Figure 1: Model Performance Comparison



Deployment Using Streamlit App

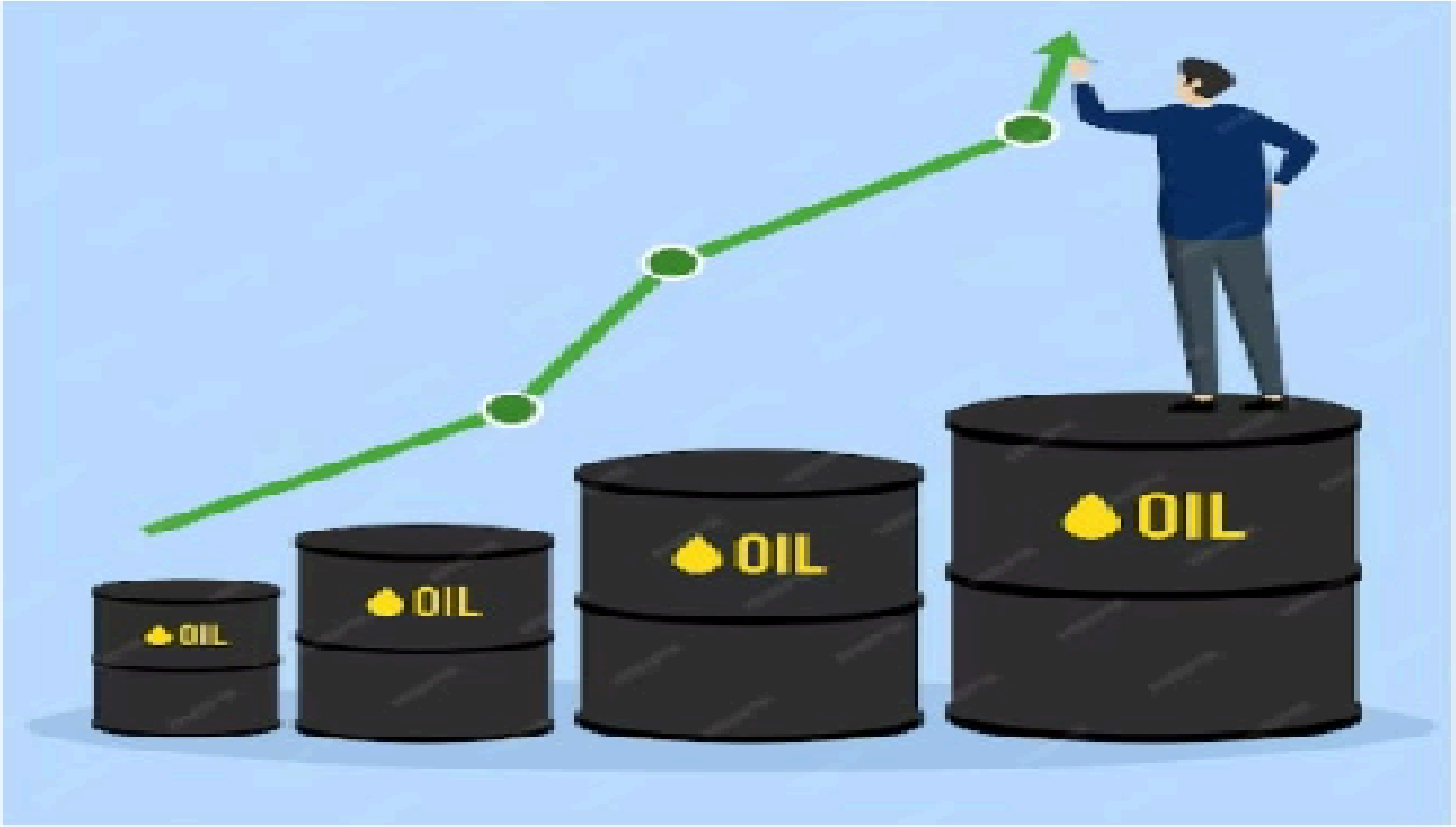
ARIMA Model Selected For Depoloyment

Amomg all the models we have the lowest MAE and MSE with ARIMA Model and also the highest R2 score.
So here is the model_comparision with all models.

	Model	MAE	MSE	R2 Score
0	ARIMA	6.134121	67.673113	-0.906154
1	Prophet	28.639741	862.754576	-23.301275
2	Linear Regression	14.153864	235.638124	-5.637237
3	LSTM	14.153864	235.638124	-5.637237

After Deployment in Streamlit App

Crude Oil Price Forecasting App for ARIMA Model



Crude Oil

Select the start date for historical data.

1986/01/02

Select the end date for historical data:

2018/07/09

Enter the number of days to forecast:

30

Forecast

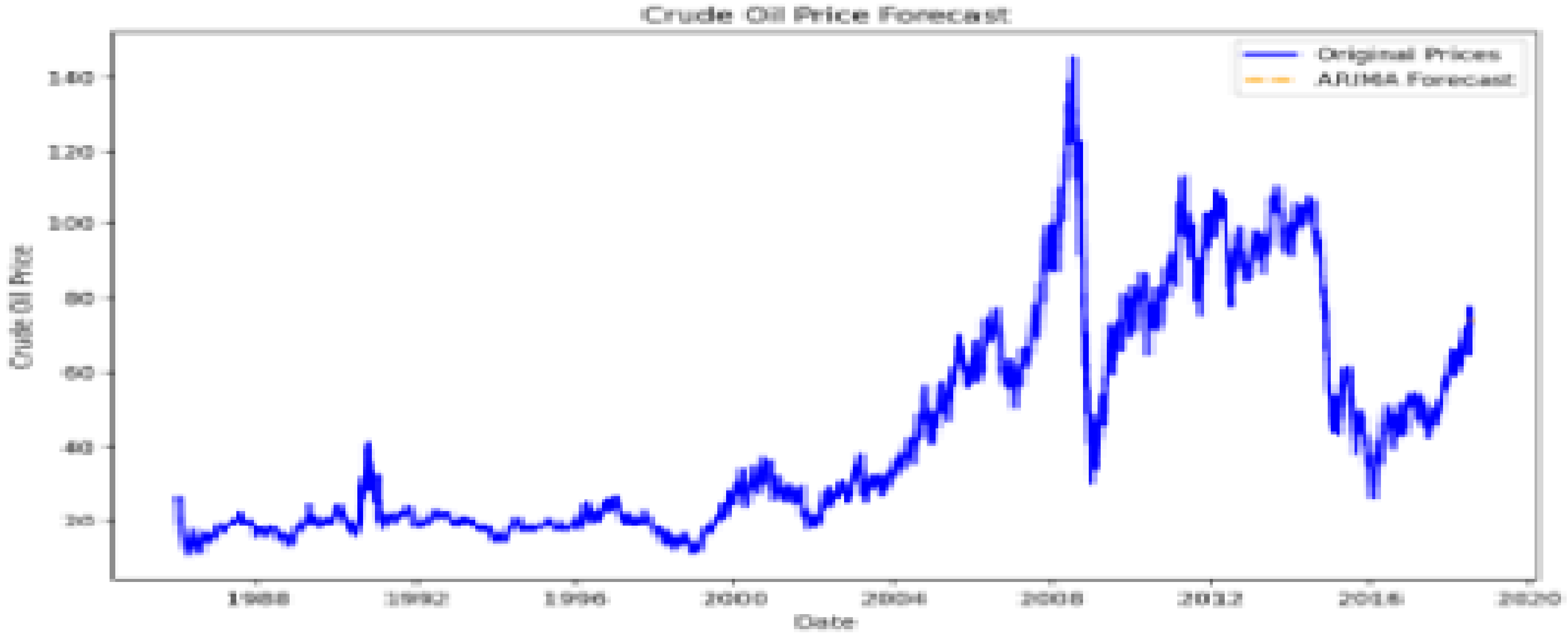
Enter the number of days to forecast:

30

Forecast

Forecasted Crude Oil Prices:

	Date	Forecasted Price
8,223	2018-07-10 00:00:00	73.8714
8,224	2018-07-11 00:00:00	73.9061
8,225	2018-07-12 00:00:00	73.9408
8,226	2018-07-13 00:00:00	73.9077
8,227	2018-07-14 00:00:00	73.9009
8,228	2018-07-15 00:00:00	73.9058
8,229	2018-07-16 00:00:00	73.9044
8,230	2018-07-17 00:00:00	73.902
8,231	2018-07-18 00:00:00	73.9036
8,232	2018-07-19 00:00:00	73.904



Explanation

In the Streamlit deployment, the ARIMA model is used to forecast crude oil prices for a user-specified number of days.

The user inputs the number of days they want to forecast, and upon clicking the "Forecast" button, the model makes predictions using the predict_crude_oil_price function.

The results, both the numerical values and a visual plot of the forecast, are then displayed in the Streamlit app.

The model's deployment in Streamlit allows for easy access and use of the forecasting functionality.

Users can simply input the number of days they are interested in and obtain the predicted prices without needing to interact with any complex code.

The visual plot further enhances the ease of understanding and interpretation of the forecasts.

Thank You