Sprint 2-Model building

In [17]:
```
training_size=int(len(data_final)*0.65)
test_size=len(data_final)-training_size
train_data,test_data=data_final[0:training_size,:],data_final[training_size:len(data_final),:1]
```

In [18]:
```
training_size,test_size
```

Out[18]: (5340, 2876)

In [19]:
```
train_data.shape
```

Out[19]: (5340, 1)

In [20]:
```
test_data.shape
```

Out[20]: (2876, 1)

In [21]:
```
def create_dataset(dataset,timestep=1):
    dataX,dataY=[],[]
    for i in range(len(dataset)-time_step-1):
        a=dataset[i:(i+time_step),0]
        dataX.append(a)
        dataY.append(dataset[i+time_step,0])
    return np.array(dataX),np.array(dataY)
```

In [22]:
```
time_step=10
x_train, y_train= create_dataset(train_data,time_step)
x_test, y_test = create_dataset(test_data, time_step)
```

In [23]:
```
print(x_test.shape),print(y_test.shape)
```

```
(2865, 10)
(2865,)
```

Out[23]: (None, None)

In [24]:
```
print(x_train.shape),print(y_train.shape)
```

```
(5329, 10)
(5329,)
```

Out[24]: (None, None)

In [25]:
```
x_train
```

Out[25]:
```
array([[0.11335703, 0.11661484, 0.12053902, ..., 0.10980305, 0.1089886 ,
        0.11054346],
       [0.11661484, 0.12053902, 0.11550422, ..., 0.1089886 , 0.11054346,
        0.10165852],
       [0.12053902, 0.11550422, 0.1156523 , ..., 0.11054346, 0.10165852,
        0.09906708],
       ...,
       [0.36731823, 0.35176958, 0.36080261, ..., 0.36391234, 0.37042796,
        0.37042796],
       [0.35176958, 0.36080261, 0.35354657, ..., 0.37042796, 0.37042796,
        0.37879461],
       [0.36080261, 0.35354657, 0.35295424, ..., 0.37042796, 0.37879461,
        0.37916482]])
```

In [1]:
```
x_train= x_train.reshape(x_train.shape[0],x_train.shape[1],1)
x_test= x_test.reshape(x_test.shape[0],x_test.shape[1],1)
```

In [28]:
```
#importing libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
```

In [ ]:
```
conda install tensorflow
```

In [29]:
```
model = Sequential()
```

In [30]:
```
model.add(LSTM(50,return_sequences=True,input_shape=(10,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
```

In [31]:
```
model.add(Dense(1))
```

In [32]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 10, 50)            10400

 lstm_1 (LSTM)               (None, 10, 50)            20200

 lstm_2 (LSTM)               (None, 50)                20200

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

In [33]:
```
model.compile(loss='mean_squared_error',optimizer='adam')
```

In [35]:
```
model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=50,batch_size=64,verbose=1)
```

```
Epoch 1/50
84/84 [==============================] - 6s 69ms/step - loss: 8.2759e-05 - val_loss: 4.0892e-04
Epoch 2/50
84/84 [==============================] - 4s 47ms/step - loss: 7.5317e-05 - val_loss: 4.8240e-04
Epoch 3/50
84/84 [==============================] - 4s 49ms/step - loss: 7.9198e-05 - val_loss: 0.0010
Epoch 4/50
84/84 [==============================] - 6s 67ms/step - loss: 8.3020e-05 - val_loss: 3.3202e-04
Epoch 5/50
84/84 [==============================] - 4s 48ms/step - loss: 6.4561e-05 - val_loss: 3.6391e-04
Epoch 6/50
84/84 [==============================] - 4s 47ms/step - loss: 6.7649e-05 - val_loss: 3.4241e-04
Epoch 7/50
84/84 [==============================] - 5s 65ms/step - loss: 6.1172e-05 - val_loss: 2.9344e-04
Epoch 8/50
84/84 [==============================] - 6s 68ms/step - loss: 6.5263e-05 - val_loss: 5.5413e-04
Epoch 9/50
84/84 [==============================] - 4s 48ms/step - loss: 5.9849e-05 - val_loss: 3.9287e-04
Epoch 10/50
84/84 [==============================] - 4s 51ms/step - loss: 5.5862e-05 - val_loss: 2.5168e-04
Epoch 11/50
84/84 [==============================] - 6s 74ms/step - loss: 5.9357e-05 - val_loss: 3.5915e-04
Epoch 12/50
84/84 [==============================] - 4s 45ms/step - loss: 5.3281e-05 - val_loss: 2.4452e-04
Epoch 13/50
84/84 [==============================] - 4s 50ms/step - loss: 5.1978e-05 - val_loss: 3.8276e-04
Epoch 14/50
84/84 [==============================] - 6s 68ms/step - loss: 4.9238e-05 - val_loss: 2.1604e-04
Epoch 15/50
84/84 [==============================] - 5s 56ms/step - loss: 4.8571e-05 - val_loss: 2.0100e-04
Epoch 16/50
84/84 [==============================] - 4s 49ms/step - loss: 4.7821e-05 - val_loss: 1.9581e-04
Epoch 17/50
84/84 [==============================] - 5s 55ms/step - loss: 4.2514e-05 - val_loss: 1.9930e-04
Epoch 18/50
84/84 [==============================] - 6s 76ms/step - loss: 4.3120e-05 - val_loss: 2.4934e-04
Epoch 19/50
84/84 [==============================] - 5s 59ms/step - loss: 4.7707e-05 - val_loss: 3.9497e-04
Epoch 20/50
84/84 [==============================] - 4s 53ms/step - loss: 4.5128e-05 - val_loss: 1.8131e-04
```

```
84/84 [==============================] - 4s 53ms/step - loss: 4.5128e-05 - val_loss: 1.8131e-04
Epoch 21/50
84/84 [==============================] - 6s 76ms/step - loss: 3.8253e-05 - val_loss: 2.7298e-04
Epoch 22/50
84/84 [==============================] - 5s 55ms/step - loss: 3.7996e-05 - val_loss: 2.5495e-04
Epoch 23/50
84/84 [==============================] - 5s 55ms/step - loss: 3.7103e-05 - val_loss: 2.3953e-04
Epoch 24/50
84/84 [==============================] - 6s 74ms/step - loss: 3.9661e-05 - val_loss: 1.7659e-04
Epoch 25/50
84/84 [==============================] - 5s 54ms/step - loss: 3.6607e-05 - val_loss: 2.7119e-04
Epoch 26/50
84/84 [==============================] - 5s 57ms/step - loss: 3.7889e-05 - val_loss: 3.6993e-04
Epoch 27/50
84/84 [==============================] - 6s 71ms/step - loss: 3.6533e-05 - val_loss: 1.9075e-04
Epoch 28/50
84/84 [==============================] - 5s 56ms/step - loss: 3.5179e-05 - val_loss: 2.3056e-04
Epoch 29/50
84/84 [==============================] - 5s 55ms/step - loss: 3.3028e-05 - val_loss: 1.8503e-04
Epoch 30/50
84/84 [==============================] - 6s 72ms/step - loss: 3.7884e-05 - val_loss: 2.4888e-04
Epoch 31/50
84/84 [==============================] - 5s 56ms/step - loss: 3.5201e-05 - val_loss: 1.8560e-04
Epoch 32/50
84/84 [==============================] - 5s 57ms/step - loss: 3.3765e-05 - val_loss: 3.4012e-04
Epoch 33/50
84/84 [==============================] - 6s 74ms/step - loss: 3.6406e-05 - val_loss: 2.0081e-04
Epoch 34/50
84/84 [==============================] - 5s 57ms/step - loss: 3.3040e-05 - val_loss: 1.8671e-04
Epoch 35/50
84/84 [==============================] - 5s 56ms/step - loss: 3.1530e-05 - val_loss: 2.9813e-04
Epoch 36/50
84/84 [==============================] - 6s 76ms/step - loss: 3.1774e-05 - val_loss: 2.0157e-04
Epoch 37/50
84/84 [==============================] - 4s 50ms/step - loss: 3.5654e-05 - val_loss: 3.4396e-04
Epoch 38/50
84/84 [==============================] - 5s 57ms/step - loss: 3.3372e-05 - val_loss: 1.9203e-04
Epoch 39/50
84/84 [==============================] - 7s 78ms/step - loss: 3.3287e-05 - val_loss: 1.9372e-04
Epoch 40/50

84/84 [==============================] - 5s 59ms/step - loss: 3.0756e-05 - val_loss: 4.9158e-04
Epoch 41/50
84/84 [==============================] - 5s 59ms/step - loss: 3.4120e-05 - val_loss: 3.6207e-04
Epoch 42/50
84/84 [==============================] - 6s 76ms/step - loss: 3.2224e-05 - val_loss: 1.8559e-04
Epoch 43/50
84/84 [==============================] - 6s 68ms/step - loss: 3.0016e-05 - val_loss: 4.4297e-04
Epoch 44/50
84/84 [==============================] - 5s 54ms/step - loss: 3.2877e-05 - val_loss: 3.0166e-04
Epoch 45/50
84/84 [==============================] - 6s 74ms/step - loss: 3.3182e-05 - val_loss: 2.4356e-04
Epoch 46/50
84/84 [==============================] - 4s 52ms/step - loss: 3.4722e-05 - val_loss: 2.8399e-04
Epoch 47/50
84/84 [==============================] - 5s 54ms/step - loss: 3.0769e-05 - val_loss: 1.8908e-04
Epoch 48/50
84/84 [==============================] - 6s 76ms/step - loss: 3.0809e-05 - val_loss: 4.1680e-04
Epoch 49/50
84/84 [==============================] - 4s 49ms/step - loss: 3.2482e-05 - val_loss: 1.8645e-04
Epoch 50/50
84/84 [==============================] - 6s 66ms/step - loss: 3.2913e-05 - val_loss: 1.8074e-04
```

Out[35]:

In [37]:
```
from tensorflow.keras.models import load_model
```

model.save("data_final.h5")

In [38]:
```
len(test_data)
```

Out[38]: 2876

In [39]:
```
x_input=test_data[2866:].reshape(1,-1)
x_input.shape
```

Out[39]: (1, 10)

In [41]:
```
temp_input=list(x_input)
temp_input=temp_input[0].tolist()
```

In [42]:
```
temp_input
```

Out[42]:
```
[0.44172960165852215,
 0.48111950244335855,
 0.49726047682511476,
 0.4679401747371539,
 0.4729749740855915,
 0.47119798608026064,
 0.47341922108692425,
 0.4649785280616022,
 0.4703835332444839,
 0.47149415074781587]
```

In [45]:
```
lst_output=[]
n_steps=10
i=0
while(i<10):
    if(len(temp_input)>10):
        x_input=np.array(temp_input[1:])
        print("{} day input {}".format(i,x_input))
        x_input=x_input.reshape(1,-1)
        x_input=x_input.reshape((1,n_steps,1))
        yhat=model.predict(x_input,verbose=0)
        print("{} day input {}".format(i,yhat))
        temp_input.extend(yhat[0].tolist())
        temp_input=temp_input[1:]
        lst_output.extend(yhat.tolist())
        i=i+1
    else:
        x_input=x_input.reshape((1,n_steps,1))
        yhat=model.predict(x_input,verbose=0)
        print(yhat[0])
        temp_input.extend(yhat[0].tolist())
        print(len(temp_input))
        lst_output.extend(yhat.tolist())
        i=i+1
```

```
0 day input [0.4811195  0.49726048 0.46794017 0.47297497 0.47119799 0.47341922
 0.46497853 0.47038353 0.47149415 0.47458544]
0 day input [[0.4780081]]
1 day input [0.49726048 0.46794017 0.47297497 0.47119799 0.47341922 0.46497853
 0.47038353 0.47149415 0.47458544 0.47800809]
1 day input [[0.48115236]]
2 day input [0.46794017 0.47297497 0.47119799 0.47341922 0.46497853 0.47038353
 0.47149415 0.47458544 0.47800809 0.48115236]
2 day input [[0.48385763]]
3 day input [0.47297497 0.47119799 0.47341922 0.46497853 0.47038353 0.47149415
 0.47458544 0.47800809 0.48115236 0.48385763]
3 day input [[0.48631832]]
4 day input [0.47119799 0.47341922 0.46497853 0.47038353 0.47149415 0.47458544
 0.47800809 0.48115236 0.48385763 0.48631832]
4 day input [[0.48859924]]
5 day input [0.47341922 0.46497853 0.47038353 0.47149415 0.47458544 0.47800809
 0.48115236 0.48385763 0.48631832 0.48859924]
5 day input [[0.49077198]]
6 day input [0.46497853 0.47038353 0.47149415 0.47458544 0.47800809 0.48115236
 0.48385763 0.48631832 0.48859924 0.49077198]
6 day input [[0.4928743]]
7 day input [0.47038353 0.47149415 0.47458544 0.47800809 0.48115236 0.48385763
 0.48631832 0.48859924 0.49077198 0.49287429]
7 day input [[0.4949832]]
8 day input [0.47149415 0.47458544 0.47800809 0.48115236 0.48385763 0.48631832
 0.48859924 0.49077198 0.49287429 0.4949832 ]
8 day input [[0.49709532]]
9 day input [0.47458544 0.47800809 0.48115236 0.48385763 0.48631832 0.48859924
 0.49077198 0.49287429 0.4949832  0.49709532]
9 day input [[0.4992116]]
```

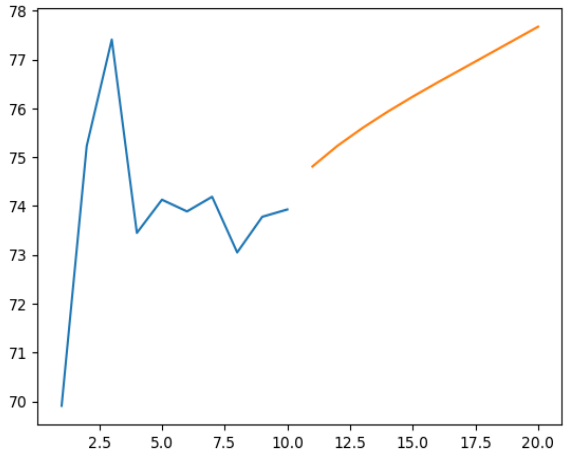
In [46]:
```
day_new=np.arange(1,11)
day_pred=np.arange(11,21)
```

In [47]:
```
len(data_final)
```

Out[47]: 8216

In [48]:
```
plt.plot(day_new,scaler.inverse_transform(data_final[8206:]))
plt.plot(day_pred,scaler.inverse_transform(lst_output))
```

Out[48]: []



In [49]:
```
df3=data_final.tolist()
df3.extend(lst_output)
plt.plot(df3[8100:])
```

Out[49]: []