FMC - LAB-5

Name: Sarvagya Singh

SAP: 60009200030

Div/Batch: K/K1

Lab 5: LSTM to analyze historical stock.

In [ ]:

```
!pip install yfinance
```

In [ ]:

```
import yfinance as yf
```

In [ ]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('whitegrid')
plt.style.use("fivethirtyeight")

# For reading stock data from yahoo
import yfinance as yf
from datetime import datetime
from pandas_datareader import data as pdr
from pandas_datareader.data import DataReader
yf.pdr_override()
```

1. Import data from Yfinance.

2. Plot Historic Data for specific stock between given duration.

In [ ]:

```
# The tech stocks we'll use for this analysis
stock = 'TSLA'

# Set up End and Start times for data grab
end = datetime.now()
start = datetime(2000, 1, 1)
start = '2020-01-01'

df = yf.download(stock, start, end)
df.tail(10)
```

```
[**********************100%**********************]  1 of 1 completed
```

Out[ ]:

Date	Open	High	Low	Close	Adj Close	Volume
2023-03-31	197.529999	207.789993	197.199997	207.460007	207.460007	169638500
2023-04-03	199.910004	202.690002	192.199997	194.770004	194.770004	169545900
2023-04-04	197.320007	198.740005	190.320007	192.580002	192.580002	126463800

Date	Open	High	Low	Close	Adj Close	Volume
2023-04-05	190.520004	190.679993	183.759995	185.520004	185.520004	133882500
2023-04-06	183.080002	186.389999	179.740005	185.059998	185.059998	123857900
2023-04-10	179.940002	185.100006	176.110001	184.509995	184.509995	142154600
2023-04-11	186.690002	189.190002	185.649994	186.789993	186.789993	115770900
2023-04-12	190.740005	191.580002	180.309998	180.539993	180.539993	150256300
2023-04-13	182.960007	186.500000	180.940002	185.899994	185.899994	112933000
2023-04-14	183.949997	186.279999	182.009995	185.000000	185.000000	96306500

In [ ]:

```
# Summary Stats
df.describe()
```

Out[ ]:

	Open	High	Low	Close	Adj Close	Volume
count	827.000000	827.000000	827.000000	827.000000	827.000000	8.270000e+02
mean	203.901476	208.699398	198.613740	203.758959	203.758959	1.350576e+08
std	92.669967	94.573502	90.404644	92.445508	92.445508	9.696314e+07
min	24.980000	26.990667	23.367332	24.081333	24.081333	2.940180e+07
25%	139.648331	142.879997	135.945000	139.983337	139.983337	7.125075e+07
50%	217.843338	222.046661	210.139999	216.866669	216.866669	9.924120e+07
75%	271.750000	276.308334	264.368332	270.083328	270.083328	1.726955e+08
max	411.470001	414.496674	405.666656	409.970001	409.970001	9.140820e+08

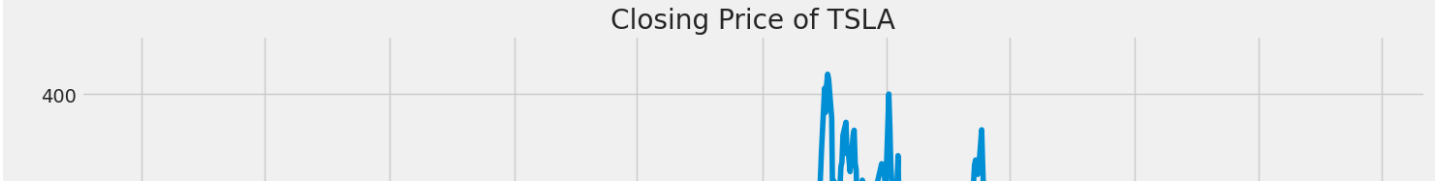
In [ ]:

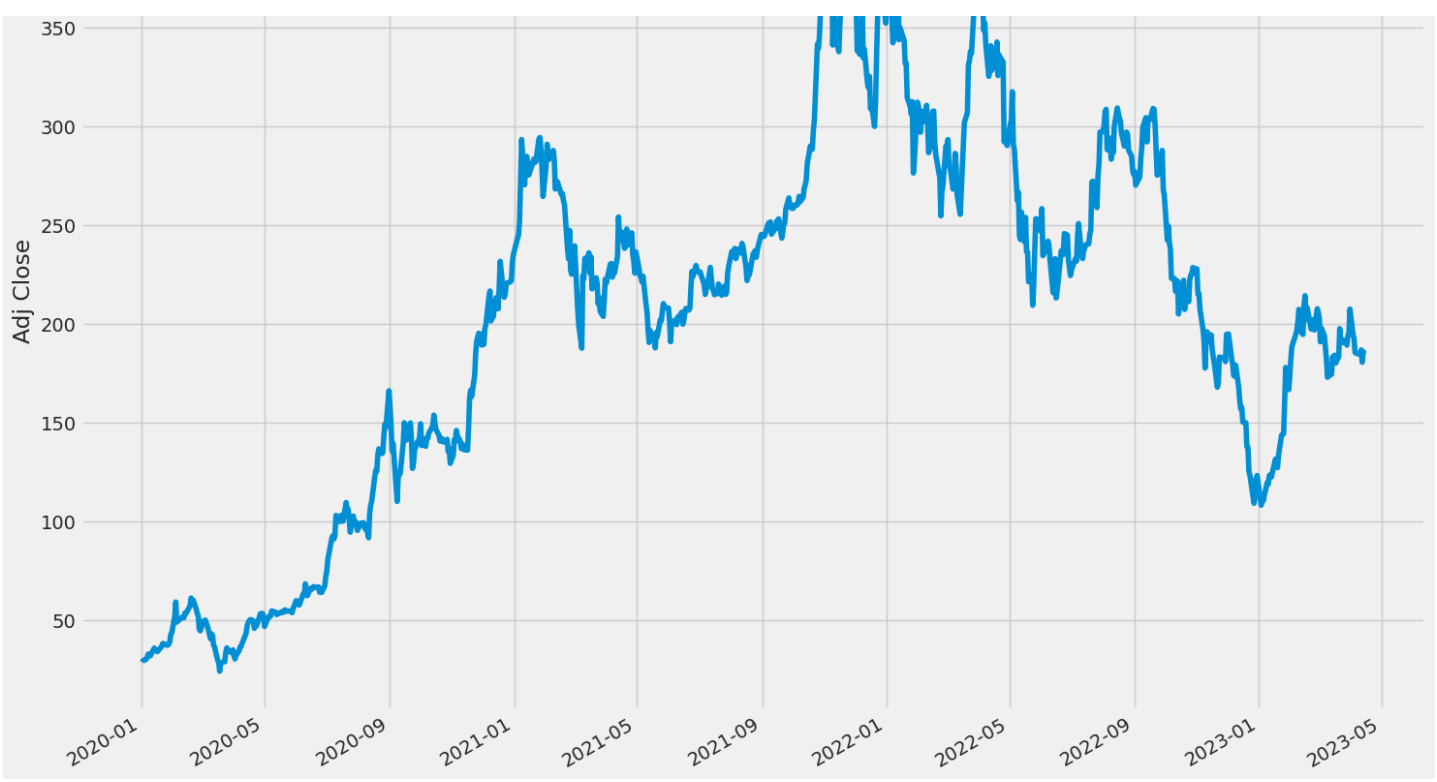
```
# General info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 827 entries, 2020-01-02 to 2023-04-14
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Open       827 non-null    float64
 1   High       827 non-null    float64
 2   Low        827 non-null    float64
 3   Close      827 non-null    float64
 4   Adj Close  827 non-null    float64
 5   Volume     827 non-null    int64  
dtypes: float64(5), int64(1)
memory usage: 45.2 KB
```

In [ ]:

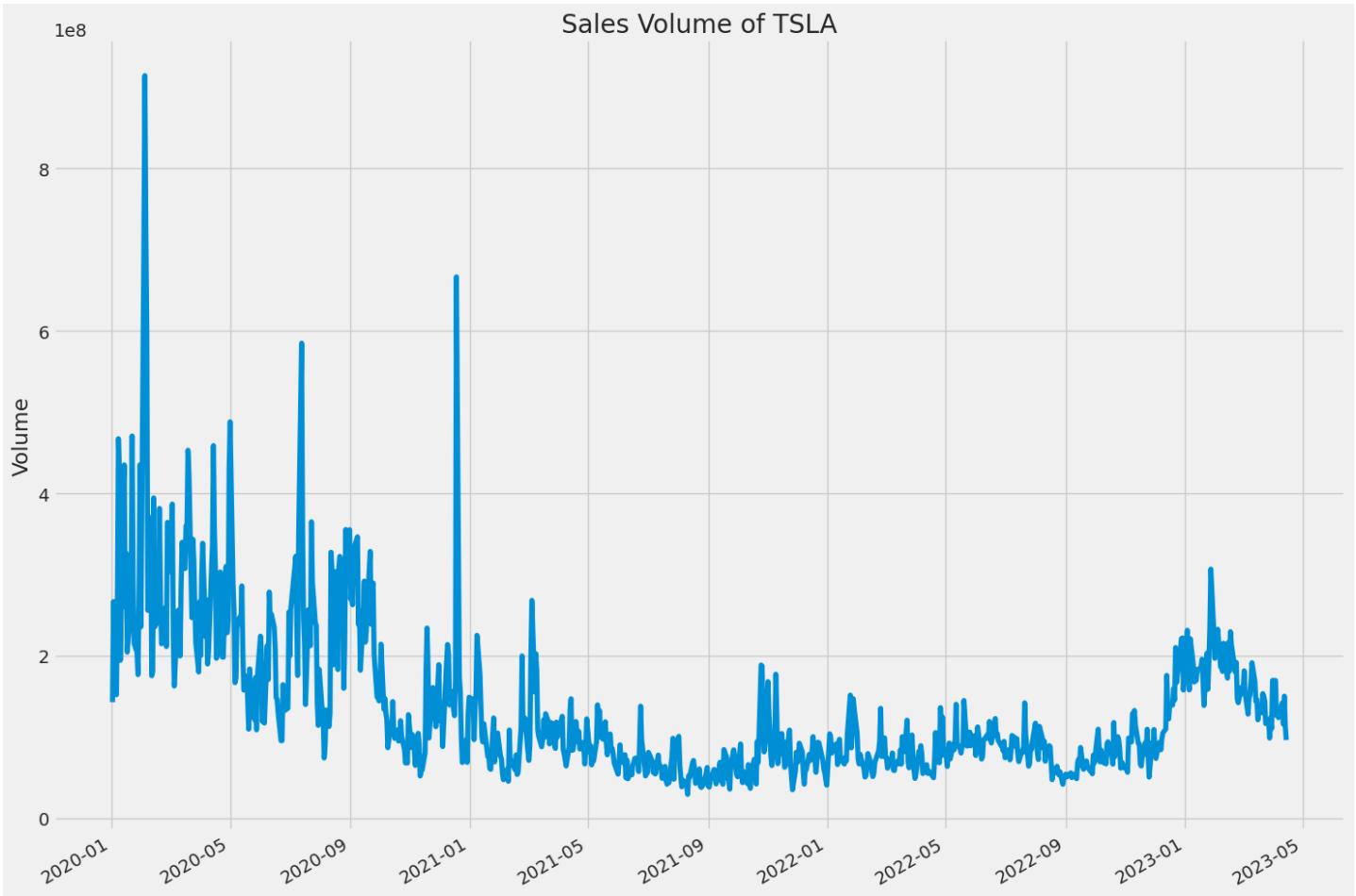
```
# Let's see a historical view of the closing price
plt.figure(figsize=(15, 10))
df['Adj Close'].plot()
plt.ylabel('Adj Close')
plt.xlabel(None)
plt.title(f"Closing Price of {stock}")

plt.tight_layout()
plt.show()
```





In [ ]:

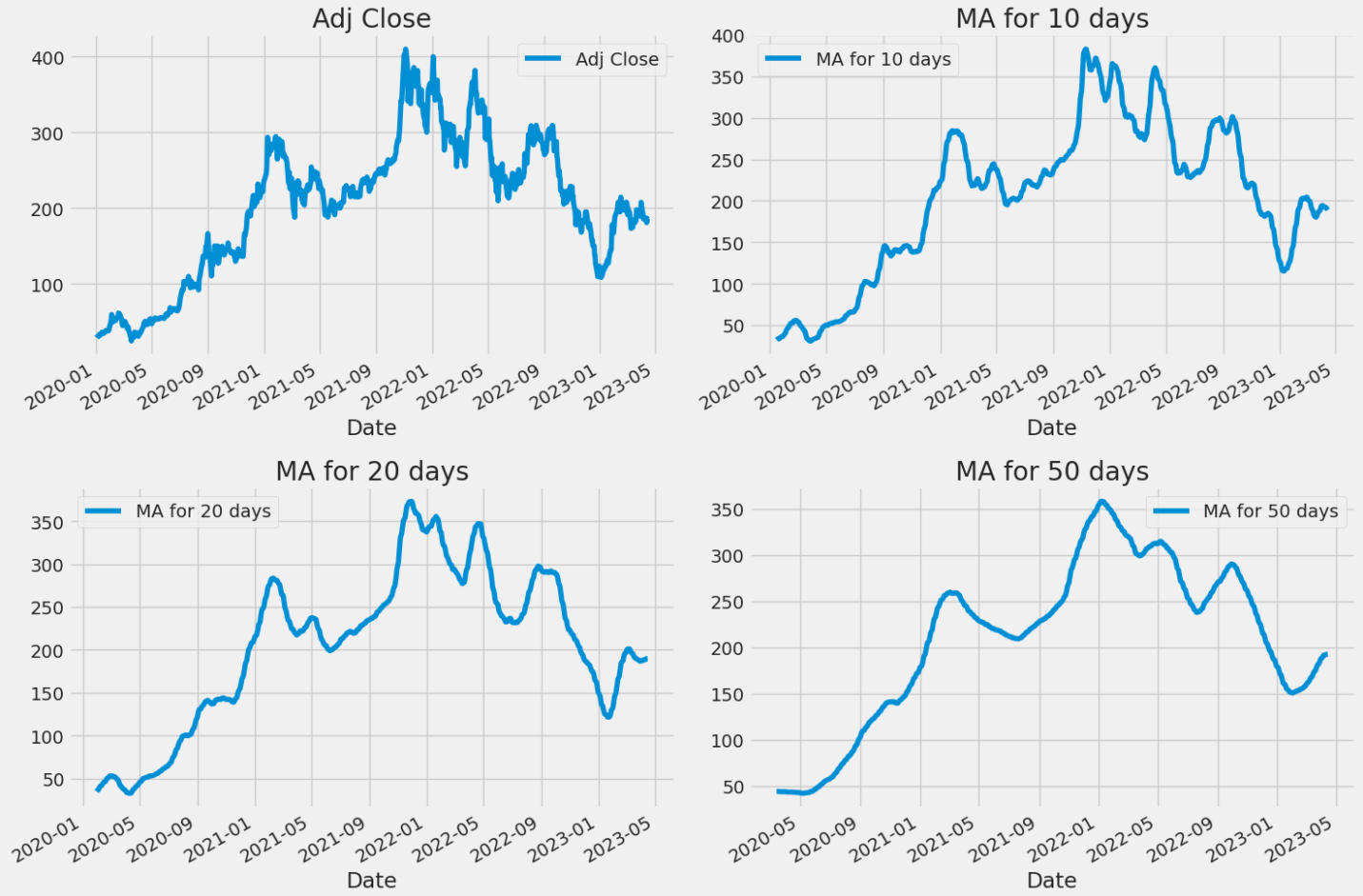
```
# Now let's plot the total volume of stock being traded each day
plt.figure(figsize=(15, 10))
df['Volume'].plot()
plt.ylabel('Volume')
plt.xlabel(None)
plt.title(f"Sales Volume of {stock}")

plt.tight_layout()
plt.show()
```



In [ ]:

```
ma_day = [10, 20, 50]
```

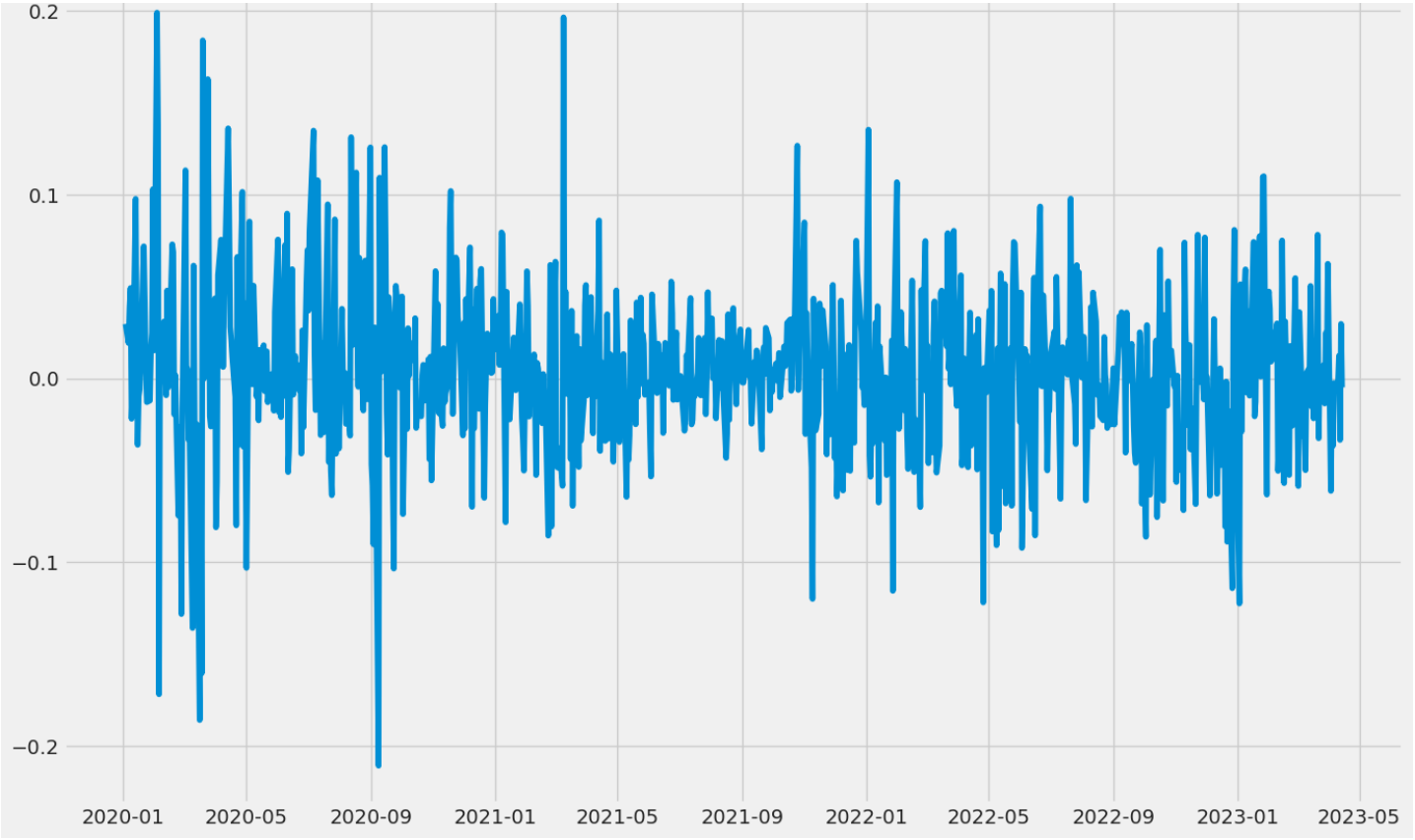
```
ma_day = [10, 20, 50]

for ma in ma_day:
    column_name = f"MA for {ma} days"
    df[column_name] = df['Adj Close'].rolling(ma).mean()

fig, axes = plt.subplots(nrows=2, ncols=2)
fig.set_figheight(10)
fig.set_figwidth(15)

df[['Adj Close']].plot(ax=axes[0,0])
axes[0,0].set_title('Adj Close')

df[['MA for 10 days']].plot(ax=axes[0,1])
axes[0,1].set_title('MA for 10 days')

df[['MA for 20 days']].plot(ax=axes[1,0])
axes[1,0].set_title('MA for 20 days')

df[['MA for 50 days']].plot(ax=axes[1,1])
axes[1,1].set_title('MA for 50 days')

fig.tight_layout()
plt.figure(figsize=(15, 10))
plt.show()
```




```
<Figure size 1500x1000 with 0 Axes>
```

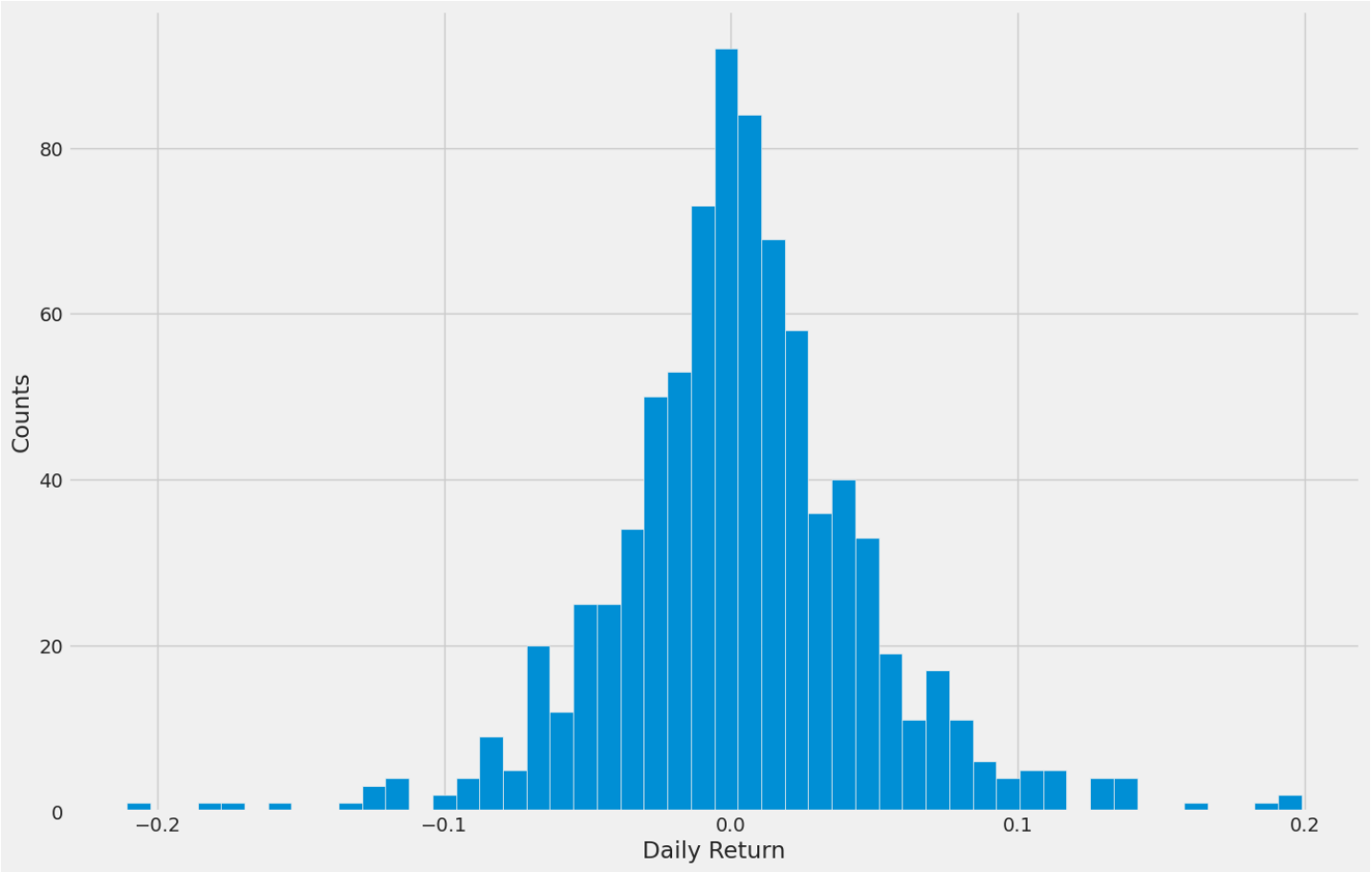
In [ ]:

```
# We'll use pct_change to find the percent change for each day
df['Daily Return'] = df['Adj Close'].pct_change()

# Then we'll plot the daily return percentage
plt.figure(figsize=(15, 10))
plt.plot(df['Daily Return'])
plt.title('Daily Returns')
plt.show()
```



In [ ]:

```
fig = plt.figure(figsize=(15, 10))
df['Daily Return'].hist(bins=50)
plt.xlabel('Daily Return')
plt.ylabel('Counts')
plt.show()
```



In [ ]:

```
# Get the stock quote
df = pdr.get_data_yahoo('AAPL', start=start, end=datetime.now())
```

```
# Show teh data
df.head()
```

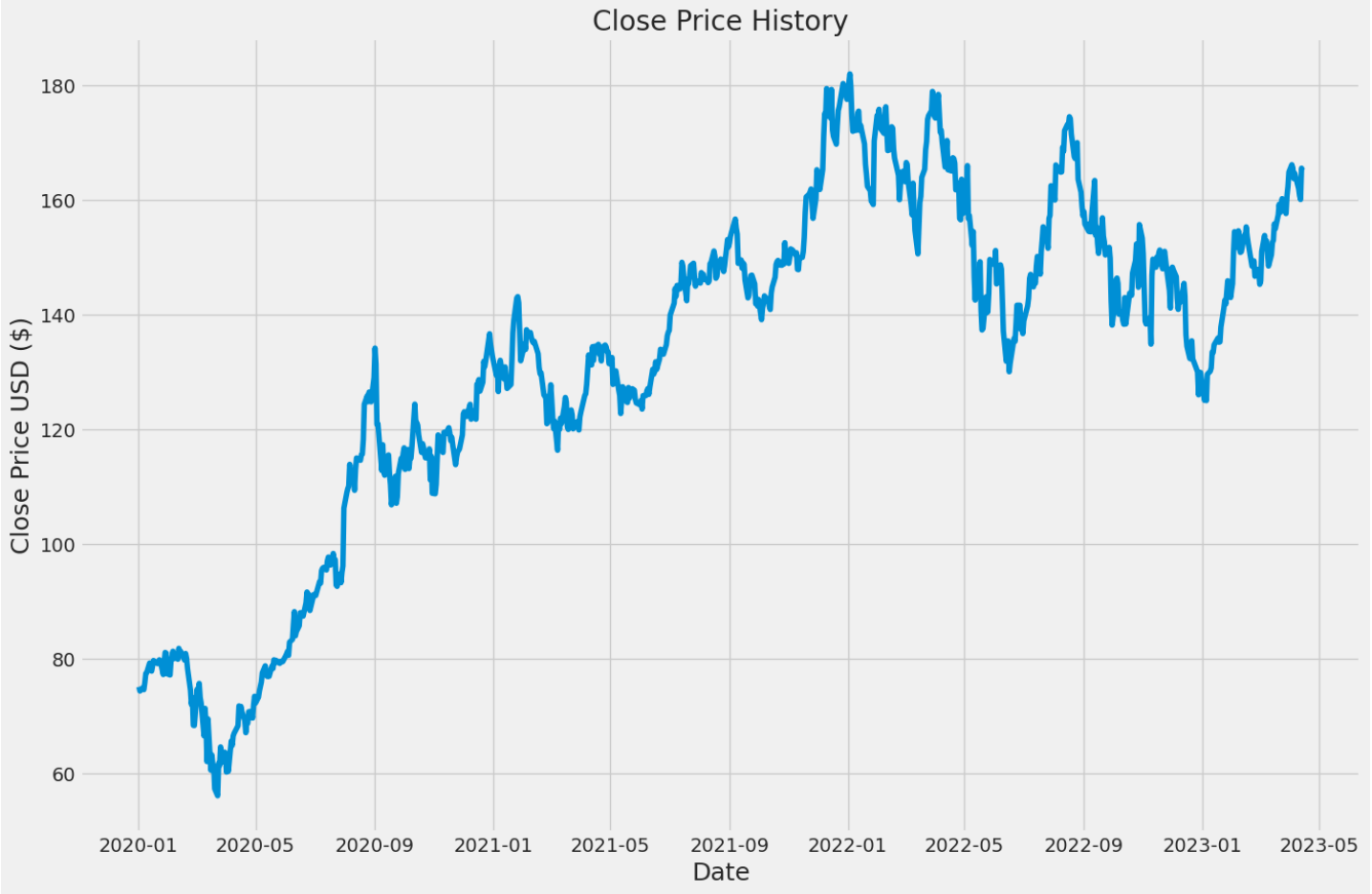
```
[**********100%**********]  1 of 1 completed
```

Out[ ]:

Date	Open	High	Low	Close	Adj Close	Volume
2020-01-02	74.059998	75.150002	73.797501	75.087502	73.449402	135480400
2020-01-03	74.287498	75.144997	74.125000	74.357498	72.735313	146322800
2020-01-06	73.447502	74.989998	73.187500	74.949997	73.314873	118387200
2020-01-07	74.959999	75.224998	74.370003	74.597504	72.970093	108872000
2020-01-08	74.290001	76.110001	74.290001	75.797501	74.143890	132079200

In [ ]:

```
plt.figure(figsize=(15,10))
plt.title('Close Price History')
plt.plot(df['Close'])
plt.xlabel('Date', fontsize=18)
plt.ylabel('Close Price USD ($)', fontsize=18)
plt.show()
```



In [ ]:

```
# Create a new dataframe with only the 'Close' column
data = df.filter(['Close'])
# Convert the dataframe to a numpy array
dataset = data.values
# Get the number of rows to train the model on
training_data_len = int(np.ceil( len(dataset) * .95 ))

training_data_len
```

Out[ ]:

786

In [ ]:

```
# Scale the data
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(dataset)

scaled_data
```

Out[ ]:

```
array([[0.15085277],
       [0.14505529],
       [0.14976075],
       [0.14696135],
       [0.15649137],
       [0.16927753],
       [0.17066729],
       [0.18383068],
       [0.17533306],
       [0.17267259],
       [0.18041574],
       [0.18734493],
       [0.1830564 ],
       [0.18529994],
       [0.18833764],
       [0.18651102],
       [0.16792745],
       [0.18528007],
       [0.19848314],
       [0.19754999],
       [0.16903929],
       [0.16735166],
       [0.18758317],
       [0.19274529],
       [0.20021045],
       [0.18992596],
       [0.19294378],
       [0.18909205],
       [0.2041615 ],
       [0.19953542],
       [0.19969428],
       [0.18788097],
       [0.19705363],
       [0.190462  ],
       [0.17606765],
       [0.14654436],
       [0.12649154],
       [0.13556496],
       [0.09758373],
       [0.09726605],
       [0.14779519],
       [0.12895351],
       [0.15559791],
       [0.13610106],
       [0.12837772],
       [0.08299089],
       [0.12105149],
       [0.1013759 ],
       [0.04737229],
       [0.1064189 ],
       [0.03542004],
       [0.05656483],
       [0.04427503],
       [0.04052257],
       [0.00966905],
       [0.        ],
       [0.04469198],
       [0.0419918 ],
```

[0.06764351],
[0.04639945],
[0.0604364],
[0.05940397],
[0.03283898],
[0.04082037],
[0.03383169],
[0.07564478],
[0.06960907],
[0.08279233],
[0.08660432],
[0.09704768],
[0.12444655],
[0.11924474],
[0.12373183],
[0.11600849],
[0.10435404],
[0.08735879],
[0.10270616],
[0.10058174],
[0.11634604],
[0.11674315],
[0.10762998],
[0.12579669],
[0.13784818],
[0.12845715],
[0.13657754],
[0.14531341],
[0.15140868],
[0.15758333],
[0.17027024],
[0.17995914],
[0.17281159],
[0.16534636],
[0.16909885],
[0.16546548],
[0.17985983],
[0.1762464],
[0.18833764],
[0.18361231],
[0.1876626],
[0.18337407],
[0.18611391],
[0.1863919],
[0.18577642],
[0.19353945],
[0.19649772],
[0.20003177],
[0.1944726],
[0.21269881],
[0.21659023],
[0.23749677],
[0.25506781],
[0.22143467],
[0.2271924],
[0.23551135],
[0.25355887],
[0.25258603],
[0.25286402],
[0.24887329],
[0.26703994],
[0.28224831],
[0.2694026],
[0.27889294],
[0.25663632],
[0.27281754],
[0.2788135],
[0.27744355],
[0.27744355],
[0.29678165],
[0.29447855],
[0.31171204],
[0.31406017],

```
[0.31496817],
[0.31629838],
[0.31278419],
[0.32533211],
[0.33063316],
[0.32108326],
[0.31953463],
[0.33565629],
[0.32487544],
[0.32703954],
[0.29187765],
[0.29005103],
[0.30748307],
[0.2951139 ],
[0.30930969],
[0.31844267],
[0.39841567],
[0.41967958],
[0.42545718],
[0.428614  ],
[0.45911013],
[0.43695282],
[0.44977866],
[0.42315407],
[0.4520222 ],
[0.46790562],
[0.46709158],
[0.46470905],
[0.47229339],
[0.47344491],
[0.49383529],
[0.54224001],
[0.55405326],
[0.54585344],
[0.55933451],
[0.5473227 ],
[0.54571451],
[0.57932771],
[0.62014809],
[0.59807015],
[0.5145234 ],
[0.51515875],
[0.45051326],
[0.48625094],
[0.45583419],
[0.44400106],
[0.4706852 ],
[0.47211471],
[0.44503346],
[0.43081779],
[0.40302181],
[0.42875299],
[0.44249211],
[0.40524554],
[0.41398141],
[0.44622473],
[0.46750851],
[0.4605992 ],
[0.47425895],
[0.48204185],
[0.45210157],
[0.47973874],
[0.45321347],
[0.46846153],
[0.46758794],
[0.48347136],
[0.54247825],
[0.51627059],
[0.51698537],
[0.51317333],
[0.49975182],
[0.47560908],
```

```
[0.48775989],
[0.4826772 ],
[0.47378246],
[0.46814386],
[0.46822329],
[0.4805329 ],
[0.43764767],
[0.47036753],
[0.4190641 ],
[0.41834931],
[0.43161201],
[0.46742907],
[0.49983126],
[0.4971311 ],
[0.47830923],
[0.47552965],
[0.50348443],
[0.50126077],
[0.50165788],
[0.50991726],
[0.50269028],
[0.49188955],
[0.49673399],
[0.48640975],
[0.45869321],
[0.46917626],
[0.47600613],
[0.48045347],
[0.49999012],
[0.52913618],
[0.5319952 ],
[0.53088336],
[0.52540357],
[0.53731613],
[0.54231938],
[0.52167095],
[0.53326584],
[0.52667427],
[0.52167095],
[0.57011536],
[0.56955944],
[0.57662756],
[0.56042653],
[0.57289495],
[0.60188225],
[0.5945759 ],
[0.60259698],
[0.64008185],
[0.62562788],
[0.61649497],
[0.60831502],
[0.58226622],
[0.59497289],
[0.55994998],
[0.59425816],
[0.60323233],
[0.57885123],
[0.57742178],
[0.59401992],
[0.57829537],
[0.56423851],
[0.56971831],
[0.60307346],
[0.6415113 ],
[0.65898315],
[0.68955866],
[0.69146471],
[0.68272878],
[0.64325849],
[0.6025176 ],
[0.61983047],
[0.62658097],
[0.61994215],
```

[0.61824215],
[0.64564102],
[0.64063771],
[0.64182904],
[0.63468143],
[0.62975761],
[0.62769281],
[0.62959874],
[0.61228587],
[0.59362281],
[0.58464876],
[0.58591934],
[0.55518497],
[0.55407314],
[0.55002285],
[0.515397],
[0.51754129],
[0.56940064],
[0.54819629],
[0.52389462],
[0.50856712],
[0.51881193],
[0.47862691],
[0.51619116],
[0.50737591],
[0.52310046],
[0.51571467],
[0.53922212],
[0.55177003],
[0.54533727],
[0.51174382],
[0.50745529],
[0.53445711],
[0.52770667],
[0.50824945],
[0.5122203],
[0.51714418],
[0.51857369],
[0.50674056],
[0.52460941],
[0.53135985],
[0.55439081],
[0.55685272],
[0.57027423],
[0.58981082],
[0.61077693],
[0.59679957],
[0.62213351],
[0.60307346],
[0.62268949],
[0.61998934],
[0.62538964],
[0.61165052],
[0.61474778],
[0.60235874],
[0.62126004],
[0.62443668],
[0.6218159],
[0.61538313],
[0.61458891],
[0.59854675],
[0.60712369],
[0.56987712],
[0.57186261],
[0.584887],
[0.58861962],
[0.56193541],
[0.55447025],
[0.52953323],
[0.54700502],
[0.56670042],
[0.55732921],
[0.54605109],

[0.54605199],
[0.54478135],
[0.56558859],
[0.5506582],
[0.56392084],
[0.56233252],
[0.56193541],
[0.54946693],
[0.544146],
[0.54152522],
[0.54771975],
[0.53564838],
[0.55431138],
[0.55439081],
[0.56106182],
[0.56415908],
[0.55605856],
[0.56590626],
[0.59076379],
[0.58409278],
[0.58814301],
[0.60116741],
[0.59060504],
[0.60521776],
[0.61855977],
[0.6163361],
[0.61403306],
[0.61165052],
[0.62491316],
[0.63722283],
[0.64222615],
[0.64468806],
[0.66605127],
[0.68241117],
[0.70266255],
[0.69210006],
[0.70695102],
[0.70210657],
[0.71116011],
[0.73903547],
[0.73371454],
[0.7171164],
[0.68582605],
[0.71521034],
[0.70925406],
[0.72037252],
[0.73434989],
[0.73776488],
[0.72013428],
[0.70591856],
[0.71116011],
[0.7129073],
[0.71020715],
[0.72481986],
[0.72156373],
[0.72243732],
[0.71513097],
[0.71473386],
[0.7108425],
[0.7129073],
[0.73697067],
[0.73863848],
[0.75468064],
[0.74729491],
[0.71687815],
[0.71957831],
[0.73141149],
[0.74348292],
[0.74276808],
[0.73276157],
[0.72624931],
[0.73466762],
[0.77056406],

```
[0.77056406],
[0.76031931],
[0.76571961],
[0.77477315],
[0.77993534],
[0.79891601],
[0.7863681 ],
[0.77810878],
[0.73760602],
[0.74221222],
[0.73085552],
[0.7380825 ],
[0.73617645],
[0.71449562],
[0.68971753],
[0.69360889],
[0.71282793],
[0.72061077],
[0.72132549],
[0.70901582],
[0.68153758],
[0.68884393],
[0.67828145],
[0.68741436],
[0.65953901],
[0.67518419],
[0.6822523 ],
[0.69249705],
[0.68939979],
[0.68868507],
[0.67836082],
[0.67359587],
[0.69622966],
[0.70480672],
[0.7183871 ],
[0.7359382 ],
[0.73990906],
[0.74165624],
[0.73538235],
[0.73498524],
[0.74038566],
[0.73665305],
[0.76619621],
[0.74419765],
[0.73752664],
[0.74594483],
[0.75761916],
[0.75341006],
[0.75595134],
[0.74928034],
[0.75221873],
[0.7292672 ],
[0.72887009],
[0.74570659],
[0.74578597],
[0.75372767],
[0.77350257],
[0.80828717],
[0.82957101],
[0.83330362],
[0.83640089],
[0.84060998],
[0.79986898],
[0.8271091 ],
[0.86729412],
[0.86308503],
[0.85506383],
[0.83981576],
[0.86745299],
[0.91399129],
[0.94496402],
[0.9408343 ],
```

[0.97966925],
[0.95020558],
[0.93900774],
[0.97847804],
[0.92256835],
[0.91367367],
[0.9026347],
[0.92836588],
[0.94941136],
[0.95449405],
[0.98665799],
[0.97839854],
[0.97911339],
[0.96974211],
[0.96473892],
[1.   ],
[0.98165467],
[0.94369332],
[0.92050355],
[0.92185362],
[0.92201249],
[0.94496402],
[0.94853777],
[0.92201249],
[0.92900123],
[0.90303181],
[0.87467986],
[0.86102011],
[0.84434259],
[0.83806858],
[0.82345586],
[0.82274114],
[0.81900852],
[0.90724091],
[0.94258148],
[0.94123141],
[0.95099968],
[0.92765103],
[0.92360081],
[0.91780339],
[0.94297859],
[0.95449405],
[0.92145651],
[0.8938194],
[0.89572545],
[0.92677744],
[0.92487151],
[0.89572545],
[0.88317754],
[0.85951128],
[0.82575903],
[0.84696337],
[0.86372038],
[0.86586455],
[0.85061649],
[0.87730063],
[0.87467986],
[0.85037825],
[0.81964388],
[0.80487229],
[0.84863106],
[0.81344935],
[0.78335021],
[0.75070979],
[0.78620923],
[0.82194692],
[0.83012687],
[0.85681101],
[0.86792948],
[0.89524897],
[0.90628794],
[0.93694294],

```
[0.942105 ],
[0.94909374],
[0.97577789],
[0.96632724],
[0.94123141],
[0.93884887],
[0.97164816],
[0.94480515],
[0.91915347],
[0.92161538],
[0.90533485],
[0.87086787],
[0.88603656],
[0.90779676],
[0.86721463],
[0.86546757],
[0.88397164],
[0.88262156],
[0.8761888 ],
[0.83941865],
[0.84807521],
[0.79978961],
[0.79796305],
[0.85411086],
[0.80653999],
[0.80900202],
[0.82107333],
[0.87301216],
[0.79955136],
[0.80360159],
[0.76214587],
[0.78160303],
[0.71798999],
[0.68669964],
[0.72283443],
[0.71036589],
[0.73975031],
[0.67288114],
[0.64532341],
[0.64722934],
[0.6910676 ],
[0.6692279 ],
[0.67049861],
[0.69638853],
[0.74292695],
[0.73657356],
[0.73554122],
[0.75539549],
[0.70909531],
[0.71513097],
[0.73554122],
[0.72958493],
[0.68733499],
[0.64357622],
[0.60188225],
[0.60887087],
[0.63007522],
[0.58742829],
[0.59934085],
[0.63356959],
[0.62943999],
[0.65262976],
[0.67955215],
[0.67955215],
[0.64603813],
[0.66025374],
[0.64032009],
[0.6578712 ],
[0.67875793],
[0.68955866],
[0.71679878],
[0.72227846],
[0.70504406],
```

```
[0.70504496],
[0.7129073 ],
[0.70996891],
[0.73363516],
[0.74713604],
[0.72251682],
[0.75372767],
[0.76992871],
[0.78827415],
[0.77826752],
[0.76921398],
[0.75849275],
[0.79971011],
[0.80415757],
[0.84513669],
[0.83719498],
[0.82528242],
[0.87388576],
[0.87134435],
[0.86769123],
[0.86387913],
[0.86427624],
[0.89858447],
[0.89262819],
[0.92129776],
[0.9299542 ],
[0.92868349],
[0.94075493],
[0.93757817],
[0.91669156],
[0.88532184],
[0.88262156],
[0.8850041 ],
[0.90485837],
[0.85395199],
[0.83616264],
[0.81654661],
[0.80312511],
[0.80900202],
[0.79192727],
[0.78176189],
[0.7931186 ],
[0.78120604],
[0.80431632],
[0.85244305],
[0.7762821 ],
[0.78795642],
[0.76460778],
[0.75134514],
[0.78136478],
[0.8005837 ],
[0.77532913],
[0.76754629],
[0.74920084],
[0.75190112],
[0.75976333],
[0.74451526],
[0.68606429],
[0.65207379],
[0.68582605],
[0.71481335],
[0.71719577],
[0.7094923 ],
[0.66708361],
[0.66970439],
[0.65826831],
[0.65318562],
[0.69011464],
[0.65350336],
[0.68550843],
[0.69615029],
[0.69702388],
[0.60200107],
```

[0.69329127],
[0.72410514],
[0.741418],
[0.76436953],
[0.7406239],
[0.70448911],
[0.79137141],
[0.77231124],
[0.75094803],
[0.70631567],
[0.65747421],
[0.65350336],
[0.65779183],
[0.66239803],
[0.62562788],
[0.72092838],
[0.74340343],
[0.73212622],
[0.74610358],
[0.73617645],
[0.75150401],
[0.75603072],
[0.72998192],
[0.74721542],
[0.75428365],
[0.73077614],
[0.6998829],
[0.67566067],
[0.73014079],
[0.73236446],
[0.72839361],
[0.71902245],
[0.68947928],
[0.67383411],
[0.68741436],
[0.683523],
[0.7020272],
[0.70981004],
[0.69186182],
[0.63857291],
[0.62276886],
[0.60577361],
[0.60521776],
[0.63023409],
[0.60466178],
[0.60172339],
[0.58719005],
[0.55550265],
[0.58385454],
[0.58639583],
[0.54779918],
[0.55804399],
[0.54740207],
[0.58393392],
[0.58814301],
[0.59274922],
[0.61466841],
[0.61403306],
[0.62475429],
[0.63412557],
[0.62832816],
[0.62880464],
[0.64945301],
[0.67518419],
[0.6864614],
[0.68114047],
[0.6978181],
[0.71346316],
[0.69019401],
[0.70043876],
[0.7094923],
[0.75229823],
[0.70150265],

```
       [0.78152365],
       [0.75952509],
       [0.78271486],
       [0.76103403],
       [0.75269521],
       [0.75380705],
       [0.77636159],
       [0.77119941],
       [0.78811529],
       [0.77524976],
       [0.76603735],
       [0.73371454],
       [0.73712953],
       [0.74102089],
       [0.7196578 ],
       [0.7292672 ],
       [0.72521697],
       [0.70853934],
       [0.71330441],
       [0.75396592],
       [0.77620272],
       [0.75849275],
       [0.76857863],
       [0.75047154],
       [0.7338734 ],
       [0.74951858],
       [0.76635496],
       [0.76953172],
       [0.79224501],
       [0.78549451],
       [0.80455456],
       [0.81948501],
       [0.80796956],
       [0.81670536],
       [0.82718847],
       [0.8115433 ],
       [0.80653999],
       [0.8313182 ],
       [0.84394548],
       [0.86411737],
       [0.87420337],
       [0.8699149 ],
       [0.85506383],
       [0.86221144],
       [0.84132471],
       [0.83155644],
       [0.82599727],
       [0.86935893],
       [0.8665794 ]])
```

3. Create the training dataset.

4. Create the scaled training dataset.

In [ ]:

```
# Create the training data set
# Create the scaled training data set
train_data = scaled_data[0:int(training_data_len), :]
# Split the data into x_train and y_train data sets
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])

# Convert the x_train and y_train to numpy arrays
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
```

5. Build the LSTM model for prediction.

In [ ]:

```
from keras.models import Sequential
from keras.layers import Dense, LSTM

# Build the LSTM model
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape= (x_train.shape[1], 1)))
model.add(LSTM(64, 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, epochs=1)
```

```
23/23 [==============================] - 12s 283ms/step - loss: 0.0436
```

Out[ ]:

```
<keras.callbacks.History at 0x7f05788da790>
```

6. Create the testing dataset.

In [ ]:

```
# Create the testing data set
# Create a new array containing scaled values from index 1543 to 2002
test_data = scaled_data[training_data_len - 60: , :]
# Create the data sets x_test and y_test
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])

# Convert the data to a numpy array
x_test = np.array(x_test)

# Reshape the data
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1 ))

# Get the models predicted price values
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)

# Get the root mean squared error (RMSE)
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
rmse
```

```
2/2 [==============================] - 2s 38ms/step
```

Out[ ]:

```
4.108398997713889
```

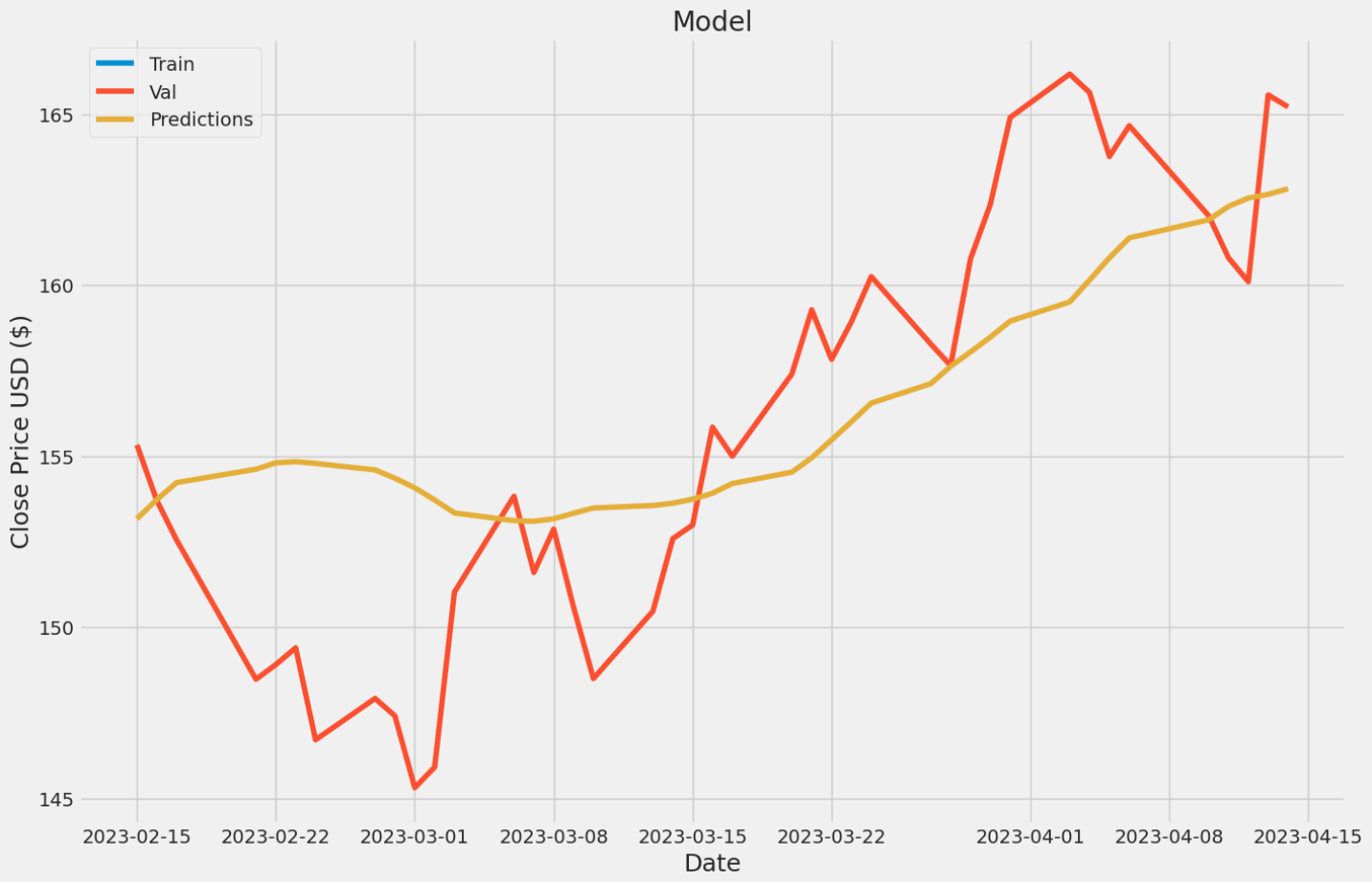
7. Plot and Visualise the data.

In [ ]:

```
# Plot the data
train = data.iloc[5300:training_data_len, :]
valid = data.iloc[training_data_len:, :]
valid['Predictions'] = predictions
```

```
# Visualize the data
plt.figure(figsize=(15, 10))

plt.title('Model')
plt.xlabel('Date', fontsize=18)
plt.ylabel('Close Price USD ($)', fontsize=18)

plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Train', 'Val', 'Predictions'], loc='best')

plt.show()
```

```
<ipython-input-60-44a072488fb9>:4: SettingWithCopyWarning:
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/user_guide/indexing.html#returning-a-view-versus-a-copy
  valid['Predictions'] = predictions
```



In [ ]:

```
!jupyter nbconvert --to html "/content/60009200040_FMC_K2_Lab5.ipynb"
```

```
[NbConvertApp] Converting notebook /content/60009200040_FMC_K2_Lab5.ipynb to html
[NbConvertApp] Writing 1569609 bytes to /content/60009200040_FMC_K2_Lab5.html
```