

Import Libraries

In [27]: 1 !pip install tensorflow

```
Requirement already satisfied: tensorflow in c:\users\high-tech\anaconda3\lib\site-packages (2.9.1)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (1.42.0)
Requirement already satisfied: numpy>=1.20 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (1.21.5)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (0.26.0)
Requirement already satisfied: tensorflow-estimator<2.10.0,>=2.9.0rc0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (2.9.0)
Requirement already satisfied: packaging in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (21.3)
Requirement already satisfied: keras-preprocessing>=1.1.1 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (4.1.1)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (1.2.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (3.19.1)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: keras<2.10.0,>=2.9.0rc0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (2.9.0)
Requirement already satisfied: six>=1.12.0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (1.16.0)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: libclang>=13.0.0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (14.0.6)
Requirement already satisfied: tensorboard<2.10,>=2.9 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (2.9.1)
Requirement already satisfied: setuptools in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (61.2.0)
Requirement already satisfied: flatbuffers<2,>=1.12 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (1.12)
Requirement already satisfied: h5py>=2.9.0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow)
```

(3.6.0)

Requirement already satisfied: wrapt>=1.11.0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (1.12.1)

Requirement already satisfied: gast<=0.4.0,>=0.2.1 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorflow) (0.4.0)

Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\high-tech\anaconda3\lib\site-packages (from astunparse>=1.6.0->tensorflow) (0.37.1)

Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (0.4.6)

Requirement already satisfied: requests<3,>=2.21.0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (2.27.1)

Requirement already satisfied: markdown>=2.6.8 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (3.3.4)

Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (0.6.1)

Requirement already satisfied: werkzeug>=1.0.1 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (2.0.3)

Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (1.8.1)

Requirement already satisfied: google-auth<3,>=1.6.3 in c:\users\high-tech\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (1.33.0)

Requirement already satisfied: rsa<5,>=3.1.4 in c:\users\high-tech\anaconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (4.7.2)

Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\high-tech\anaconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (0.2.8)

Requirement already satisfied: cachetools<5.0,>=2.0.0 in c:\users\high-tech\anaconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (4.2.2)

Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\high-tech\anaconda3\lib\site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow) (1.3.1)

Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\high-tech\anaconda3\lib\site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (0.4.8)

Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\high-tech\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (1.26.9)

Requirement already satisfied: certifi>=2017.4.17 in c:\users\high-tech\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (2021.10.8)

Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\high-tech\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (2.0.4)

Requirement already satisfied: idna<4,>=2.5 in c:\users\high-tech\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (3.3)

Requirement already satisfied: oauthlib>=3.0.0 in c:\users\high-tech\anaconda3\lib\site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow) (3.2.0)

Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\high-tech\anaconda3\lib\site-packages (from packaging->tensorflow) (3.0.4)

```
In [8]: 1 import pandas as pd
        2 import numpy as np
        3 import seaborn as sns
        4 import matplotlib.pyplot as plt
        5 import math
        6 from sklearn.preprocessing import MinMaxScaler
        7 from keras.models import Sequential
        8 from keras.layers import Dense,LSTM
        9 plt.style.use('fivethirtyeight')
```

```
In [9]: 1 df=pd.read_excel('1613615-Stock_Price_data_set.xlsx',index_col='Date')
        2 df.head()
```

Out[9]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2018-02-05	262.000000	267.899994	250.029999	254.259995	254.259995	11896100.0
2018-02-06	247.699997	266.700012	245.000000	265.720001	265.720001	12595800.0
2018-02-07	266.579987	272.450012	264.329987	264.559998	264.559998	8981500.0
2018-02-08	267.079987	267.619995	250.000000	250.100006	250.100006	9306700.0
2018-02-09	253.850006	255.800003	236.110001	249.470001	249.470001	16906900.0

```
In [10]: 1 #Get number of rows and columns in data set
         2 df.shape
```

Out[10]: (1009, 6)

```
In [11]: 1 #visualization of the closing histry
2 plt.figure(figsize=(16,8))
3 plt.title('Close price history')
4 plt.plot(df['Close'])
5 plt.xlabel('date',fontsize=18)
6 plt.ylabel('close price ')
7 plt.show()
```



```
In [12]: 1 #create a new dataframe with only the close column
          2 data=df.filter(['Close'])
          3 #convert the dataframe into numpy array
          4 dataset=data.values
          5 #get the number of rows to train the model
          6 training_data_len=math.ceil(len(dataset)*0.8)
          7 training_data_len
          8 training_data_len
```

Out[12]: 808

```
In [13]: 1 #scale the data
          2 scaler=MinMaxScaler(feature_range=(0,1))
          3 scaled_data=scaler.fit_transform(dataset)
          4 scaled_data
```

Out[13]: array([[0.04451626],
 [0.06954849],
 [0.06701469],
 ...,
 [0.4272515],
 [0.37509011],
 [0.38507243]])

In [14]:

```
1 #create the training data set
2 #create the scaled training data set
3 train_data=scaled_data[0:training_data_len,:]
4 #split the data into x_train and y_train data sets
5 x_train=[]
6 y_train=[]
7
8 for i in range (60,len(train_data)):
9     x_train.append(train_data[i-60:i,0])
10    y_train.append(train_data[i,0])
11    if i<=61:
12        print(x_train)
13        print(y_train)
14        print()
```

```
[array([0.04451626, 0.06954849, 0.06701469, 0.03542955, 0.03405342,
        0.05257641, 0.05327534, 0.0701601 , 0.10133021, 0.09750767,
        0.09757319, 0.10301218, 0.09667768, 0.11369343, 0.13167034,
        0.12391599, 0.12559796, 0.12343551, 0.14672022, 0.1771914 ,
        0.19951508, 0.19064677, 0.18156003, 0.2131015 , 0.19095254,
        0.17911361, 0.19149862, 0.19049385, 0.18472731, 0.17387127,
        0.18265218, 0.18042421, 0.15906164, 0.14647998, 0.18887749,
        0.1459339 , 0.11334393, 0.13426968, 0.10137394, 0.10875693,
        0.12026823, 0.13125532, 0.12007165, 0.12243068, 0.14021101,
        0.15244317, 0.16463161, 0.16987394, 0.16142066, 0.22319301,
        0.21982915, 0.21585376, 0.20508505, 0.18525152, 0.15976057,
        0.15700838, 0.17496343, 0.17011425, 0.17164323, 0.17347804]))]
[0.17360909661393864]
```

```
[array([0.04451626, 0.06954849, 0.06701469, 0.03542955, 0.03405342,
        0.05257641, 0.05327534, 0.0701601 , 0.10133021, 0.09750767,
        0.09757319, 0.10301218, 0.09667768, 0.11369343, 0.13167034,
        0.12391599, 0.12559796, 0.12343551, 0.14672022, 0.1771914 ,
        0.19951508, 0.19064677, 0.18156003, 0.2131015 , 0.19095254,
        0.17911361, 0.19149862, 0.19049385, 0.18472731, 0.17387127,
        0.18265218, 0.18042421, 0.15906164, 0.14647998, 0.18887749,
        0.1459339 , 0.11334393, 0.13426968, 0.10137394, 0.10875693,
        0.12026823, 0.13125532, 0.12007165, 0.12243068, 0.14021101,
        0.15244317, 0.16463161, 0.16987394, 0.16142066, 0.22319301,
        0.21982915, 0.21585376, 0.20508505, 0.18525152, 0.15976057,
        0.15700838, 0.17496343, 0.17011425, 0.17164323, 0.17347804]), array([0.06954849, 0.06701469, 0.0354295
5, 0.03405342, 0.05257641,
```

```
0.05327534, 0.0701601 , 0.10133021, 0.09750767, 0.09757319,  
0.10301218, 0.09667768, 0.11369343, 0.13167034, 0.12391599,  
0.12559796, 0.12343551, 0.14672022, 0.1771914 , 0.19951508,  
0.19064677, 0.18156003, 0.2131015 , 0.19095254, 0.17911361,  
0.19149862, 0.19049385, 0.18472731, 0.17387127, 0.18265218,  
0.18042421, 0.15906164, 0.14647998, 0.18887749, 0.1459339 ,  
0.11334393, 0.13426968, 0.10137394, 0.10875693, 0.12026823,  
0.13125532, 0.12007165, 0.12243068, 0.14021101, 0.15244317,  
0.16463161, 0.16987394, 0.16142066, 0.22319301, 0.21982915,  
0.21585376, 0.20508505, 0.18525152, 0.15976057, 0.15700838,  
0.17496343, 0.17011425, 0.17164323, 0.17347804, 0.1736091 ])]  
[0.17360909661393864, 0.16996133223364263]
```

```
In [15]: 1 #convert the x_train and y_train to numpy arrays  
2 x_train,y_train=np.array(x_train),np.array(y_train)
```

```
In [16]: 1 #reshape the data  
2 x_train=np.reshape(x_train,(x_train.shape[0],x_train.shape[1],1))  
3 x_train.shape
```

```
Out[16]: (748, 60, 1)
```

```
In [17]: 1 #build the LSTM model  
2 model=Sequential()  
3 model.add(LSTM(50,return_sequences=True,input_shape=(x_train.shape[1],1)))  
4 model.add(LSTM(50,return_sequences=False))  
5 model.add(Dense(25))  
6 model.add(Dense(1))
```

```
In [18]: 1 #compile the model  
2 model.compile(optimizer='adam',loss='mean_squared_error')
```

```
In [19]: 1 #train the model  
2 model.fit(x_train,y_train,batch_size=1,epochs=1)
```

```
748/748 [=====] - 19s 21ms/step - loss: 0.0037
```

```
Out[19]: <keras.callbacks.History at 0x181fc805880>
```



```
In [20]: 1 #create the testing data set
2 #create the new array containing scaled values from index
3 test_data=scaled_data[training_data_len-60:,:]
4 #create the data set x_test and y_test
5 x_test=[]
6 y_test=dataset[training_data_len:,:]
7 for i in range (60,len(test_data)):
8     x_test.append(test_data[i-60:i,0])
```

```
In [21]: 1 #convert data into numpy
2 x_test=np.array(x_test)
```

```
In [22]: 1 #reshape the data
2 x_test=np.reshape(x_test,(x_test.shape[0],x_test.shape[1],1))
```

```
In [23]: 1 #get the model predicted values
2 predictions=model.predict(x_test)
3 predictions=scaler.inverse_transform(predictions)
4 predictions
```

```
[599.7139 ],
[605.62585],
[612.5955 ],
[619.39215],
[624.69977],
[628.87683],
[631.10583],
[630.9749 ],
[630.07007],
[629.07996],
[628.3645 ],
[626.49774],
[624.0264 ],
[623.11426],
[623.27875],
[623.9655 ],
[624.9394 ],
[625.1488 ],
[626.4504 ],
[626.7784 ]]
```

```
In [24]: 1  # get the root mean squared error(RMSE)
          2  rmse=np.sqrt(np.mean(predictions-y_test)**2)
          3  rmse
```

```
Out[24]: 39.04711016061295
```

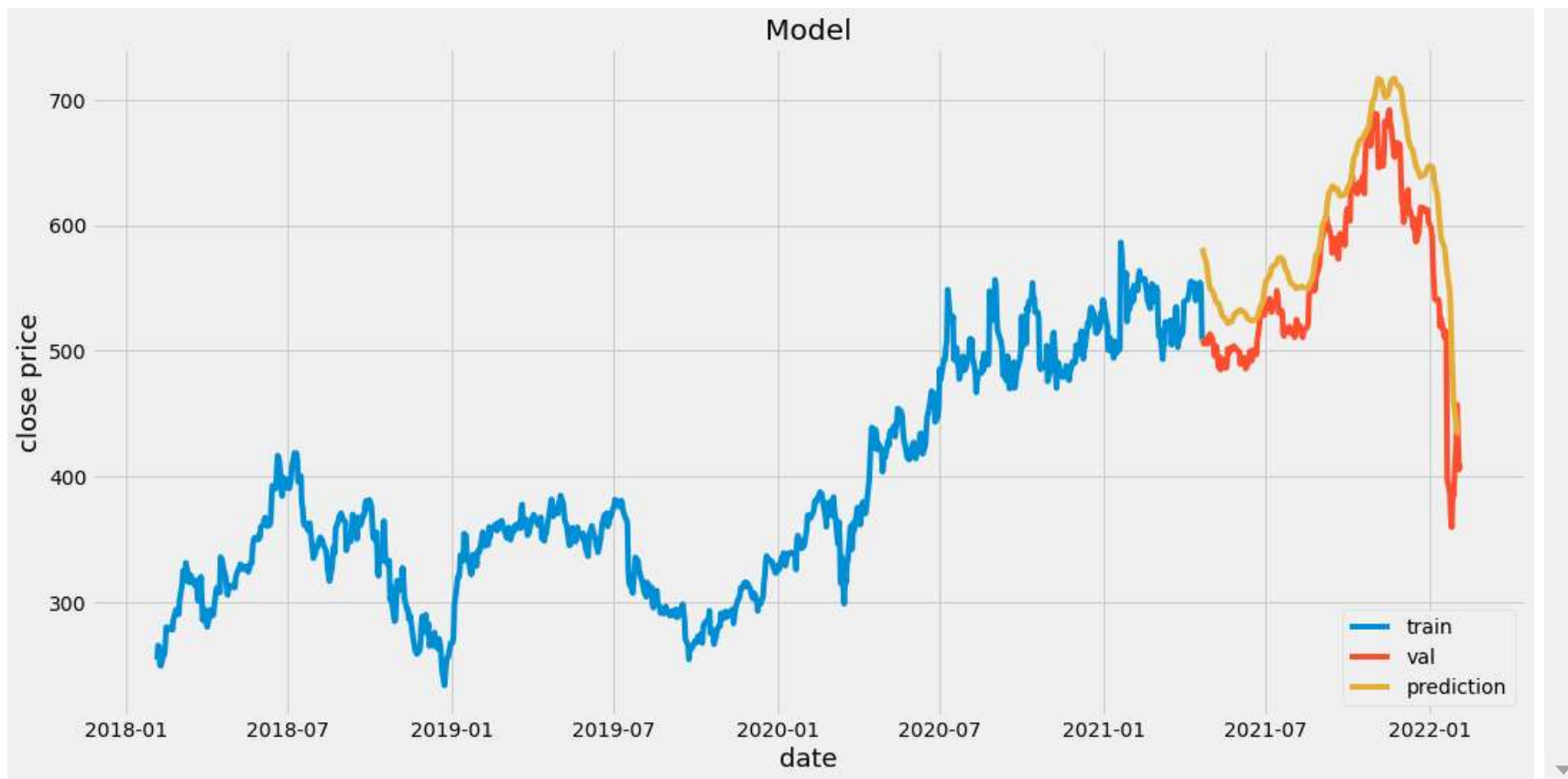
In [25]:

```
1  #plot the data
2  train=data[:training_data_len]
3  valid=data[training_data_len:]
4  valid['predictions']=predictions
5  #visualization of data
6  plt.figure(figsize=(16,8))
7  plt.title('Model')
8  plt.xlabel('date',fontsize=18)
9  plt.ylabel('close price',fontsize=18)
10 plt.plot(train['Close'])
11 plt.plot(valid[['Close','predictions']])
12 plt.legend(['train','val','prediction'],loc='lower right')
13 plt.show()
```

C:\Users\High-Tech\AppData\Local\Temp\ipykernel_3092\2617299793.py: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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
valid['predictions']=predictions
```



In [26]:

```
1 # show the valid and actual prices
2 valid
```

Out[26]:

	Close	predictions
Date		
2021-04-22	508.779999	581.925720
2021-04-23	505.549988	576.519592
2021-04-26	510.299988	570.161987
2021-04-27	505.549988	564.237671
2021-04-28	506.519989	558.561890
...
2022-01-31	427.140015	444.053070
2022-02-01	457.130005	435.527283
2022-02-02	429.480011	434.567169
2022-02-03	405.600006	435.619385
2022-02-04	410.170013	435.763000

201 rows × 2 columns

In []:

```
1
```