```
import math
import pandas_datareader as web
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
```

```
from datetime import datetime
```

```
df = pd.read_csv('TATAMOTORS.csv')
```

```
df
```

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP
0	2000-01-03	TELCO	EQ	201.60	207.4	217.25	207.40	217.00	216.75	214.28
1	2000-01-04	TELCO	EQ	216.75	217.0	219.00	206.00	211.90	208.20	209.50
2	2000-01-05	TELCO	EQ	208.20	194.0	217.80	194.00	213.10	213.25	210.33
3	2000-01-06	TELCO	EQ	213.25	215.0	229.90	215.00	222.00	222.10	225.29
4	2000-01-07	TELCO	EQ	222.10	224.0	239.90	223.10	239.90	239.90	236.32
...	...	...	...	...	...	...	...	...	...	...
5301	2021-04-26	TATAMOTORS	EQ	294.00	297.0	299.00	294.60	295.85	295.40	296.48
5302	2021-04-27	TATAMOTORS	EQ	295.40	295.7	302.50	295.10	302.10	301.50	299.05

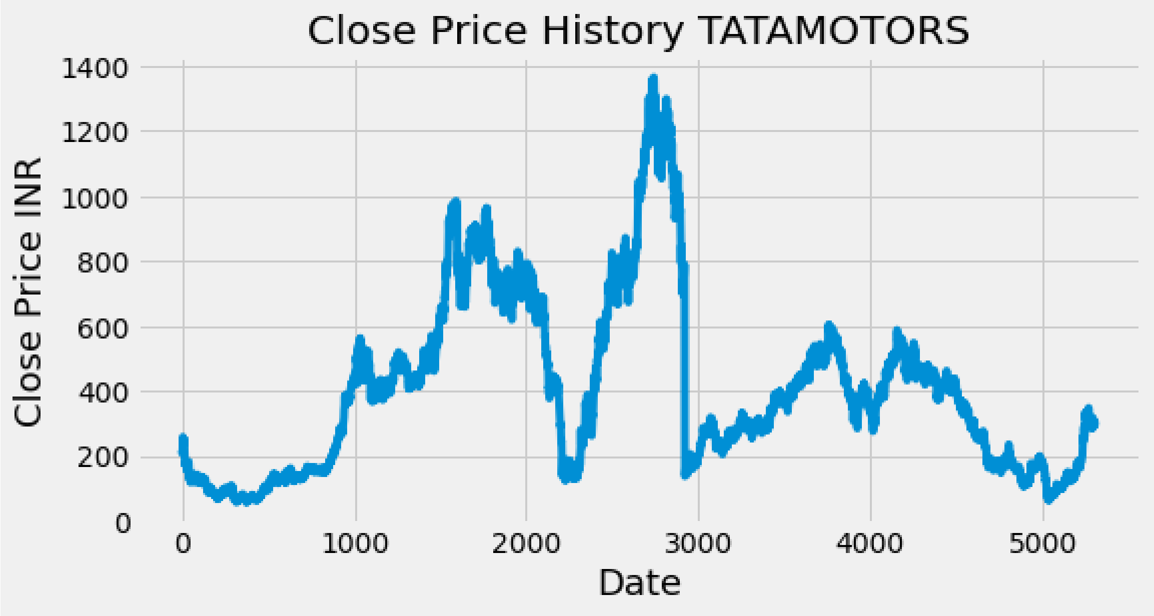
```
df['Date'] = pd.to_datetime(df['Date'])
```

```
df['Year'] = pd.DatetimeIndex(df['Date']).year
```

```
df['Year'].unique().flatten()
```

```
array([2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
       2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021])
```

```
plt.figure(figsize=(8,4))
plt.title('Close Price History TATAMOTORS')
plt.title('Close Price History TATAMOTORS')
plt.plot(df['Close'].values.flatten())
plt.xlabel('Date',fontsize=18)
plt.ylabel('Close Price INR' , fontsize=18)
plt.show()
```



```
data = df.filter(['Close'])

dataset = data.values

training_data_len = math.ceil( len(dataset) * .8)

training_data_len
```

```
4245
```

```
#Scale the data
scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(dataset)
scaled_data
```

```
array([[0.1209094 ],
       [0.11436445],
       [0.11823018],
       ...,
       [0.18915298],
       [0.18609102],
       [0.17992881]])
```

```
 #create the training data set
train_data = scaled_data[0:training_data_len , :]

x_train = []
y_train = []

for i in range(60,len(train_data)):
  x_train.append(train_data[i-60:i, 0])
  y_train.append(train_data[i,0])

x_train, y_train = np.array(x_train) , np.array(y_train)
```

```
#reshape the data
x_train = np.reshape(x_train, (x_train.shape[0],x_train.shape[1],1))
x_train.shape
```

```
(4185, 60, 1)
```

```
#Build the LSTM model
model = Sequential()
model.add(LSTM(50,return_sequences=True, input_shape = (x_train.shape[1],1)))
model.add(LSTM(50,return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
```

```
#compile the model
model.compile(optimizer ='adam', loss = 'mean_squared_error')
```

```
#train the model
model.fit(x_train, y_train, batch_size = 1, epochs=1 )
```

```
4185/4185 [==============================] - 117s 27ms/step - loss: 0.0011
<keras.callbacks.History at 0x7f57cc49f390>
```

```
#create the testing dataset
test_data = scaled_data[training_data_len - 60: , :]
x_test = []
y_test = dataset[training_data_len:, :]
for i in range(60,len(test_data)):
  x_test.append(test_data[i-60:i,0])
```

```
x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1],1))
```

```
#prediction
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)
```

```
34/34 [==============================] - 1s 17ms/step
```

```
rsme = np.sqrt(np.mean(predictions - y_test)**2)
rsme
```

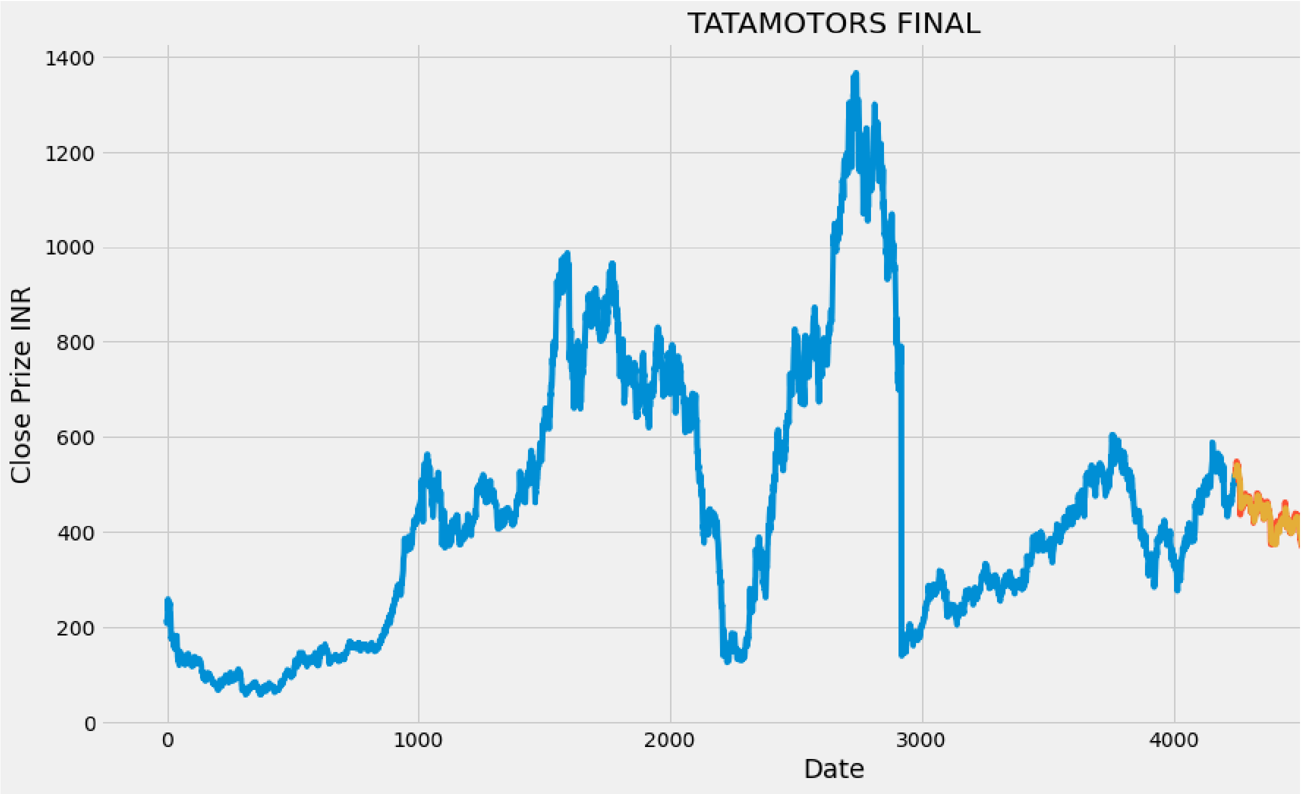
```
2.3559422571180453
```

```
#plot the data
train = data[:training_data_len]
```

```
valid = data[training_data_len:]
valid['Predictions'] = predictions
#Visualize
plt.figure(figsize = (16,8))
plt.title('TATAMOTORS FINAL')
plt.xlabel('Date',fontsize= 18)
plt.ylabel('Close Prize INR', fontsize = 18)
plt.plot(train['Close'])
plt.plot(valid[['Close','Predictions']])
plt.legend(['Train','Valid','Predictions'], loc = 'lower right')
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: SettingWithCopyWarnir
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  after removing the cwd from sys.path.
```



```
# show teh valid and predicted price
valid
```

	Close	Predictions
4245	526.40	513.436707
4246	523.70	516.792297
4247	522.45	518.773254
4248	531.45	519.268005
4249	523.20	522.449707
...	...	...
5301	295.40	292.855316
5302	301.50	291.849335
5303	305.90	293.669250
5304	301.90	297.239685
5305	293.85	298.861053

1061 rows × 2 columns

Colab paid products - Cancel contracts here