#Experiment No.4

Name : Sahil Tike

Class : BE-(B3)

Roll No : B212066

In [21]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
df_train = pd.read_csv('Google_Stock_Price_Train.csv')
df_test = pd.read_csv('Google_Stock_Price_Test.csv')
```

In [22]:
```
df_train.tail()
```

Out[22]:

	Date	Open	High	Low	Close	Volume
1253	12/23/2016	790.90	792.74	787.28	789.91	623,400
1254	12/27/2016	790.68	797.86	787.66	791.55	789,100
1255	12/28/2016	793.70	794.23	783.20	785.05	1,153,800
1256	12/29/2016	783.33	785.93	778.92	782.79	744,300
1257	12/30/2016	782.75	782.78	770.41	771.82	1,770,000

In [6]:
```
df_test.head()
```

Out[6]:

	Date	Open	High	Low	Close	Volume
0	1/3/2017	778.81	789.63	775.80	786.14	1,657,300
1	1/4/2017	788.36	791.34	783.16	786.90	1,073,000
2	1/5/2017	786.08	794.48	785.02	794.02	1,335,200
3	1/6/2017	795.26	807.90	792.20	806.15	1,640,200
4	1/9/2017	806.40	809.97	802.83	806.65	1,272,400

In [7]:
```
plt.plot(pd.to_datetime(df_train['Date']), df_train['Open'])
```

Out[7]:
```
[<matplotlib.lines.Line2D at 0x1b11195cfd0>]
```

In [8]:
```
    df_train['Open']
```

Out[8]:
```
0       325.25
1       331.27
2       329.83
3       328.34
4       322.04
         ...
1253    790.90
1254    790.68
1255    793.70
1256    783.33
1257    782.75
Name: Open, Length: 1258, dtype: float64
```

In [9]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(df_train[['Open']])
X_train = []
y_train = []
for i in range(1198):
    row = X_scaled[i:60+i,0]
    y = X_scaled[i+60,0]
    X_train.append(row)
    y_train.append(y)
X_train = np.array(X_train)
y_train = np.array(y_train)
X_train.shape
```

Out[9]:
```
(1198, 60)
```

In [10]:
```
X_train[:2]
```

Out[10]:
```
array([[0.08581368, 0.09701243, 0.09433366, 0.09156187, 0.07984225,
        0.0643277 , 0.0585423 , 0.06568569, 0.06109085, 0.06639259,
        0.0614257 , 0.07474514, 0.02797827, 0.02379269, 0.02409033,
        0.0159238 , 0.01078949, 0.00967334, 0.01642607, 0.02100231,
        0.02280676, 0.02273235, 0.02810849, 0.03212665, 0.0433812 ,
        0.04475779, 0.04790163, 0.0440695 , 0.04648783, 0.04745517,
        0.04873875, 0.03936305, 0.04137213, 0.04034898, 0.04784582,
        0.04325099, 0.04356723, 0.04286033, 0.04602277, 0.05398467,
        0.05738894, 0.05714711, 0.05569611, 0.04421832, 0.04514845,
        0.04605997, 0.04412531, 0.03675869, 0.04486941, 0.05065481,
        0.05214302, 0.05612397, 0.05818885, 0.06540665, 0.06882953,
        0.07243843, 0.07993526, 0.07846566, 0.08034452, 0.08497656],
       [0.09701243, 0.09433366, 0.09156187, 0.07984225, 0.0643277 ,
        0.0585423 , 0.06568569, 0.06109085, 0.06639259, 0.0614257 ,
        0.07474514, 0.02797827, 0.02379269, 0.02409033, 0.0159238 ,
        0.01078949, 0.00967334, 0.01642607, 0.02100231, 0.02280676,
        0.02273235, 0.02810849, 0.03212665, 0.0433812 , 0.04475779,
        0.04790163, 0.0440695 , 0.04648783, 0.04745517, 0.04873875,
        0.03936305, 0.04137213, 0.04034898, 0.04784582, 0.04325099,
        0.04356723, 0.04286033, 0.04602277, 0.05398467, 0.05738894,
        0.05714711, 0.05569611, 0.04421832, 0.04514845, 0.04605997,
        0.04412531, 0.03675869, 0.04486941, 0.05065481, 0.05214302,
        0.05612397, 0.05818885, 0.06540665, 0.06882953, 0.07243843,
        0.07993526, 0.07846566, 0.08034452, 0.08497656, 0.08627874]])
```

In [11]:
```
from tensorflow.keras.layers import LSTM, Dense
from tensorflow.keras.models import Sequential
X_train = X_train.reshape(1198,60,1)
model = Sequential()
model.add(LSTM(70, return_sequences=True))
model.add(LSTM(70, return_sequences=True))
model.add(LSTM(70, return_sequences=True))
model.add(LSTM(70, return_sequences=False))
model.add(Dense(1))

model.compile(loss='mean_squared_error')
model.fit(X_train,y_train, epochs = 30, batch_size = 32)
```

```
Epoch 1/30
38/38 [==============================] - 11s 67ms/step - loss: 0.0345
Epoch 2/30
38/38 [==============================] - 2s 57ms/step - loss: 0.0104
Epoch 3/30
38/38 [==============================] - 2s 57ms/step - loss: 0.0094
Epoch 4/30
38/38 [==============================] - 2s 58ms/step - loss: 0.0074
Epoch 5/30
38/38 [==============================] - 2s 59ms/step - loss: 0.0072
Epoch 6/30
38/38 [==============================] - 2s 60ms/step - loss: 0.0053
Epoch 7/30
38/38 [==============================] - 2s 58ms/step - loss: 0.0051
Epoch 8/30
38/38 [==============================] - 2s 56ms/step - loss: 0.0050
Epoch 9/30
38/38 [==============================] - 2s 59ms/step - loss: 0.0042
Epoch 10/30
38/38 [==============================] - 2s 57ms/step - loss: 0.0041
Epoch 11/30
38/38 [==============================] - 2s 61ms/step - loss: 0.0042
Epoch 12/30
38/38 [==============================] - 2s 56ms/step - loss: 0.0033
Epoch 13/30
38/38 [==============================] - 2s 60ms/step - loss: 0.0040
Epoch 14/30
38/38 [==============================] - 2s 56ms/step - loss: 0.0035
Epoch 15/30
38/38 [==============================] - 2s 54ms/step - loss: 0.0037
Epoch 16/30
38/38 [==============================] - 2s 57ms/step - loss: 0.0026
Epoch 17/30
38/38 [==============================] - 2s 57ms/step - loss: 0.0027
Epoch 18/30
38/38 [==============================] - 2s 55ms/step - loss: 0.0029
Epoch 19/30
38/38 [==============================] - 2s 55ms/step - loss: 0.0023
Epoch 20/30
38/38 [==============================] - 2s 55ms/step - loss: 0.0028
Epoch 21/30
38/38 [==============================] - 2s 59ms/step - loss: 0.0028
Epoch 22/30
38/38 [==============================] - 2s 59ms/step - loss: 0.0028
Epoch 23/30
38/38 [==============================] - 2s 56ms/step - loss: 0.0020
Epoch 24/30
38/38 [==============================] - 2s 58ms/step - loss: 0.0024
Epoch 25/30
38/38 [==============================] - 2s 58ms/step - loss: 0.0023
Epoch 26/30
38/38 [==============================] - 2s 59ms/step - loss: 0.0023
Epoch 27/30
38/38 [==============================] - 2s 58ms/step - loss: 0.0023
Epoch 28/30
38/38 [==============================] - 2s 57ms/step - loss: 0.0021
Epoch 29/30
```

```
38/38 [==============================] - 2s 62ms/step - loss: 0.0022
Epoch 30/30
38/38 [==============================] - 2s 59ms/step - loss: 0.0021
```

Out[11]:
```
<keras.callbacks.History at 0x1b1221fcb80>
```

In [12]:
```
 X_test_scaled = scaler.transform(df_test[['Open']])
X = np.vstack([X_scaled[-60:],X_test_scaled])
X_test = []
for i in range(20):
    row = X[i:60+i,0]
    X_test.append(row)
X_test = np.array(X_test)
X_test.shape
```

Out[12]:
```
(20, 60)
```

In [13]:
```
    X_test = X_test.reshape(20,60,1)
yp = model.predict(X_test)
```

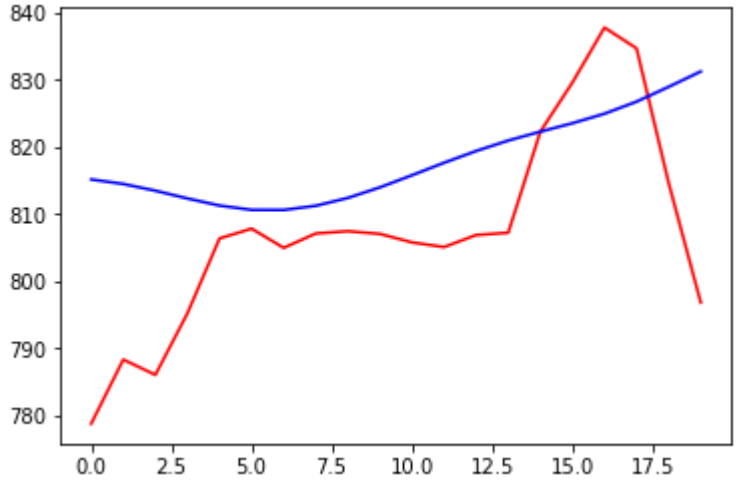
```
1/1 [==============================] - 1s 1s/step
```

In [14]:
```
yp
```

Out[14]:
```
array([[0.9972302 ],
       [0.9959745 ],
       [0.99410826],
       [0.9919541 ],
       [0.989986  ],
       [0.98879987],
       [0.9887742 ],
       [0.98992586],
       [0.9921009 ],
       [0.9950321 ],
       [0.9983955 ],
       [1.0018523 ],
       [1.0051098 ],
       [1.008005  ],
       [1.0104644 ],
       [1.0128046 ],
       [1.0154438 ],
       [1.0187842 ],
       [1.0228859 ],
       [1.0271397 ]], dtype=float32)
```

In [15]:
```
plt.plot(df_test['Open'], c='red')
plt.plot(scaler.inverse_transform(yp), c='blue')
```

Out[15]:
```
[<matplotlib.lines.Line2D at 0x1b12f1e0dc0>]
```



In [16]:
```
scaler.inverse_transform(yp)
```

Out[16]:
```
array([[815.19104],
       [814.51605],
       [813.5128 ],
       [812.3548 ],
       [811.2969 ],
       [810.65924],
       [810.64545],
       [811.2645 ],
       [812.4338 ],
       [814.0094 ],
       [815.8175 ],
       [817.67566],
       [819.4268 ],
       [820.98315],
       [822.30524],
       [823.56323],
       [824.98193],
       [826.7776 ],
       [828.98254],
       [831.26917]], dtype=float32)
```

In [17]:
```
from sklearn.metrics import mean_squared_error, r2_score
```

In [18]:
```
yp[:,0]
```

Out[18]:
```
array([0.9972302 , 0.9959745 , 0.99410826, 0.9919541 , 0.989986  ,
       0.98879987, 0.9887742 , 0.98992586, 0.9921009 , 0.9950321 ,
       0.9983955 , 1.0018523 , 1.0051098 , 1.008005  , 1.0104644 ,
       1.0128046 , 1.0154438 , 1.0187842 , 1.0228859 , 1.0271397 ],
      dtype=float32)
```

In [19]:
```
mean_squared_error(X_test_scaled,yp[:,0])
```

Out[19]:
```
0.0009451006198222933
```

In [20]:
```
r2_score(X_test_scaled,yp[:,0])
```

Out[20]:
```
-0.25658817114834975
```

In [ ]:
```

```