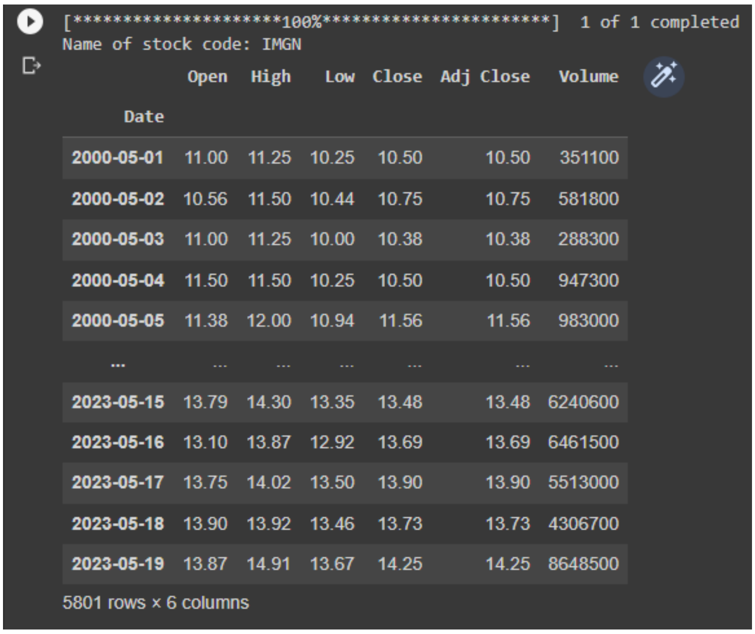
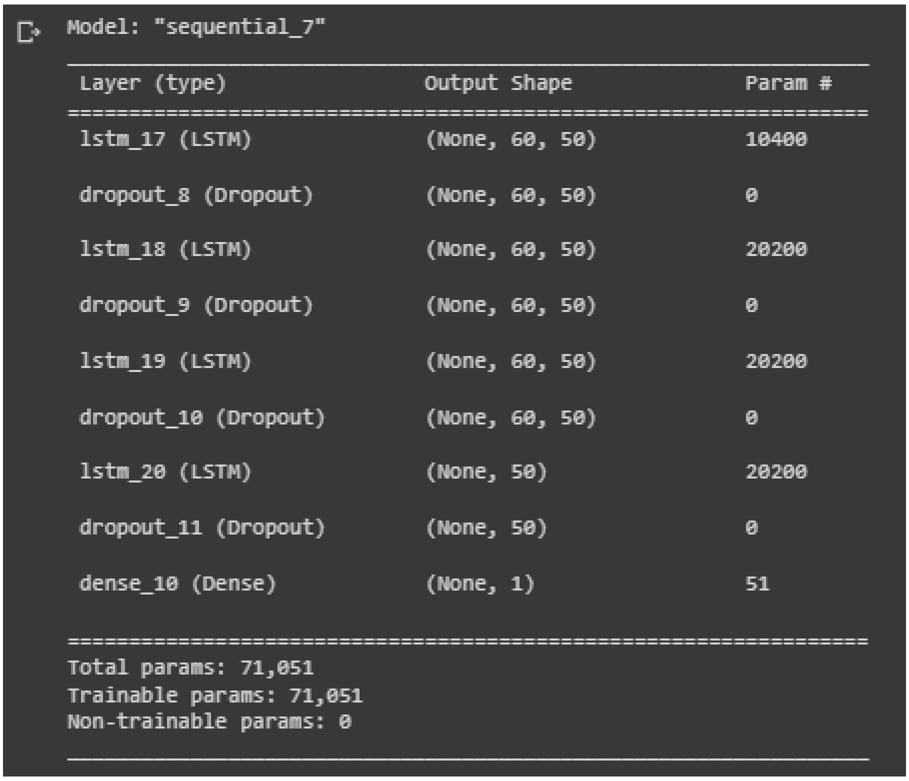
Results images of 4 stock codes IMGN, AMD, F, and UBER in terms of applying the Long-Short Term Memory model to predict stock price or trends

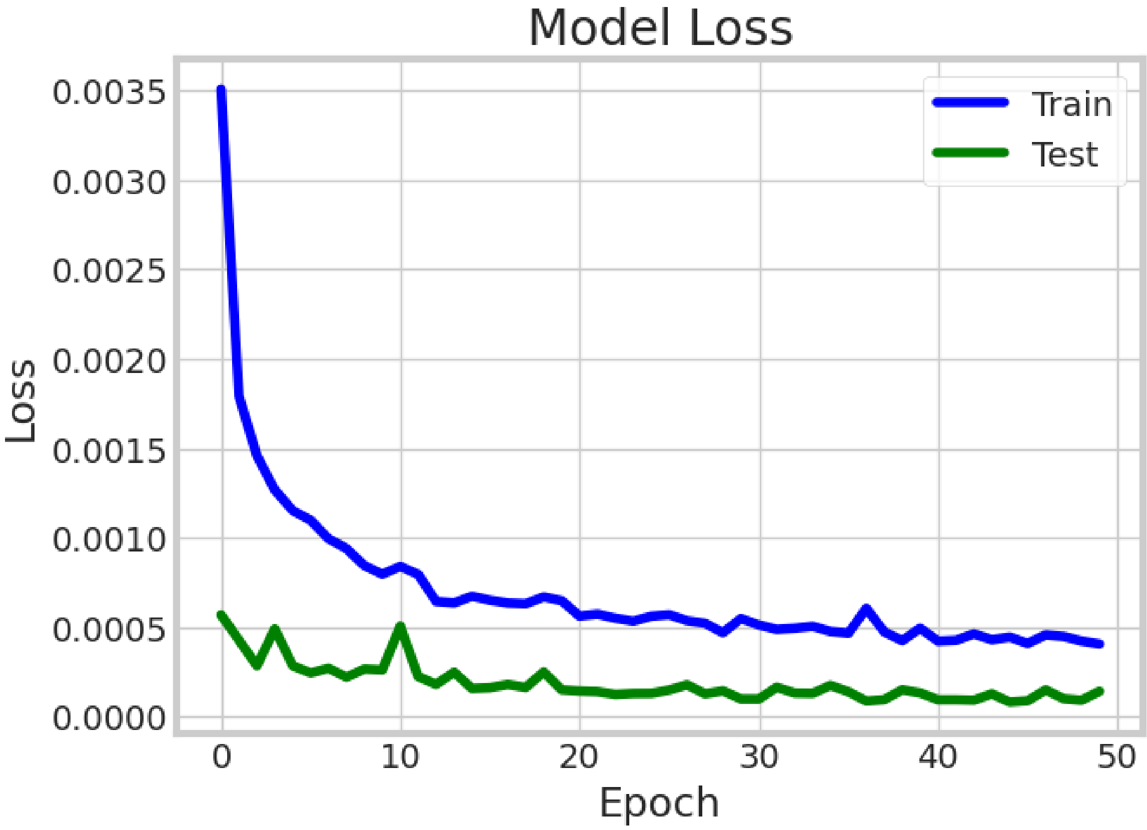
1. ImmunoGen, Inc. (IMGN)

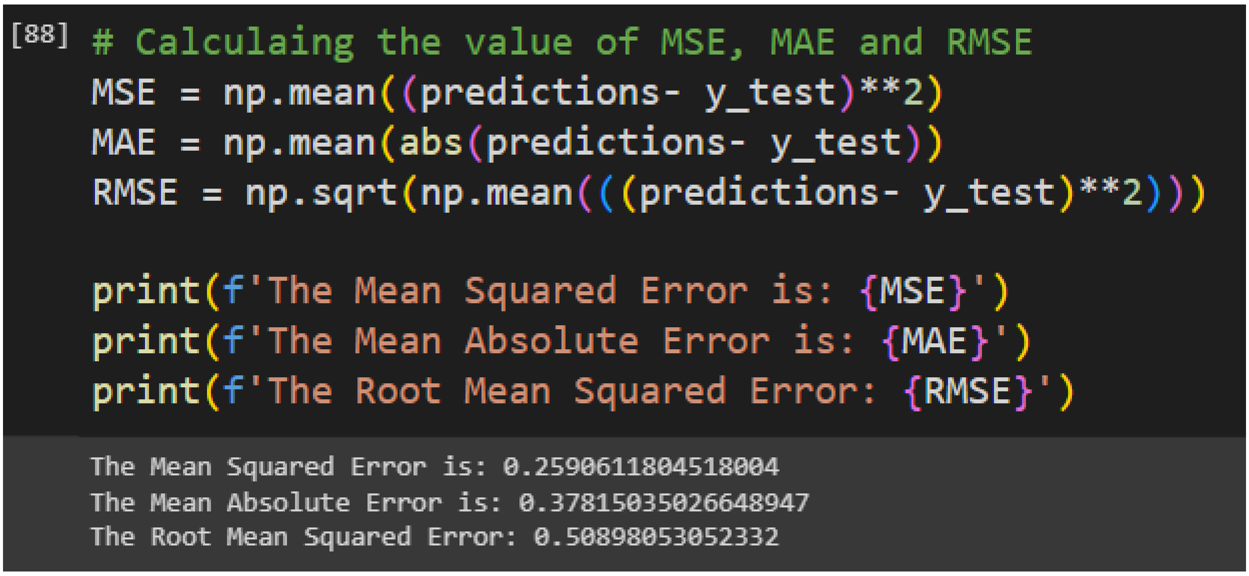


```
[*********************100%***********************]  1 of 1 completed
Name of stock code: IMGN
             Open   High    Low  Close  Adj Close   Volume
Date
2000-05-01  11.00  11.25  10.25  10.50      10.50   351100
2000-05-02  10.56  11.50  10.44  10.75      10.75   581800
2000-05-03  11.00  11.25  10.00  10.38      10.38   288300
2000-05-04  11.50  11.50  10.25  10.50      10.50   947300
2000-05-05  11.38  12.00  10.94  11.56      11.56   983000
   ...       ...    ...    ...    ...        ...      ...
2023-05-15  13.79  14.30  13.35  13.48      13.48  6240600
2023-05-16  13.10  13.87  12.92  13.69      13.69  6461500
2023-05-17  13.75  14.02  13.50  13.90      13.90  5513000
2023-05-18  13.90  13.92  13.46  13.73      13.73  4306700
2023-05-19  13.87  14.91  13.67  14.25      14.25  8648500
5801 rows × 6 columns
```



```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_17 (LSTM)              (None, 60, 50)            10400

 dropout_8 (Dropout)         (None, 60, 50)            0

 lstm_18 (LSTM)              (None, 60, 50)            20200

 dropout_9 (Dropout)         (None, 60, 50)            0

 lstm_19 (LSTM)              (None, 60, 50)            20200

 dropout_10 (Dropout)        (None, 60, 50)            0

 lstm_20 (LSTM)              (None, 50)                20200

 dropout_11 (Dropout)        (None, 50)                0

 dense_10 (Dense)            (None, 1)                 51

=================================================================
Total params: 71,051
Trainable params: 71,051
Non-trainable params: 0
_________________________________________________________________
```

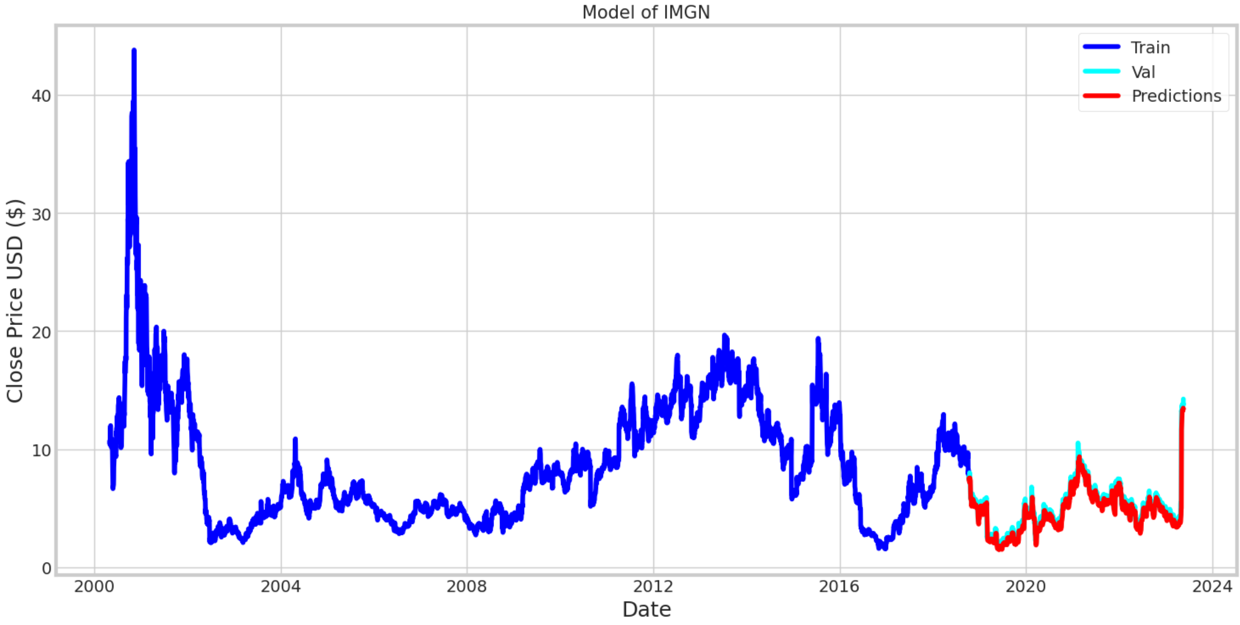


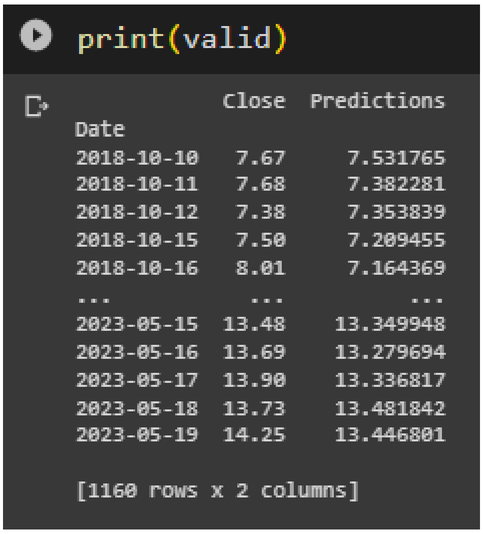


[88]

```
# Calculaing the value of MSE, MAE and RMSE
MSE = np.mean((predictions- y_test)**2)
MAE = np.mean(abs(predictions- y_test))
RMSE = np.sqrt(np.mean(((predictions- y_test)**2)))

print(f'The Mean Squared Error is: {MSE}')
print(f'The Mean Absolute Error is: {MAE}')
print(f'The Root Mean Squared Error: {RMSE}')
```

```
The Mean Squared Error is: 0.2590611804518004
The Mean Absolute Error is: 0.37815035026648947
The Root Mean Squared Error: 0.50898053052332
```

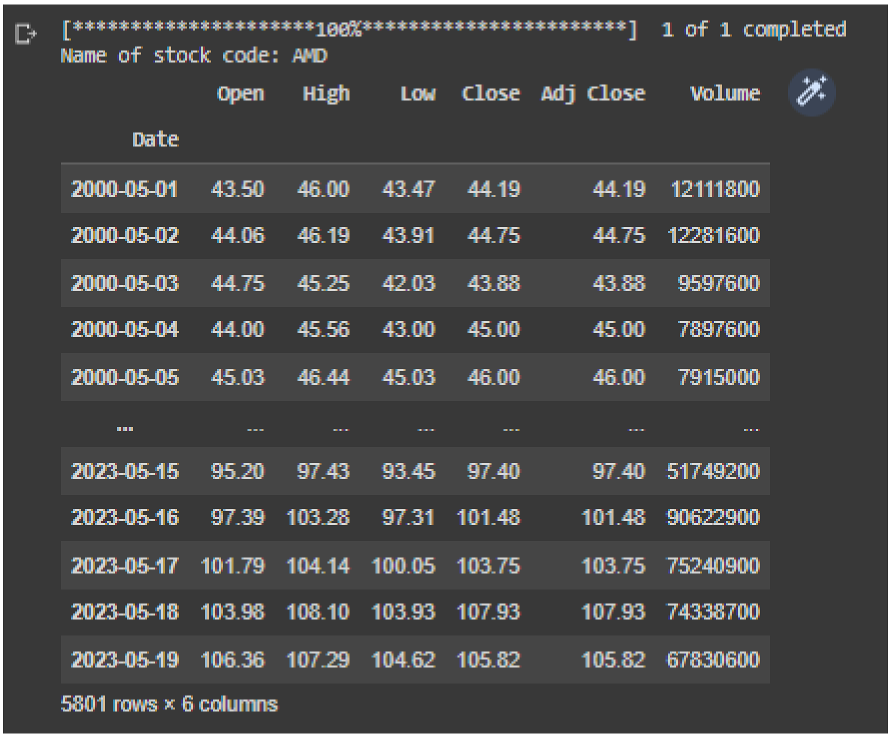




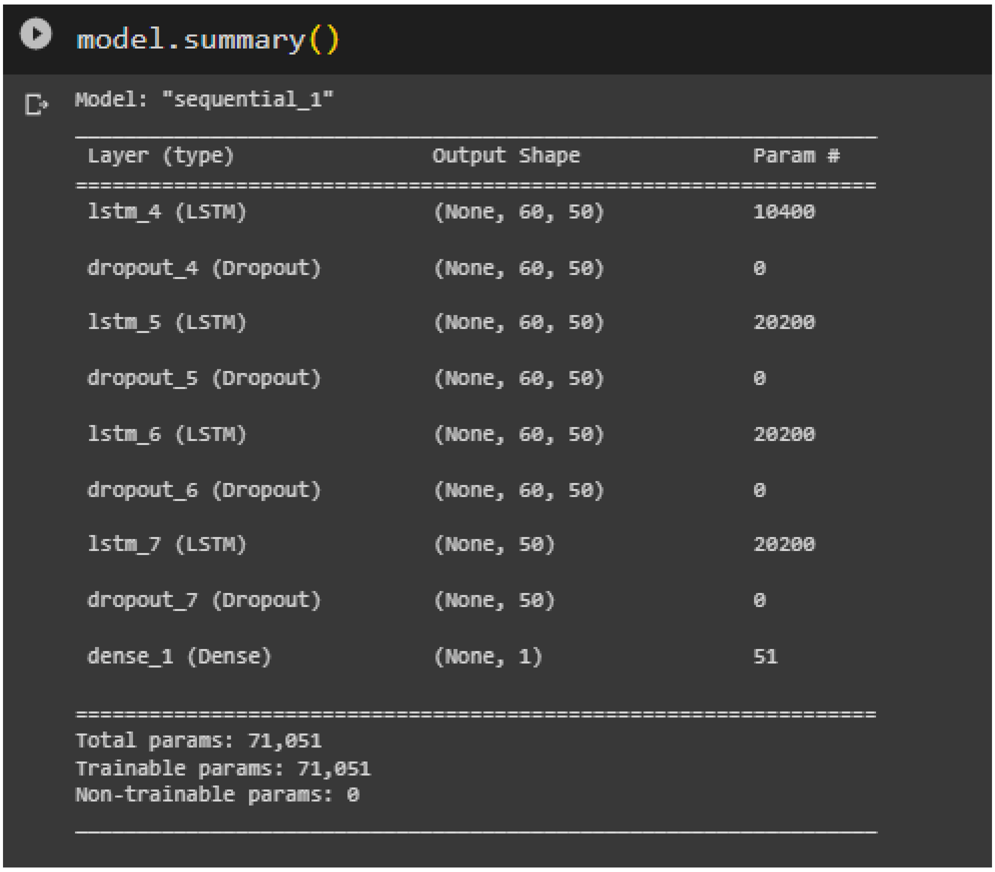
```
print(valid)
```

```
                Close  Predictions
Date
2018-10-10   7.67     7.531765
2018-10-11   7.68     7.382281
2018-10-12   7.38     7.353839
2018-10-15   7.50     7.209455
2018-10-16   8.01     7.164369
...           ...          ...
2023-05-15  13.48    13.349948
2023-05-16  13.69    13.279694
2023-05-17  13.90    13.336817
2023-05-18  13.73    13.481842
2023-05-19  14.25    13.446801

[1160 rows x 2 columns]
```

2. Advanced Micro Devices, Inc. (AMD)

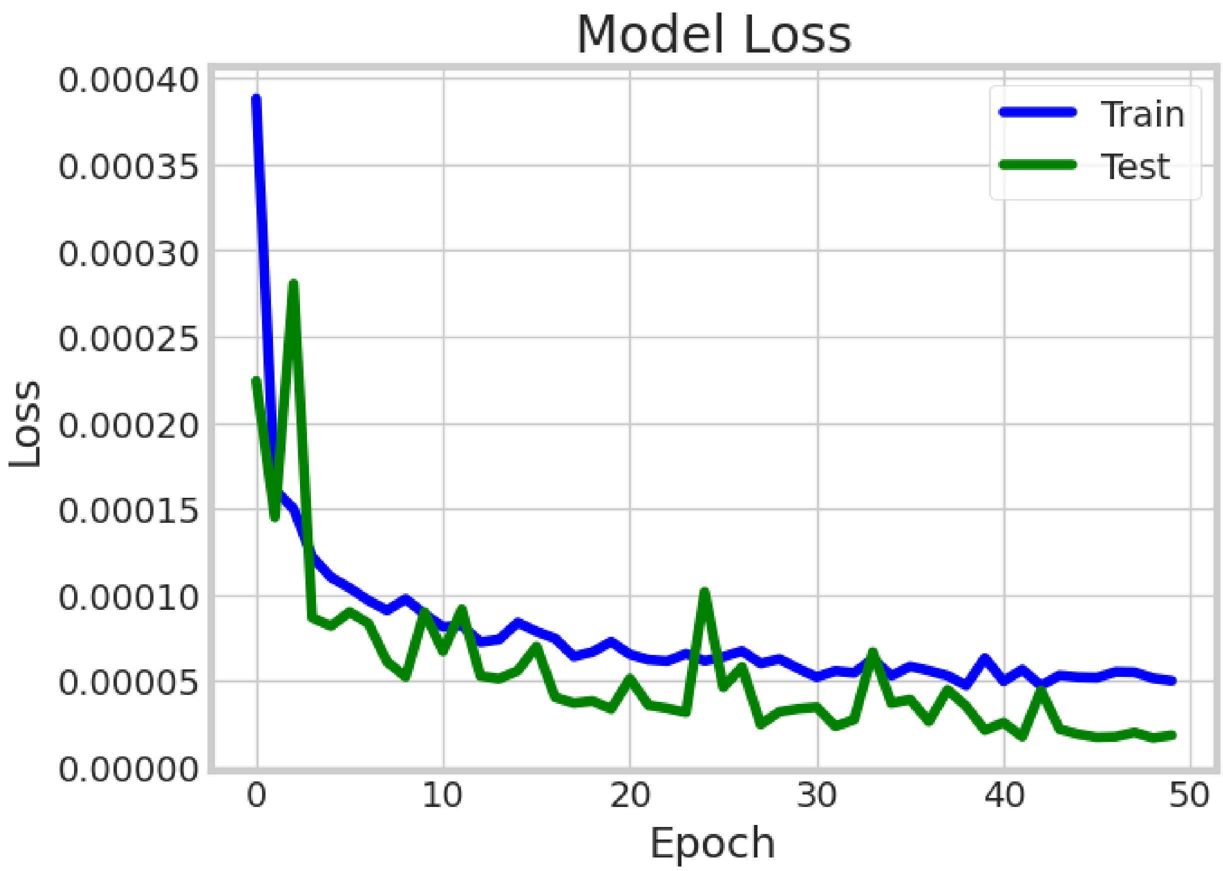


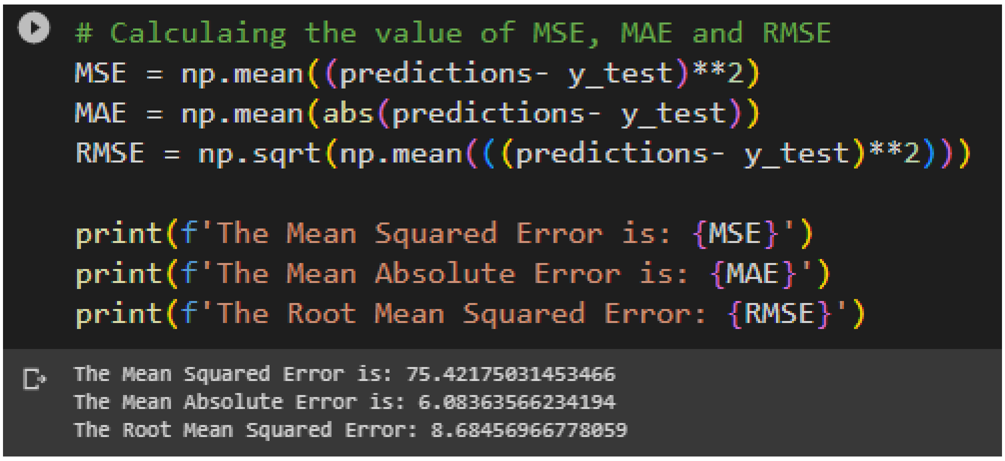
```
[*********************100%***********************]  1 of 1 completed
Name of stock code: AMD
                Open    High     Low   Close  Adj Close     Volume
      Date
2000-05-01     43.50   46.00   43.47   44.19      44.19   12111800
2000-05-02     44.06   46.19   43.91   44.75      44.75   12281600
2000-05-03     44.75   45.25   42.03   43.88      43.88    9597600
2000-05-04     44.00   45.56   43.00   45.00      45.00    7897600
2000-05-05     45.03   46.44   45.03   46.00      46.00    7915000
   ...           ...     ...     ...     ...        ...        ...
2023-05-15     95.20   97.43   93.45   97.40      97.40   51749200
2023-05-16     97.39  103.28   97.31  101.48     101.48   90622900
2023-05-17    101.79  104.14  100.05  103.75     103.75   75240900
2023-05-18    103.98  108.10  103.93  107.93     107.93   74338700
2023-05-19    106.36  107.29  104.62  105.82     105.82   67830600
5801 rows × 6 columns
```



```
model.summary()
```

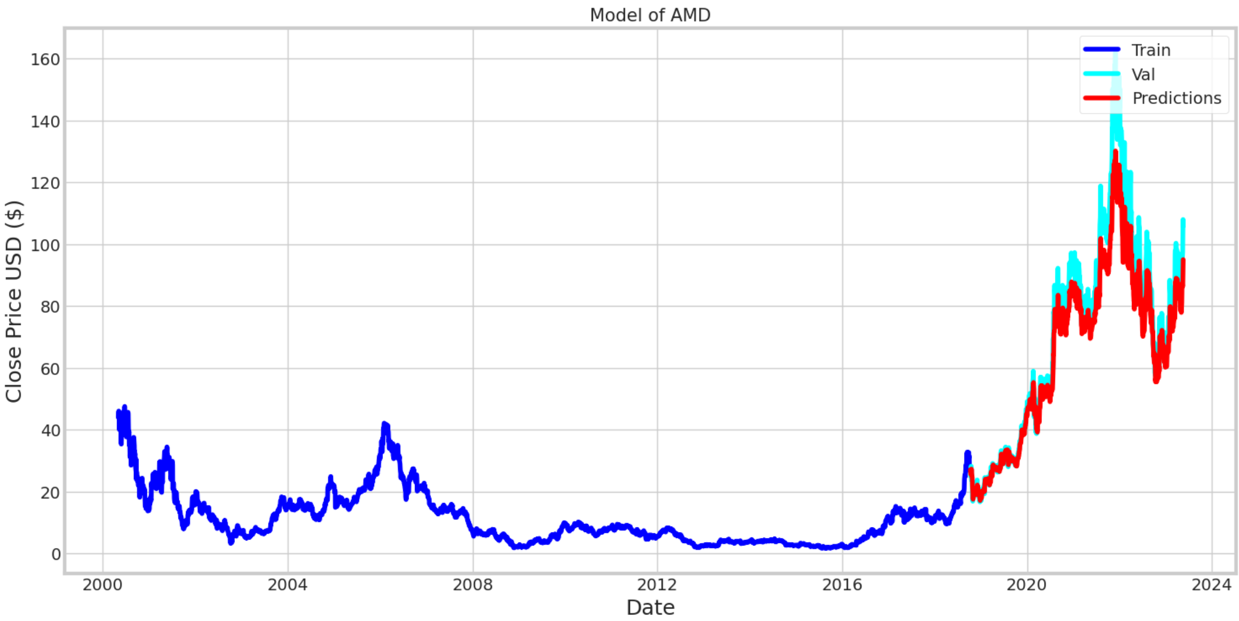
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 60, 50)            10400

 dropout_4 (Dropout)         (None, 60, 50)            0

 lstm_5 (LSTM)               (None, 60, 50)            20200

 dropout_5 (Dropout)         (None, 60, 50)            0

 lstm_6 (LSTM)               (None, 60, 50)            20200

 dropout_6 (Dropout)         (None, 60, 50)            0

 lstm_7 (LSTM)               (None, 50)                20200

 dropout_7 (Dropout)         (None, 50)                0

 dense_1 (Dense)             (None, 1)                 51

=================================================================
Total params: 71,051
Trainable params: 71,051
Non-trainable params: 0
_________________________________________________________________
```

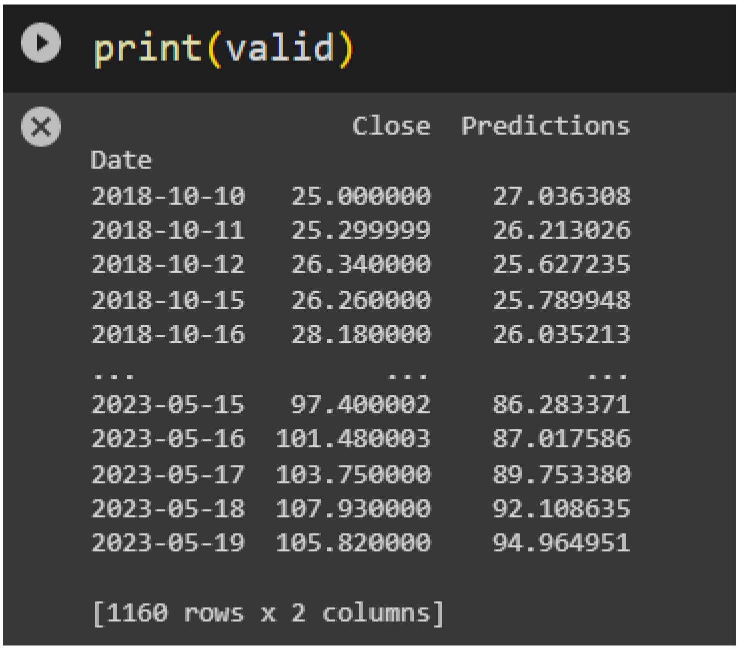




```
# Calculaing the value of MSE, MAE and RMSE
MSE = np.mean((predictions- y_test)**2)
MAE = np.mean(abs(predictions- y_test))
RMSE = np.sqrt(np.mean(((predictions- y_test)**2)))

print(f'The Mean Squared Error is: {MSE}')
print(f'The Mean Absolute Error is: {MAE}')
print(f'The Root Mean Squared Error: {RMSE}')
```

```
The Mean Squared Error is: 75.42175031453466
The Mean Absolute Error is: 6.08363566234194
The Root Mean Squared Error: 8.68456966778059
```

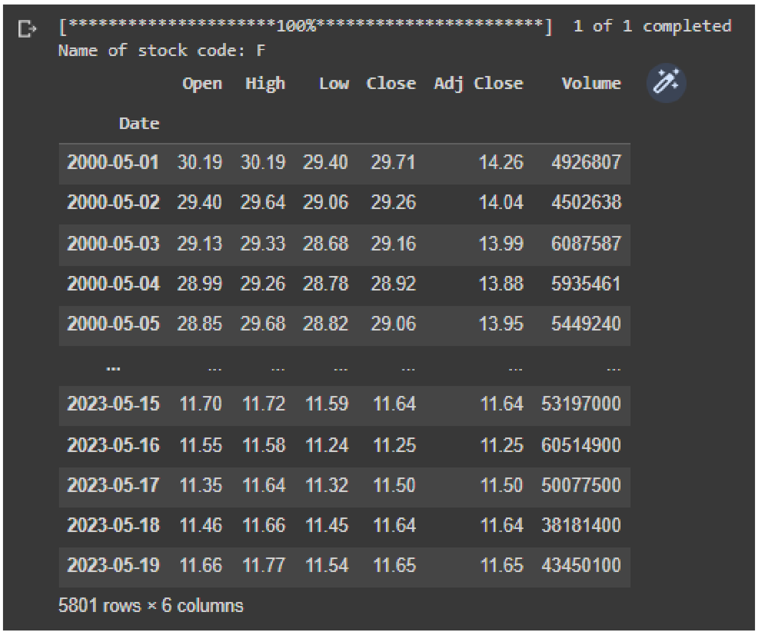




```
print(valid)
```

```
                 Close  Predictions
Date
2018-10-10   25.000000    27.036308
2018-10-11   25.299999    26.213026
2018-10-12   26.340000    25.627235
2018-10-15   26.260000    25.789948
2018-10-16   28.180000    26.035213
...                ...          ...
2023-05-15   97.400002    86.283371
2023-05-16  101.480003    87.017586
2023-05-17  103.750000    89.753380
2023-05-18  107.930000    92.108635
2023-05-19  105.820000    94.964951

[1160 rows x 2 columns]
```

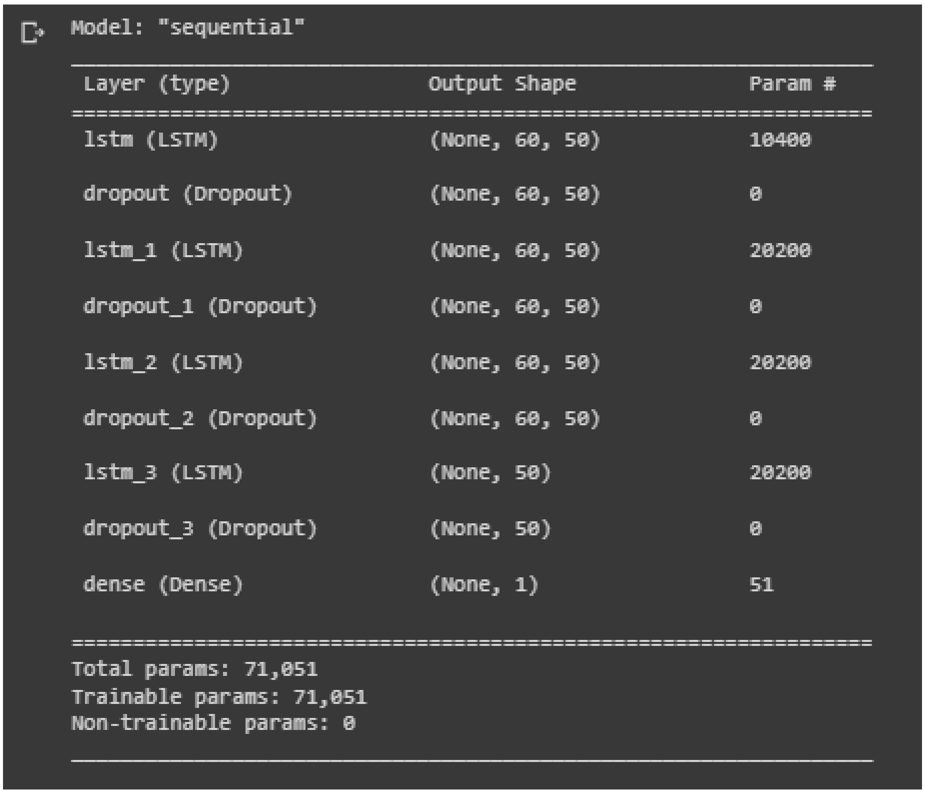
3. Ford Motor Company (F)



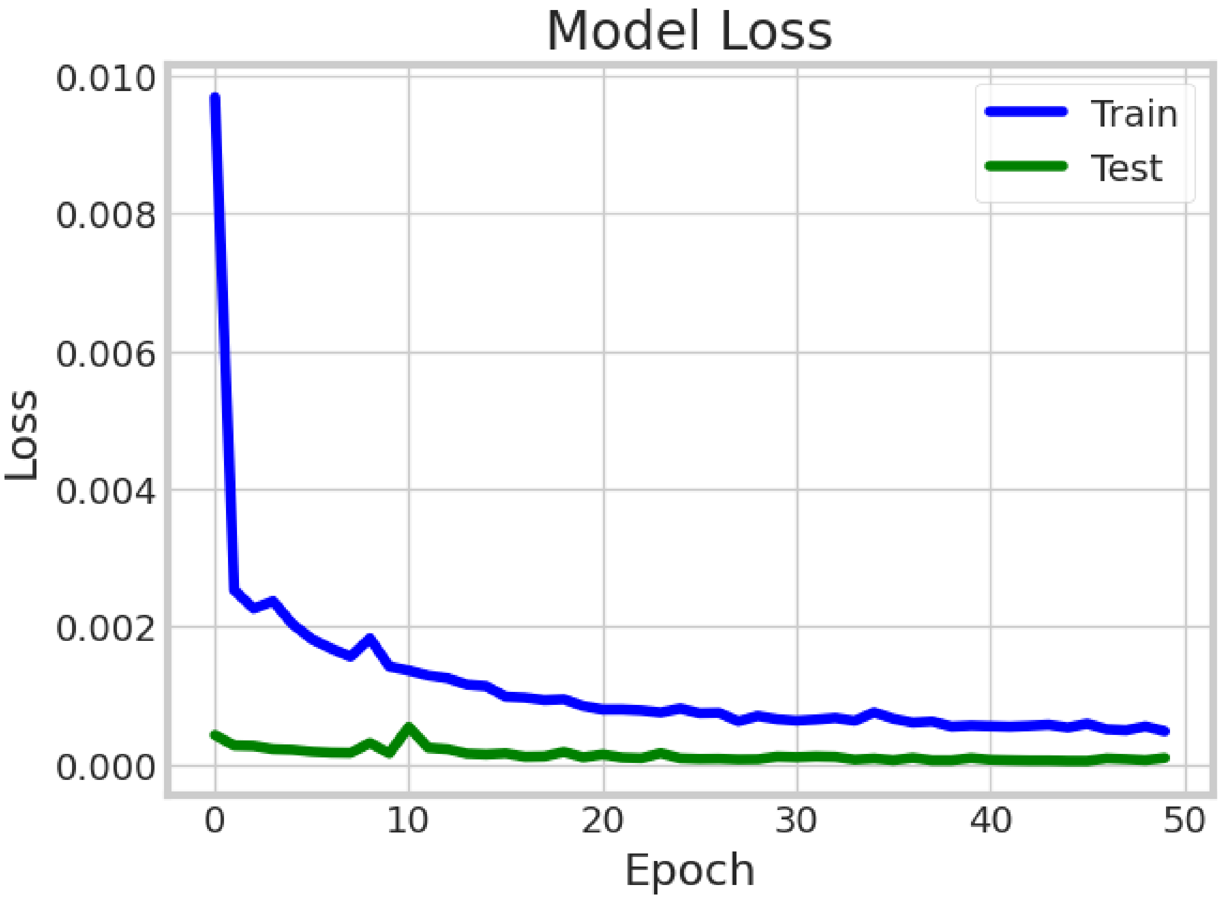
```
[*********************100%***********************]  1 of 1 completed
Name of stock code: F
```

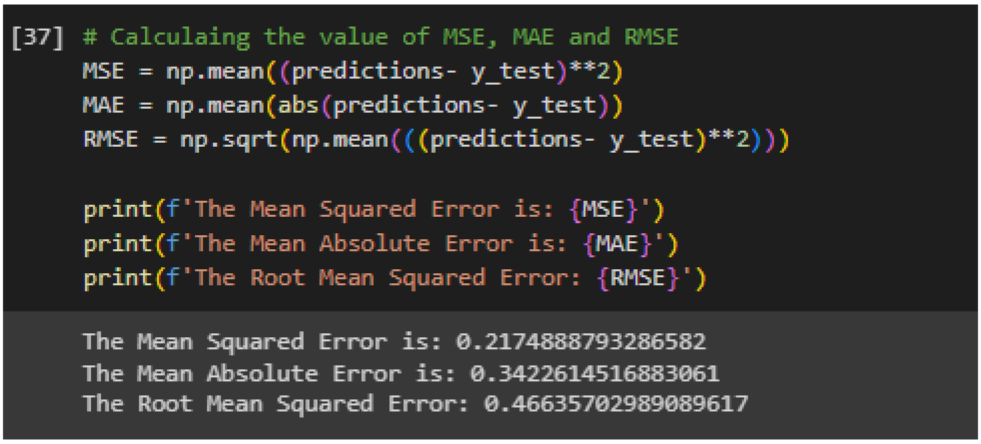
Date	Open	High	Low	Close	Adj Close	Volume
2000-05-01	30.19	30.19	29.40	29.71	14.26	4926807
2000-05-02	29.40	29.64	29.06	29.26	14.04	4502638
2000-05-03	29.13	29.33	28.68	29.16	13.99	6087587
2000-05-04	28.99	29.26	28.78	28.92	13.88	5935461
2000-05-05	28.85	29.68	28.82	29.06	13.95	5449240
...	...	...	...	...	...	...
2023-05-15	11.70	11.72	11.59	11.64	11.64	53197000
2023-05-16	11.55	11.58	11.24	11.25	11.25	60514900
2023-05-17	11.35	11.64	11.32	11.50	11.50	50077500
2023-05-18	11.46	11.66	11.45	11.64	11.64	38181400
2023-05-19	11.66	11.77	11.54	11.65	11.65	43450100

5801 rows × 6 columns



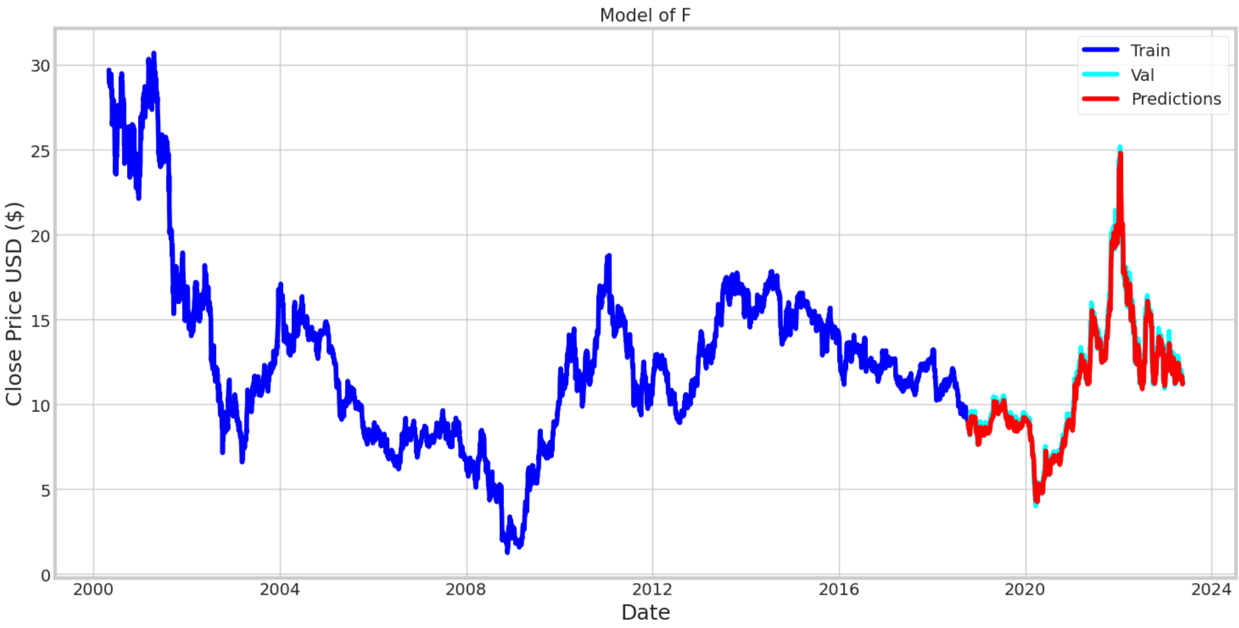
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 lstm (LSTM)                 (None, 60, 50)            10400     
                                                                 
 dropout (Dropout)           (None, 60, 50)            0         
                                                                 
 lstm_1 (LSTM)               (None, 60, 50)            20200     
                                                                 
 dropout_1 (Dropout)         (None, 60, 50)            0         
                                                                 
 lstm_2 (LSTM)               (None, 60, 50)            20200     
                                                                 
 dropout_2 (Dropout)         (None, 60, 50)            0         
                                                                 
 lstm_3 (LSTM)               (None, 50)                20200     
                                                                 
 dropout_3 (Dropout)         (None, 50)                0         
                                                                 
 dense (Dense)               (None, 1)                 51        
                                                                 
=================================================================
Total params: 71,051
Trainable params: 71,051
Non-trainable params: 0
_________________________________________________________________
```

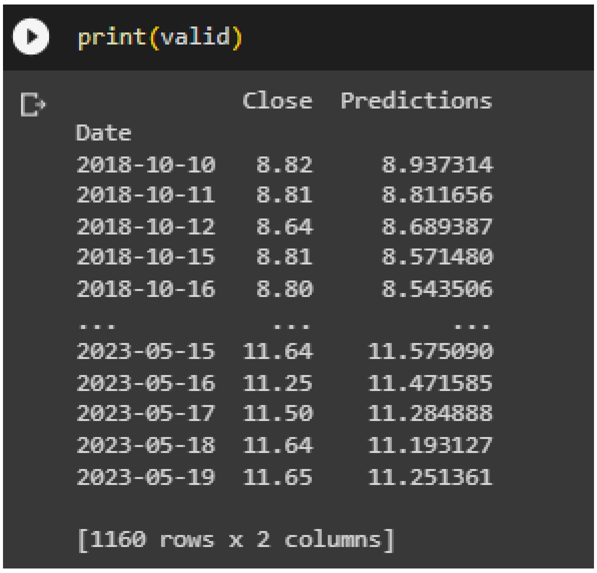




```
[37] # Calculaing the value of MSE, MAE and RMSE
     MSE = np.mean((predictions- y_test)**2)
     MAE = np.mean(abs(predictions- y_test))
     RMSE = np.sqrt(np.mean(((predictions- y_test)**2)))

     print(f'The Mean Squared Error is: {MSE}')
     print(f'The Mean Absolute Error is: {MAE}')
     print(f'The Root Mean Squared Error: {RMSE}')
```

```
     The Mean Squared Error is: 0.2174888793286582
     The Mean Absolute Error is: 0.3422614516883061
     The Root Mean Squared Error: 0.46635702989089617
```

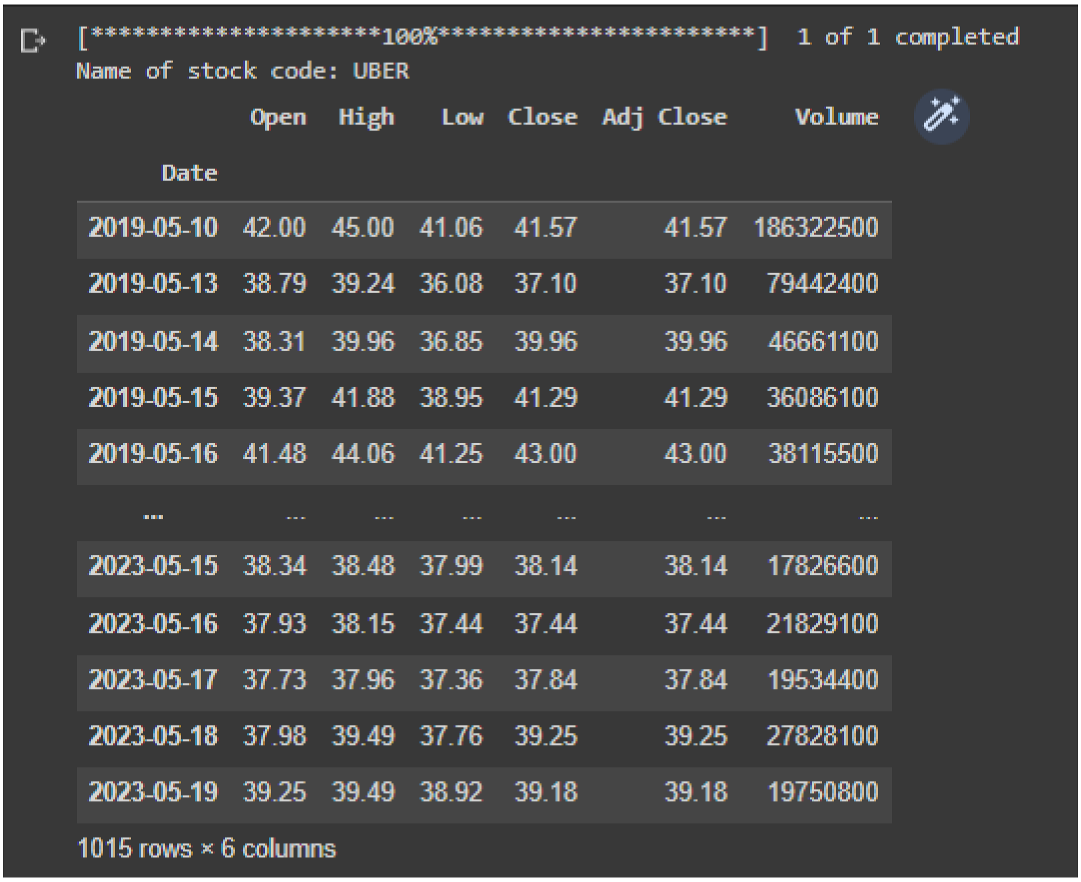




```
print(valid)
```

```
                Close  Predictions
Date
2018-10-10   8.82     8.937314
2018-10-11   8.81     8.811656
2018-10-12   8.64     8.689387
2018-10-15   8.81     8.571480
2018-10-16   8.80     8.543506
...           ...          ...
2023-05-15  11.64    11.575090
2023-05-16  11.25    11.471585
2023-05-17  11.50    11.284888
2023-05-18  11.64    11.193127
2023-05-19  11.65    11.251361

[1160 rows x 2 columns]
```

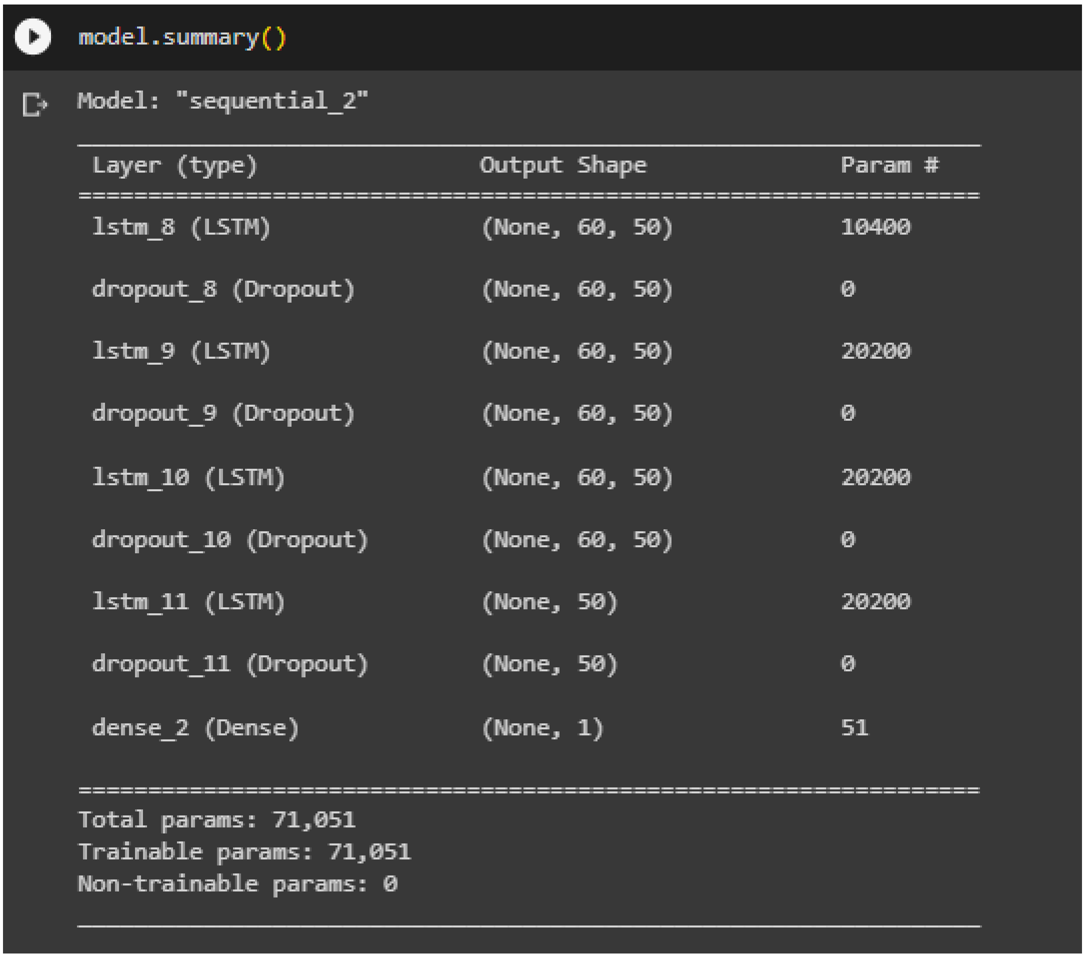
4. Uber Technologies, Inc. (UBER)



```
[*********************100%***********************]  1 of 1 completed
Name of stock code: UBER
```

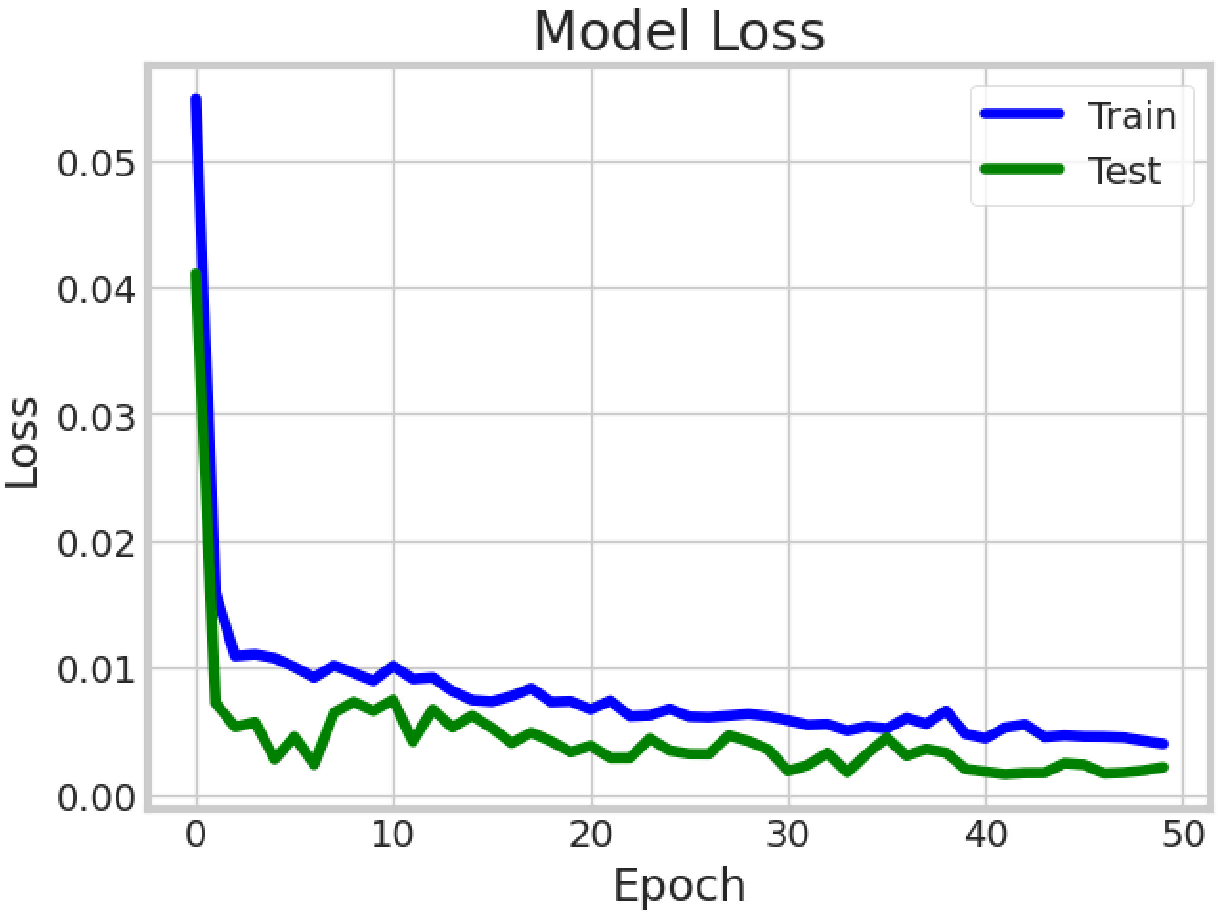
Date	Open	High	Low	Close	Adj Close	Volume
2019-05-10	42.00	45.00	41.06	41.57	41.57	186322500
2019-05-13	38.79	39.24	36.08	37.10	37.10	79442400
2019-05-14	38.31	39.96	36.85	39.96	39.96	46661100
2019-05-15	39.37	41.88	38.95	41.29	41.29	36086100
2019-05-16	41.48	44.06	41.25	43.00	43.00	38115500
...	...	...	...	...	...	...
2023-05-15	38.34	38.48	37.99	38.14	38.14	17826600
2023-05-16	37.93	38.15	37.44	37.44	37.44	21829100
2023-05-17	37.73	37.96	37.36	37.84	37.84	19534400
2023-05-18	37.98	39.49	37.76	39.25	39.25	27828100
2023-05-19	39.25	39.49	38.92	39.18	39.18	19750800

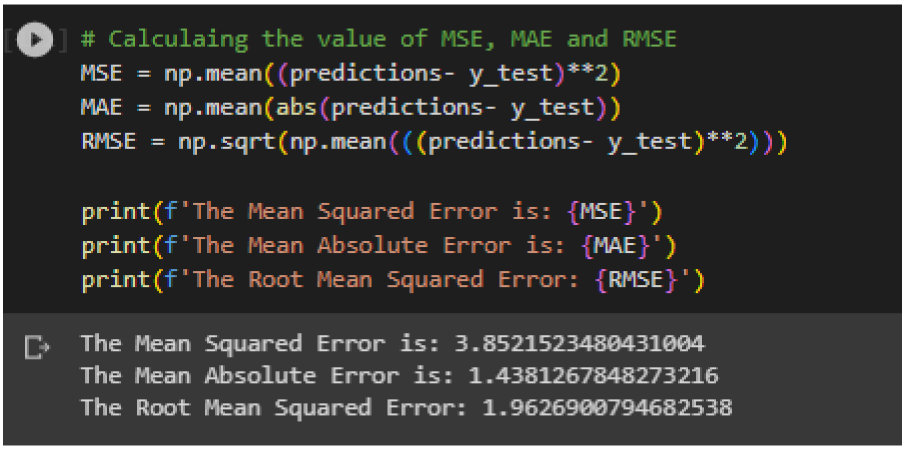
1015 rows × 6 columns



```
model.summary()
```

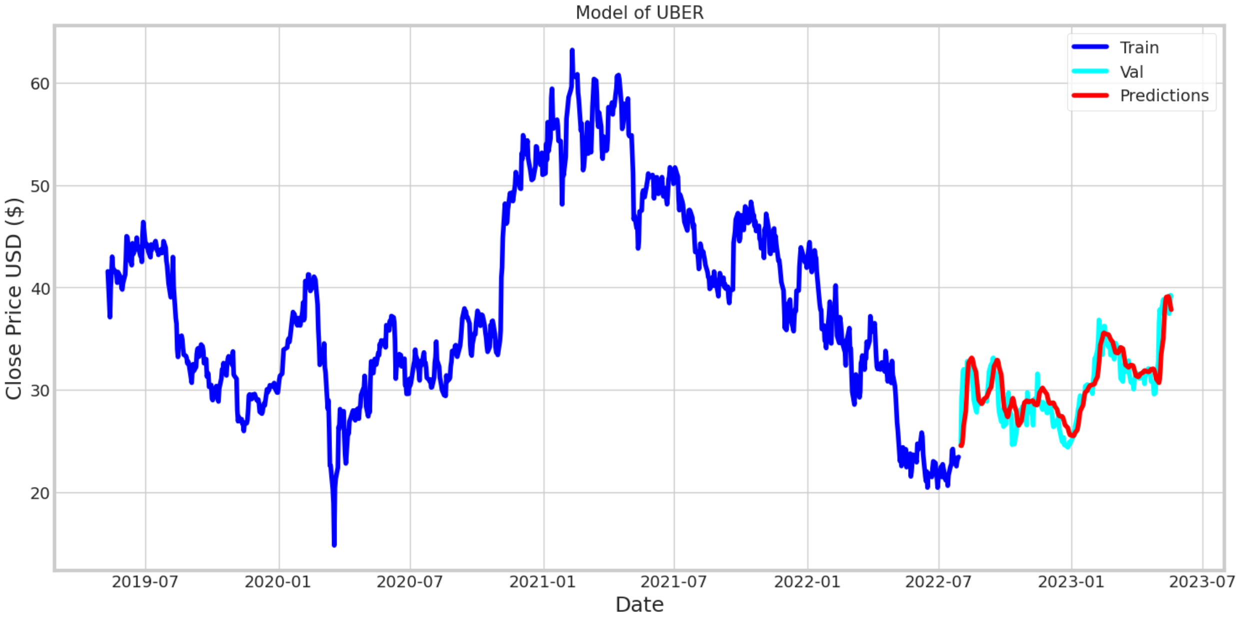
```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_8 (LSTM)               (None, 60, 50)            10400

 dropout_8 (Dropout)         (None, 60, 50)            0

 lstm_9 (LSTM)               (None, 60, 50)            20200

 dropout_9 (Dropout)         (None, 60, 50)            0

 lstm_10 (LSTM)              (None, 60, 50)            20200

 dropout_10 (Dropout)        (None, 60, 50)            0

 lstm_11 (LSTM)              (None, 50)                20200

 dropout_11 (Dropout)        (None, 50)                0

 dense_2 (Dense)             (None, 1)                 51

=================================================================
Total params: 71,051
Trainable params: 71,051
Non-trainable params: 0
_________________________________________________________________
```

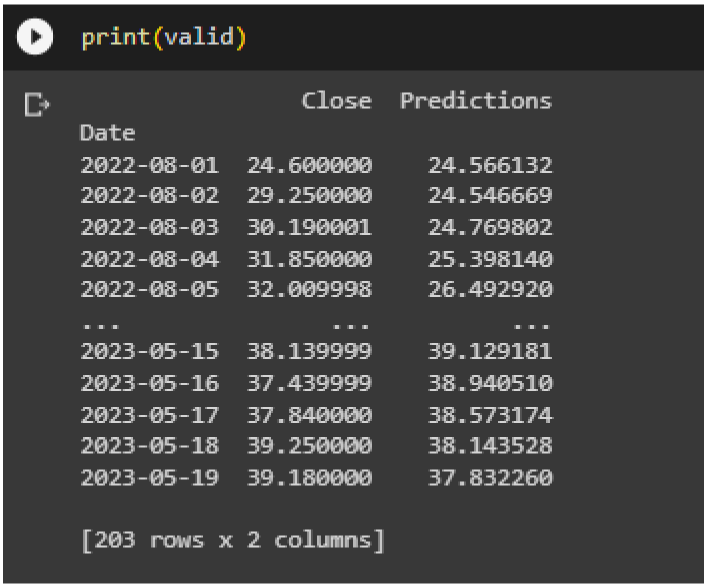




```
# Calculaing the value of MSE, MAE and RMSE
MSE = np.mean((predictions- y_test)**2)
MAE = np.mean(abs(predictions- y_test))
RMSE = np.sqrt(np.mean(((predictions- y_test)**2)))

print(f'The Mean Squared Error is: {MSE}')
print(f'The Mean Absolute Error is: {MAE}')
print(f'The Root Mean Squared Error: {RMSE}')
```

```
The Mean Squared Error is: 3.8521523480431004
The Mean Absolute Error is: 1.4381267848273216
The Root Mean Squared Error: 1.9626900794682538
```





```
print(valid)
```

```
                Close  Predictions
Date                               
2022-08-01  24.600000    24.566132
2022-08-02  29.250000    24.546669
2022-08-03  30.190001    24.769802
2022-08-04  31.850000    25.398140
2022-08-05  32.009998    26.492920
...               ...          ...
2023-05-15  38.139999    39.129181
2023-05-16  37.439999    38.940510
2023-05-17  37.840000    38.573174
2023-05-18  39.250000    38.143528
2023-05-19  39.180000    37.832260

[203 rows x 2 columns]
```