Untitled14

May 23, 2021

[93]:
```
import math
import numpy as np
import pandas as pd
pd.options.mode.chained_assignment = None

from keras import optimizers
from keras.models import Sequential
from keras.layers import Dense , BatchNormalization , Dropout , Activation
from keras.layers import LSTM , GRU, Bidirectional
from keras.callbacks import ReduceLROnPlateau

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error

%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
```

[94]:
```
df= pd.read_csv('cowin_vaccine_data_statewise1.csv')
df
```

[94]:

	Updated On	...	Total Doses Administered
0	16-01-2021	...	23
1	17-01-2021	...	23
2	18-01-2021	...	42
3	19-01-2021	...	89
4	20-01-2021	...	124
...	...	...	...
4570	19-05-2021	...	12869283
4571	20-05-2021	...	12933777
4572	21-05-2021	...	0
4573	22-05-2021	...	0
4574	23-05-2021	...	0

[4575 rows x 4 columns]

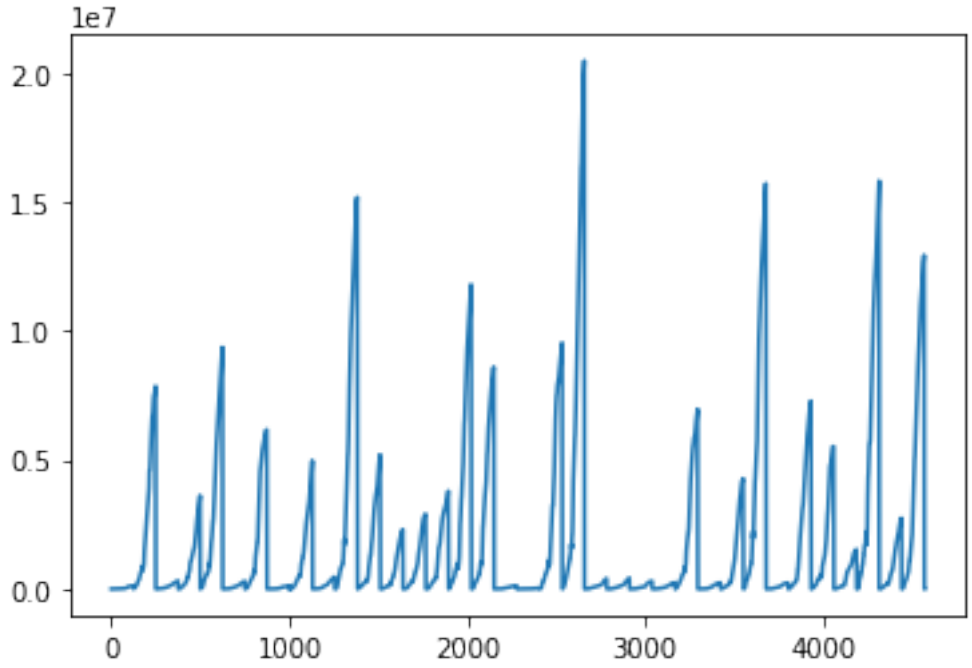
[95]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 4575 entries, 0 to 4574
Data columns (total 4 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Updated On                     4575 non-null   object
 1   State                          4575 non-null   object
 2   Total Individuals Vaccinated   4499 non-null   float64
 3   Total Doses Administered       4575 non-null   int64
dtypes: float64(1), int64(1), object(2)
memory usage: 143.1+ KB
```

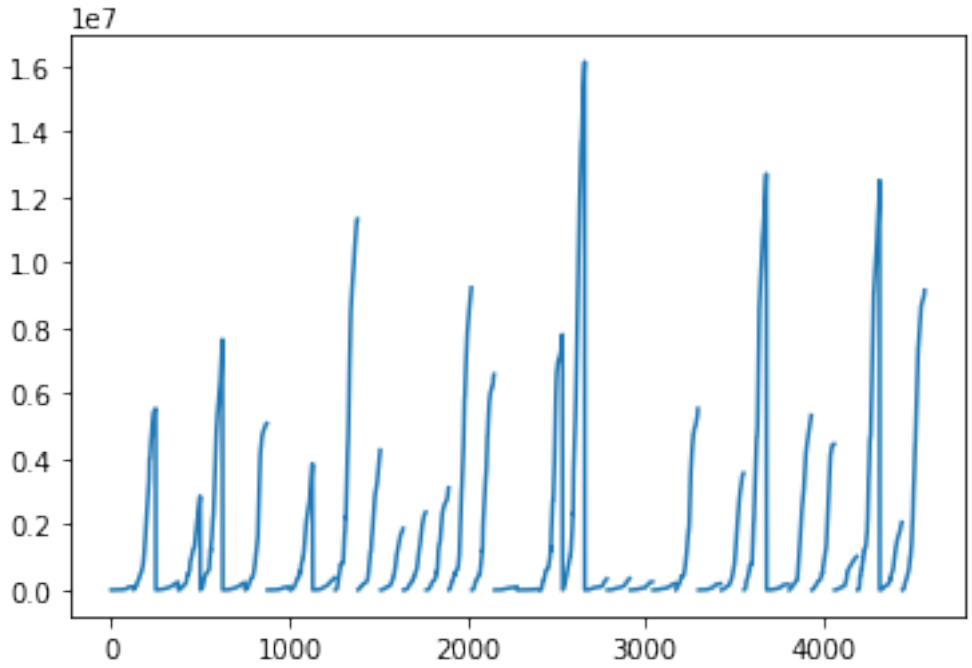
[96]:
```
df['Total Doses Administered'].plot()
```

[96]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd87f6df250>
```



[97]:
```
df['Total Individuals Vaccinated'].plot()
```

[97]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd87f769f50>
```



[98]:
```
df1=df.loc[df['State'] == "Delhi"]
df1
```

[98]:

	Updated On	State	Total Individuals Vaccinated	Total Doses Administered
1006	16-01-2021	Delhi	892.0	892
1007	17-01-2021	Delhi	1005.0	1005
1008	18-01-2021	Delhi	1366.0	1366
1009	19-01-2021	Delhi	3041.0	3041
1010	20-01-2021	Delhi	3041.0	3041
...	...	...	...	...
1126	16-05-2021	Delhi	3544850.0	4603220
1127	17-05-2021	Delhi	3639716.0	4717683
1128	18-05-2021	Delhi	3729586.0	4824666
1129	19-05-2021	Delhi	3791772.0	4893993
1130	20-05-2021	Delhi	3849484.0	4972927

[125 rows x 4 columns]

[99]:
```
# df1.drop(['Total Sessions Conducted','First Dose Administered','Second Dose␣
↪Administered','Male(Individuals Vaccinated)','Female(Individuals␣
↪Vaccinated)','Transgender(Individuals Vaccinated)'], axis=1, inplace=True)
```

[100]:
```
# df1.drop(['Total Covaxin Administered','Total CoviShield␣
↪Administered','AEFI','18-30 years (Age)','30-45 years (Age)','45-60 years␣
↪(Age)','60+ years (Age)'], axis=1, inplace=True)
```

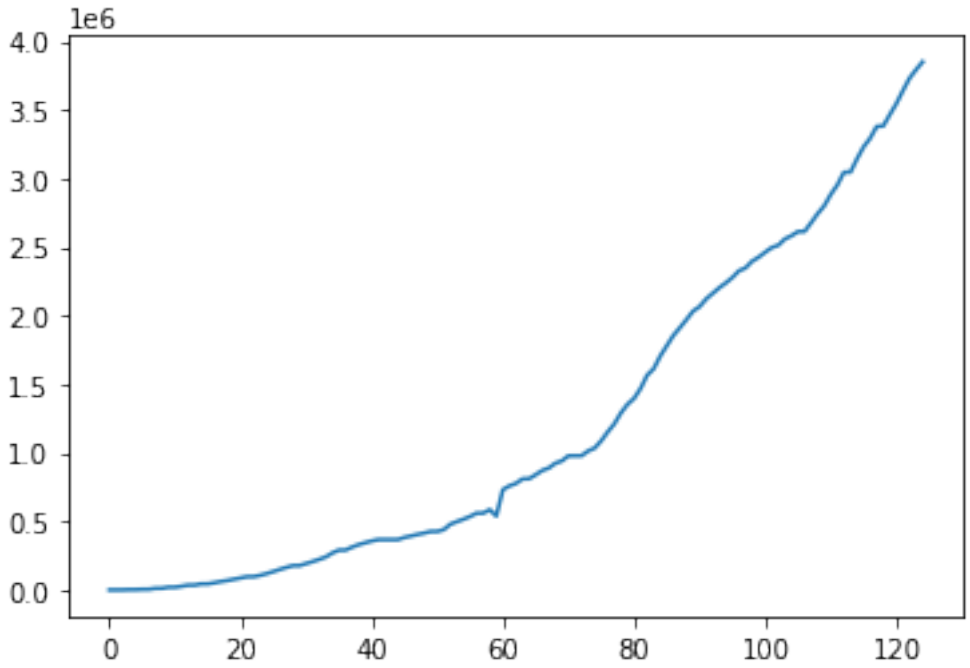
[101]:
```
# df1
```

[102]:
```
df2=df.loc[df['State'] == "Maharashtra"]
df1.reset_index(drop=True, inplace=True)
df2.reset_index(drop=True, inplace=True)
df3=df.loc[df['State'] == "Uttar Pradesh"]
df3.reset_index(drop=True, inplace=True)
df4=df.loc[df['State'] == "Rajasthan"]
df4.reset_index(drop=True, inplace=True)
df2.head()
df3.head()
df4.head()
```

[102]:
```
   Updated On  ... Total Doses Administered
0  16-01-2021  ...                     3285
1  17-01-2021  ...                     3455
2  18-01-2021  ...                     4037
3  19-01-2021  ...                    13022
4  20-01-2021  ...                    13146

[5 rows x 4 columns]
```

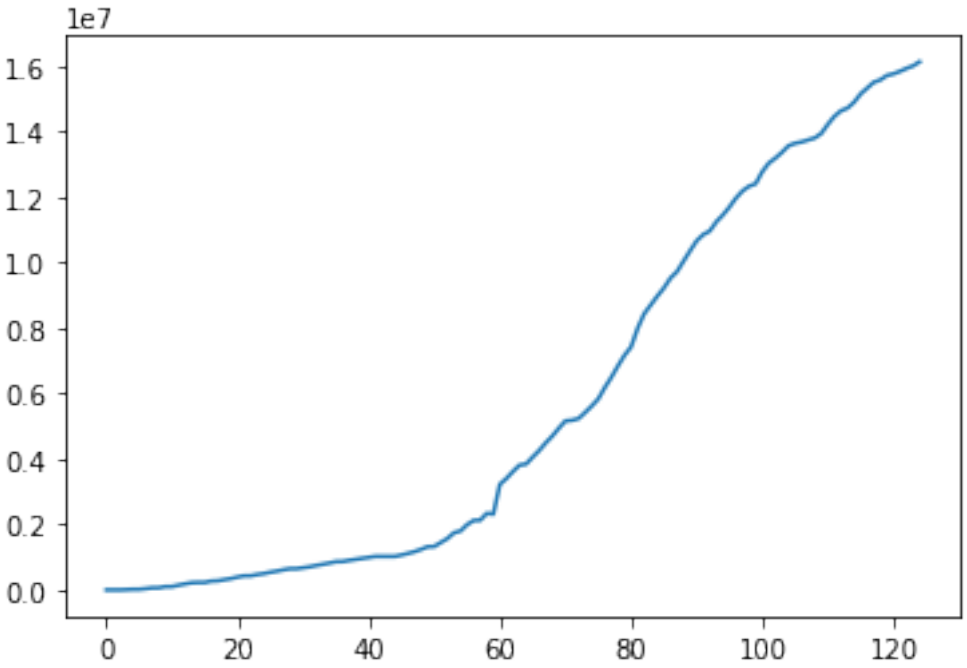
[103]:
```
df1['Total Individuals Vaccinated'].plot()
```

[103]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd87f66acd0>
```



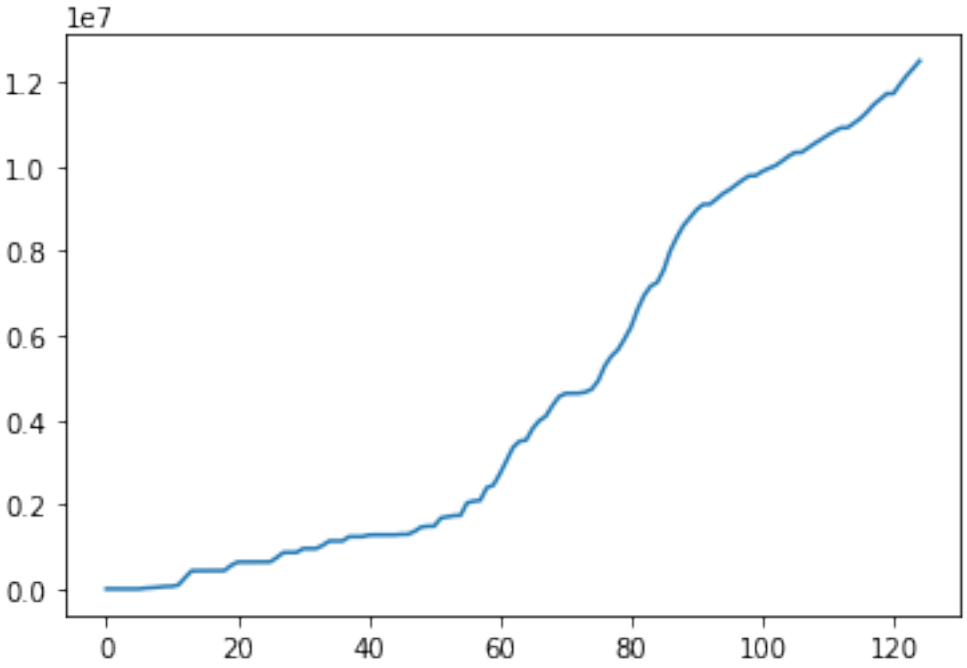
[104]:
```
df2['Total Individuals Vaccinated'].plot()
```

[104]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd87f5a36d0>
```



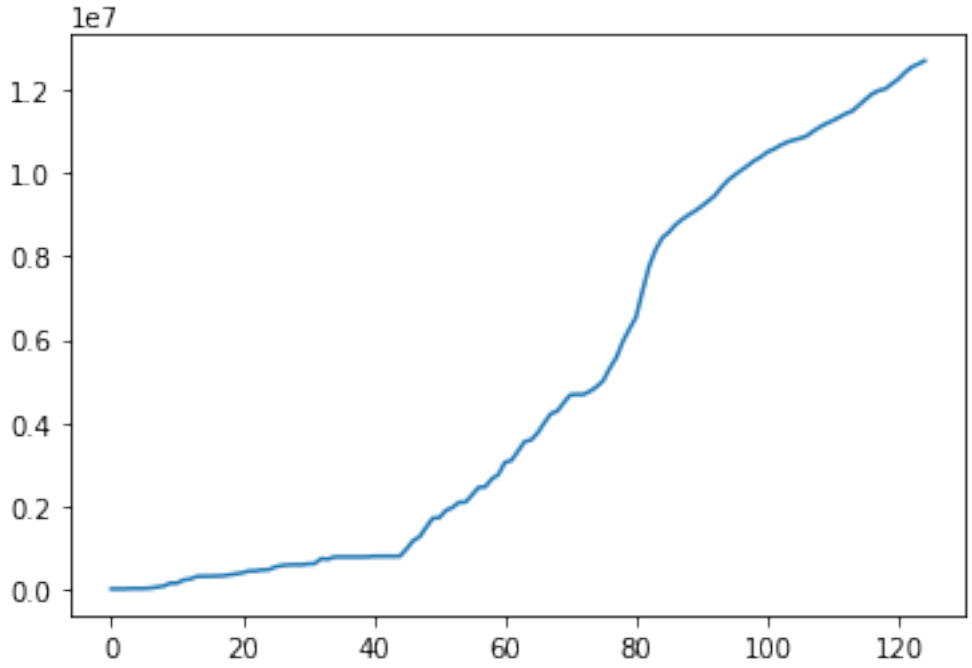
[105]:
```
df3['Total Individuals Vaccinated'].plot()
```

[105]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd87f5a8650>
```



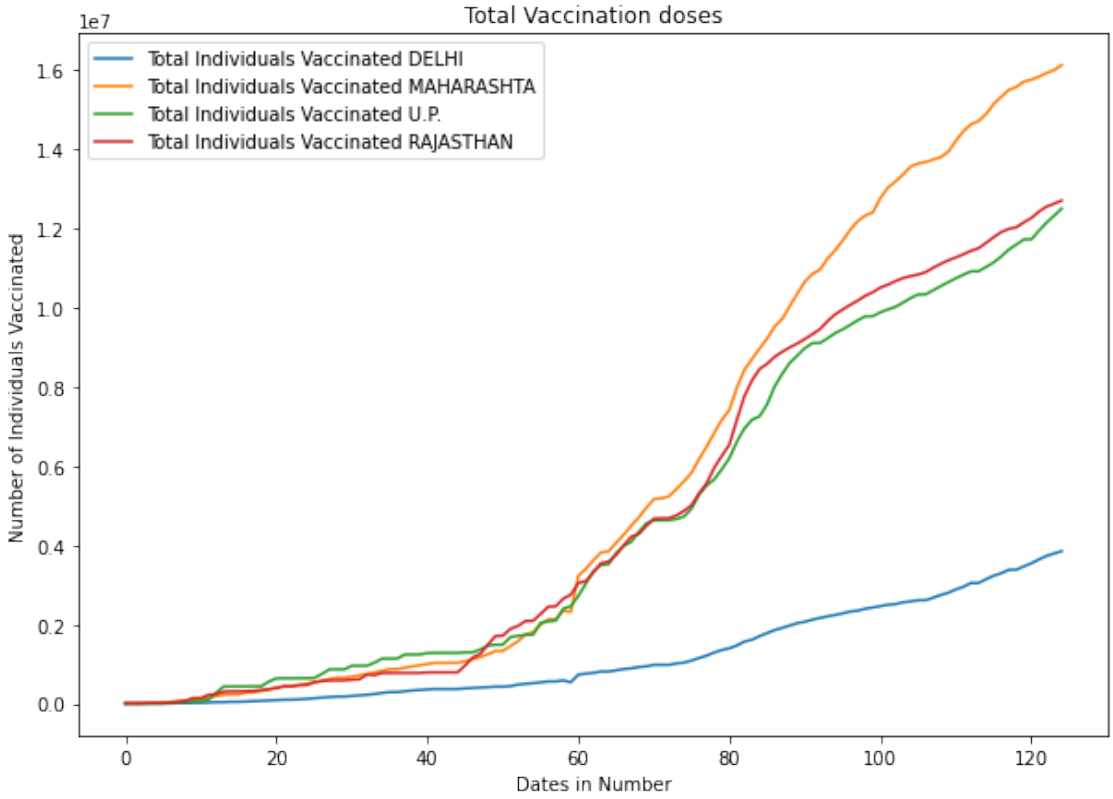
[106]:
```
df4['Total Individuals Vaccinated'].plot()
```

[106]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd87f494910>
```



[107]:
```
from matplotlib.dates import DateFormatter
import matplotlib.dates as mdates

df["Updated On"]= pd.to_datetime(df["Updated On"])

plt.figure(figsize = (10,7))
plt.plot( df1['Total Individuals Vaccinated'], label='Total Individuals␣
 ↪Vaccinated DELHI')
plt.plot( df2['Total Individuals Vaccinated'], label='Total Individuals␣
 ↪Vaccinated MAHARASHTA')
plt.plot( df3['Total Individuals Vaccinated'], label='Total Individuals␣
 ↪Vaccinated U.P.')
plt.plot( df4['Total Individuals Vaccinated'], label='Total Individuals␣
 ↪Vaccinated RAJASTHAN')
plt.xlabel("Dates in Number")
plt.ylabel("Number of Individuals Vaccinated")
plt.title("Total Vaccination doses")
# # myFmt = mdates.DateFormatter("%Y-%m-%d")
# # plt.gca().xaxis.set_major_formatter(myFmt)
# plt.xaxis.set_major_locator(mdates.WeekdayLocator(interval=50))
# plt.xaxis.set_major_formatter(DateFormatter("%d-%m-%Y"))
plt.legend()
```

```
plt.show()
```



[108]:
```
data = df1['Total Individuals Vaccinated']
look_back = 14

data = np.array(data)
data = data.reshape(len(data),1)
print(data.shape)

scaler = MinMaxScaler(feature_range=(0, 1))
data = scaler.fit_transform(data)

train, test= np.split(data, [int(.8 *len(data))])
train = train.reshape(len(train) , 1)
test = test.reshape(len(test) , 1)
print(test.shape)
print(train.shape)

def preprocessing(data , days=look_back):
  X, Y = [], []
  for i in range(len(data)-days-1):
    a = data[i:(i+days), 0]
```

```
    X.append(a)
    Y.append(data[i + days, 0])
  return np.array(X), np.array(Y)

X_train,Y_train =preprocessing(train)
X_test,Y_test =preprocessing(test)

Y_train = Y_train.reshape(len(Y_train),1)
Y_test = Y_test.reshape(len(Y_test),1)

X_train = X_train.reshape(X_train.shape[0] , 1 ,X_train.shape[1])
X_test = X_test.reshape(X_test.shape[0] , 1 ,X_test.shape[1])

print(X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)
```

```
(125, 1)
(25, 1)
(100, 1)
(85, 1, 14) (10, 1, 14) (85, 1) (10, 1)
```

[109]:
```
X,Y = preprocessing(data)
Y = Y.reshape(len(Y),1)
X = X.reshape(X.shape[0] , 1 ,X.shape[1])

def learning_plot(history):
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('model loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(['train', 'test'], loc='upper left')
    plt.show()
    return
```

[110]:
```
def model_score(model, X_train, y_train, X_test, y_test):

  # trainPredict = model.predict(X_train)
  testPredict = model.predict(X_test)
  # trainPredict = scaler.inverse_transform(trainPredict)
  # y_train = scaler.inverse_transform(y_train)
  # testPredict = scaler.inverse_transform(testPredict)
  # y_test = scaler.inverse_transform(y_test)

  # print(y_train.shape)
  # print(trainPredict.shape)

  # trainScore = math.sqrt(mean_squared_error(y_train[0], trainPredict[0]))
  # print('Train Score: %.2f RMSE' % (trainScore))
```

```
  # testScore = math.sqrt(mean_squared_error(y_test[0], testPredict[0]))
  # print('Test Score: %.2f RMSE' % (testScore))
  # return trainScore, testScore

  trainScore = model.evaluate(X_train, y_train, verbose=0)
  print('Train Score: %.5f MSE (%.5f RMSE)' % (trainScore[0], math.
↪sqrt(trainScore[0])))
  testScore = model.evaluate(X_test, y_test, verbose=0)
  print('Test Score: %.5f MSE (%.5f RMSE)' % (testScore[0], math.
↪sqrt(testScore[0])))

  #print(classification_report(y_test, testPredict))
  #print(confusion_matrix(y_test, testPredict))

  #MNB_f1 = round(f1_score(y_test, testPredict, average='weighted'), 3)
  #MNB_accuracy = round((accuracy_score(y_test, testPredict)*100),2)
  score=model.evaluate(X_train, Y_train)
  # print("Accuracy : " , round((score[1]*100),2), " %")
  #print("f1_score : " , MNB_f1)

  return trainScore[0], testScore[0]
```

[ ]:
```
days = 14
model = Sequential()
model.add(GRU(256 , input_shape = (1 , days) , return_sequences=True))
model.add(Dropout(0.3))
model.add(LSTM(256))
model.add(Dropout(0.3))
model.add(Dense(64 ,  activation = 'relu'))
model.add(Dense(1))
print(model.summary())
```

[ ]:
```
optimizer = optimizers.Adam(lr=0.01)

model.compile(loss='mean_squared_error', optimizer=optimizer , metrics =␣
 ↪['mean_squared_error'])
history = model.fit(X_train, Y_train, epochs=100 , batch_size = 128 ,␣
 ↪validation_data = (X_test,Y_test))
```

[ ]:
```
learning_plot(history)
```

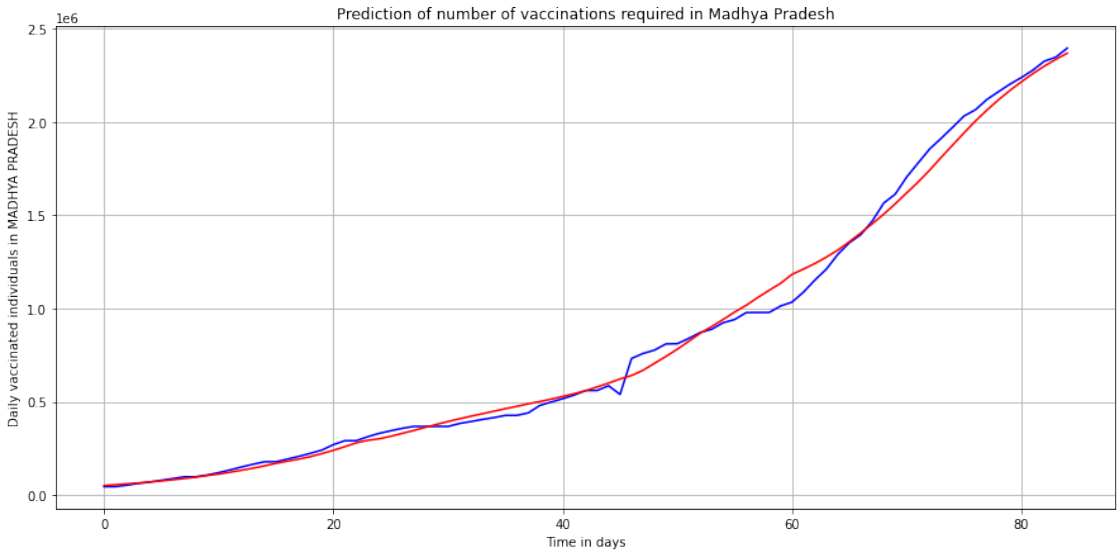
[90]:
```
model_score(model, X_train, Y_train , X_test, Y_test)
```

```
Train Score: 0.00017 MSE (0.01323 RMSE)
Test Score: 0.03871 MSE (0.19675 RMSE)
3/3 [==============================] - 0s 6ms/step - loss: 1.7497e-04 -
mean_squared_error: 1.7497e-04
Accuracy :  0.02  %
```

[90]:
```
(0.00017496921645943075, 0.03871142119169235)
```

[91]:
```
pred = model.predict(X_train)
pred = scaler.inverse_transform(pred)
y_test = Y_train.reshape(Y_train.shape[0] , 1)
y_test = scaler.inverse_transform(y_test)
print("Red - Predicted,  Blue - Actual")
plt.rcParams["figure.figsize"] = (15,7)
plt.plot(y_test , 'b')
plt.plot(pred , 'r')
plt.xlabel('Time in days')
plt.ylabel('Daily vaccinated individuals in MADHYA PRADESH')
plt.title("Prediction of number of vaccinations required in Madhya Pradesh")
plt.grid(True)
plt.show()
```

```
Red - Predicted,  Blue - Actual
```



[92]:
```
%%capture
!wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
from colab_pdf import colab_pdf
colab_pdf('Untitled14.ipynb')
```

[92]:

```
import math
import numpy as np
import pandas as pd
pd.options.mode.chained_assignment = None

from keras import optimizers
from keras.models import Sequential
from keras.layers import Dense , BatchNormalization , Dropout , Activation
from keras.layers import LSTM , GRU, Bidirectional
from keras.callbacks import ReduceLROnPlateau

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error

%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
```

Download Latest Data from OWID, Preprocess & Save

```
data_in= pd.read_csv('case_time_series (4).csv')
data_in
```

	Unnamed: 0	Date	Date_YMD	Daily Confirmed	Total Confirmed	Daily Recovered	Total Recovered	Daily Deceased	Total Deceased
0	0	2020-03-19	2020-03-19	27	198	5	20	1	4
1	1	2020-03-20	2020-03-20	58	256	3	23	0	4
2	2	2020-03-21	2020-03-21	78	334	0	23	0	4

```
data_in["Date"]= pd.to_datetime(data_in["Date"])
data_in
```

	Unnamed: 0	Date	Date_YMD	Daily Confirmed	Total Confirmed	Daily Recovered	Total Recovered	Daily Deceased	Total Deceased
0	0	2020-03-19	2020-03-19	27	198	5	20	1	4
1	1	2020-03-20	2020-03-20	58	256	3	23	0	4
2	2	2020-03-21	2020-03-21	78	334	0	23	0	4
3	3	2020-03-22	2020-03-22	69	403	0	23	3	7
4	4	2020-03-23	2020-03-23	94	497	2	25	2	9
...	...	...	...	...	...	...	...	...	...
423	423	2021-05-16	2021-05-16	281837	24964862	378526	21167515	4098	273826
424	424	2021-05-17	2021-05-17	263021	25227883	422391	21589906	4334	278160

```
data_in.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 428 entries, 0 to 427
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype         
---  ------           --------------  -----         
 0   Unnamed: 0       428 non-null    int64         
 1   Date             428 non-null    datetime64[ns]
 2   Date_YMD         428 non-null    object        
 3   Daily Confirmed  428 non-null    int64         
 4   Total Confirmed  428 non-null    int64         
 5   Daily Recovered  428 non-null    int64         
 6   Total Recovered  428 non-null    int64         
 7   Daily Deceased   428 non-null    int64         
 8   Total Deceased   428 non-null    int64         
dtypes: datetime64[ns](1), int64(7), object(1)
memory usage: 30.2+ KB
```

```
data_in.tail()
```

	Unnamed: 0	Date	Date_YMD	Daily Confirmed	Total Confirmed	Daily Recovered	Total Recovered	Daily Deceased	Total Deceased
423	423	2021-05-16	2021-05-16	281837	24964862	378526	21167515	4098	273826
424	424	2021-05-17	2021-05-17	263021	25227883	422391	21589906	4334	278160
425	425	2021-05-18	2021-05-18	267246	25495129	389758	21979664	4529	282689

```
# data_in.drop(['Total Confirmed','Total Recovered','Total Deceased'], axis=1, inplace=True)
```

```
data_in.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 428 entries, 0 to 427
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype         
---  ------           --------------  -----         
```

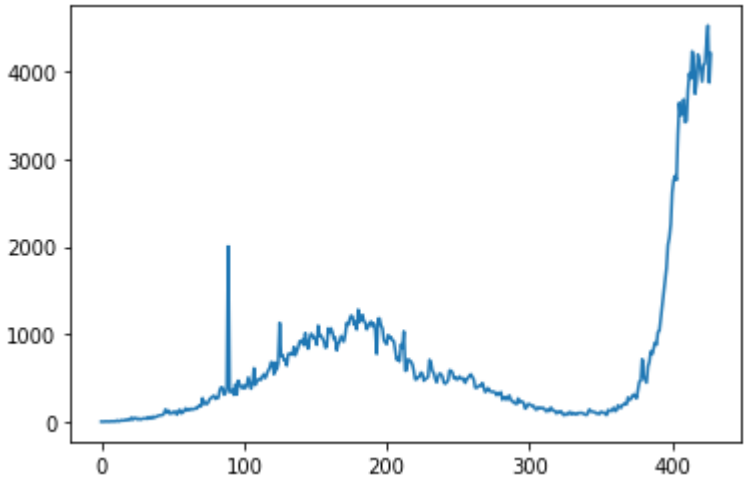
```
 0   Unnamed: 0       428 non-null    int64
 1   Date             428 non-null    datetime64[ns]
 2   Date_YMD         428 non-null    object
 3   Daily Confirmed  428 non-null    int64
 4   Total Confirmed  428 non-null    int64
 5   Daily Recovered  428 non-null    int64
 6   Total Recovered  428 non-null    int64
 7   Daily Deceased   428 non-null    int64
 8   Total Deceased   428 non-null    int64
dtypes: datetime64[ns](1), int64(7), object(1)
memory usage: 30.2+ KB
```

```
df1  = data_in[data_in['Date']>'2020-03-18']
# df1.drop(['location'],axis = 1,inplace = True)
df1.reset_index(drop=True, inplace=True)
df1
```

	Unnamed: 0	Date	Date_YMD	Daily Confirmed	Total Confirmed	Daily Recovered	Total Recovered	Daily Deceased	Total Deceased
0	0	2020-03-19	2020-03-19	27	198	5	20	1	4
1	1	2020-03-20	2020-03-20	58	256	3	23	0	4
2	2	2020-03-21	2020-03-21	78	334	0	23	0	4
3	3	2020-03-22	2020-03-22	69	403	0	23	3	7
4	4	2020-03-23	2020-03-23	94	497	2	25	2	9
...	...	...	...	...	...	...	...	...	...
423	423	2021-05-16	2021-05-16	281837	24964862	378526	21167515	4098	273826
424	424	2021-05-17	2021-05-17	263021	25227883	422391	21589906	4334	278160

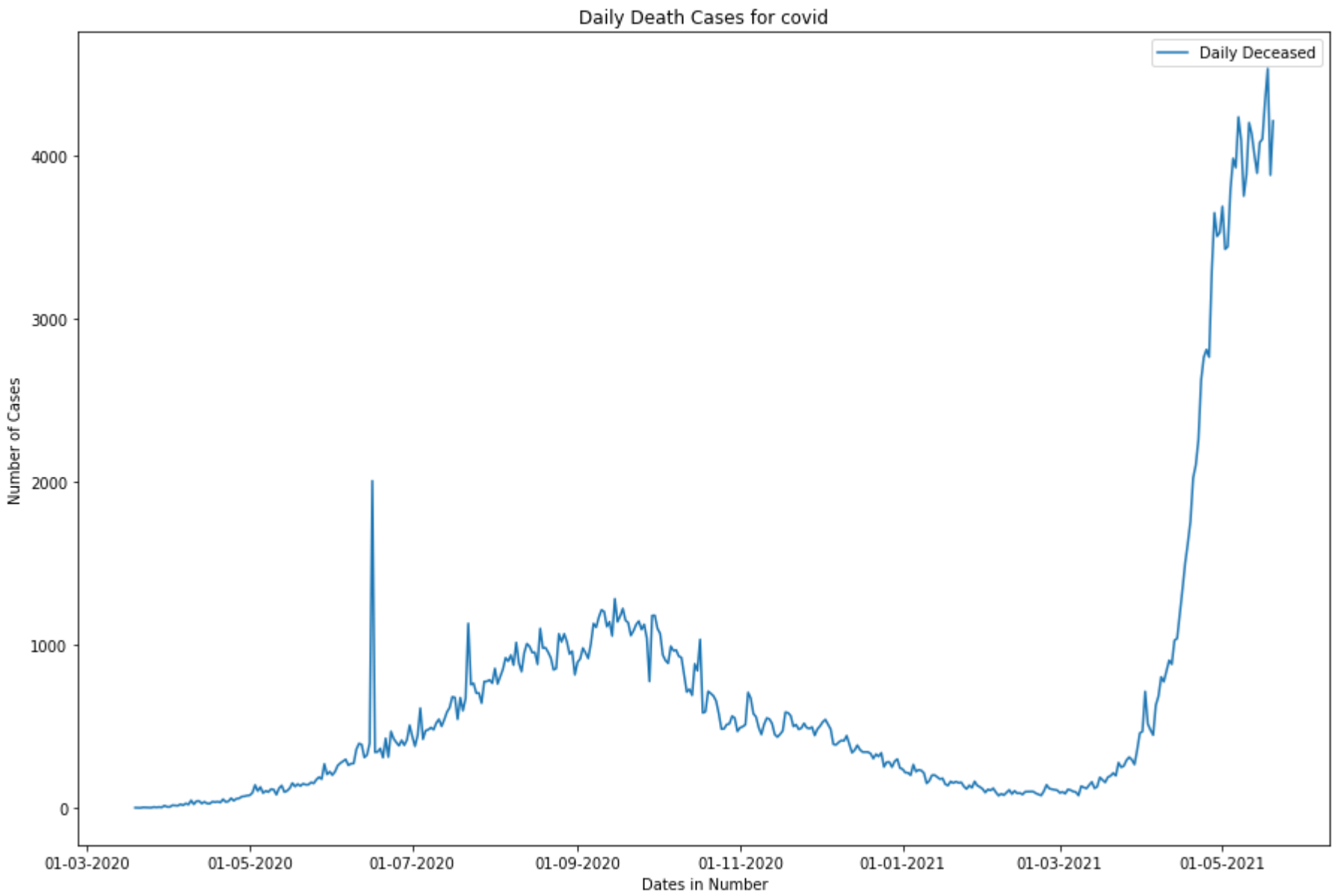
```
# df1.drop(['total_cases','total_deaths'],axis =1,inplace = True)


data = df1['Daily Deceased']


df1.to_csv('case_time_series (4).csv')


df1['Daily Deceased'].plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbf36377650>
```




```
from matplotlib.dates import DateFormatter
import matplotlib.dates as mdates

df1["Date_YMD"]= pd.to_datetime(df1["Date_YMD"])

plt.figure(figsize = (15,10))
# plt.plot(df1['Date_YMD'], df1['Daily Confirmed'], label='Daily Confirmed')
# plt.plot(df1['Date_YMD'], df1['Daily Recovered'], label='Daily Recovered')
plt.plot(df1['Date_YMD'], df1['Daily Deceased'], label='Daily Deceased')
plt.xlabel("Dates in Number")
plt.ylabel("Number of Cases")
plt.title("Daily Death Cases for covid")
myFmt = mdates.DateFormatter("%d-%m-%Y")
```

```
    plt.gca().xaxis.set_major_formatter(myFmt)
    plt.legend()
    plt.show()
```



Start with pre-processed data

```
df2 = pd.read_csv('case_time_series (4).csv')
df2.head()
```

	Unnamed: 0	Unnamed: 0.1	Date	Date_YMD	Daily Confirmed	Total Confirmed	Daily Recovered	Total Recovered	Daily Deceased	Total Deceased
0	0	0	2020-03-19	2020-03-19	27	198	5	20	1	4
1	1	1	2020-03-20	2020-03-20	58	256	3	23	0	4
2	2	2	2020-03-21	2020-03-21	78	334	0	23	0	4

```
data = df2['Daily Deceased']
```

Prediction and Analysis

Preprocessing

```
look_back = 14


data = np.array(data)
data = data.reshape(len(data),1)
print(data.shape)

scaler = MinMaxScaler(feature_range=(0, 1))
data = scaler.fit_transform(data)
```

```
(428, 1)
```

```
train, test= np.split(data, [int(.8 *len(data))])
train = train.reshape(len(train) , 1)
test = test.reshape(len(test) , 1)
print(test.shape)
print(train.shape)
```

```
(86, 1)
(342, 1)
```

```
def preprocessing(data , days=look_back):
    X, Y = [], []
    for i in range(len(data)-days-1):
        a = data[i:(i+days), 0]
        X.append(a)
        Y.append(data[i + days, 0])
    return np.array(X), np.array(Y)
```

```
X_train,Y_train =preprocessing(train)
X_test,Y_test =preprocessing(test)
```

```
Y_train = Y_train.reshape(len(Y_train),1)
Y_test = Y_test.reshape(len(Y_test),1)
```

```
X_train = X_train.reshape(X_train.shape[0] , 1 ,X_train.shape[1])
X_test = X_test.reshape(X_test.shape[0] , 1 ,X_test.shape[1])
```

```
print(X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)
```

```
(327, 1, 14) (71, 1, 14) (327, 1) (71, 1)
```

Training

```
X,Y = preprocessing(data)
Y = Y.reshape(len(Y),1)
X = X.reshape(X.shape[0] , 1 ,X.shape[1])


def learning_plot(history):
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('model loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(['train', 'test'], loc='upper left')
    plt.show()
    return


def model_score(model, X_train, y_train, X_test, y_test):

    # trainPredict = model.predict(X_train)
    # testPredict = model.predict(X_test)
    # trainPredict = scaler.inverse_transform(trainPredict)
    # y_train = scaler.inverse_transform(y_train)
    # testPredict = scaler.inverse_transform(testPredict)
    # y_test = scaler.inverse_transform(y_test)

    # print(y_train.shape)
    # print(trainPredict.shape)

    # trainScore = math.sqrt(mean_squared_error(y_train[0], trainPredict[0]))
    # print('Train Score: %.2f RMSE' % (trainScore))
    # testScore = math.sqrt(mean_squared_error(y_test[0], testPredict[0]))
    # print('Test Score: %.2f RMSE' % (testScore))
    # return trainScore, testScore

    trainScore = model.evaluate(X_train, y_train, verbose=0)
    print('Train Score: %.5f MSE (%.5f RMSE)' % (trainScore[0], math.sqrt(trainScore[0])))
    testScore = model.evaluate(X_test, y_test, verbose=0)
    print('Test Score: %.5f MSE (%.5f RMSE)' % (testScore[0], math.sqrt(testScore[0])))
    return trainScore[0], testScore[0]
```

Model A

```
days = 14
model = Sequential()
model.add(GRU(256 , input_shape = (1 , days) , return_sequences=True))
model.add(Dropout(0.3))
model.add(LSTM(256))
model.add(Dropout(0.3))
model.add(Dense(64 ,  activation = 'relu'))
model.add(Dense(1))
print(model.summary())
```

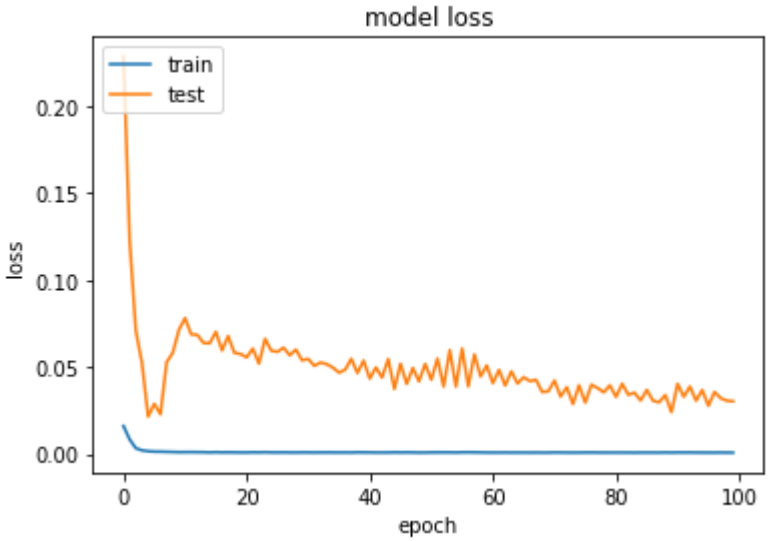
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
gru (GRU)                    (None, 1, 256)            208896    
_________________________________________________________________
dropout (Dropout)            (None, 1, 256)            0         
_________________________________________________________________
lstm (LSTM)                  (None, 256)               525312    
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0         
_________________________________________________________________
dense (Dense)                (None, 64)                16448     
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 65        
=================================================================
Total params: 750,721
Trainable params: 750,721
Non-trainable params: 0
_________________________________________________________________
None
```

```
optimizer = optimizers.Adam(lr=0.01)

model.compile(loss='mean_squared_error', optimizer=optimizer , metrics = ['mean_squared_error'])
history = model.fit(X_train, Y_train, epochs=100 , batch_size = 128 , validation_data = (X_test,Y_test))
```

```
3/3 [==============================] - 0s 34ms/step - loss: 6.4942e-04 - mean_squared_error: 6.4942e-04 - val_loss:
Epoch 71/100
3/3 [==============================] - 0s 32ms/step - loss: 9.8927e-04 - mean_squared_error: 9.8927e-04 - val_loss:
Epoch 72/100
3/3 [==============================] - 0s 38ms/step - loss: 7.1405e-04 - mean_squared_error: 7.1405e-04 - val_loss:
Epoch 73/100
3/3 [==============================] - 0s 34ms/step - loss: 6.9774e-04 - mean_squared_error: 6.9774e-04 - val_loss:
Epoch 74/100
3/3 [==============================] - 0s 37ms/step - loss: 5.6179e-04 - mean_squared_error: 5.6179e-04 - val_loss:
Epoch 75/100
3/3 [==============================] - 0s 36ms/step - loss: 7.1817e-04 - mean_squared_error: 7.1817e-04 - val_loss:
Epoch 76/100
3/3 [==============================] - 0s 34ms/step - loss: 5.7591e-04 - mean_squared_error: 5.7591e-04 - val_loss:
Epoch 77/100
3/3 [==============================] - 0s 32ms/step - loss: 6.4950e-04 - mean_squared_error: 6.4950e-04 - val_loss:
Epoch 78/100
3/3 [==============================] - 0s 34ms/step - loss: 9.7999e-04 - mean_squared_error: 9.7999e-04 - val_loss:
Epoch 79/100
3/3 [==============================] - 0s 34ms/step - loss: 6.4503e-04 - mean_squared_error: 6.4503e-04 - val_loss:
Epoch 80/100
3/3 [==============================] - 0s 34ms/step - loss: 7.0313e-04 - mean_squared_error: 7.0313e-04 - val_loss:
Epoch 81/100
3/3 [==============================] - 0s 33ms/step - loss: 7.0526e-04 - mean_squared_error: 7.0526e-04 - val_loss:
Epoch 82/100
3/3 [==============================] - 0s 33ms/step - loss: 8.9680e-04 - mean_squared_error: 8.9680e-04 - val_loss:
Epoch 83/100
3/3 [==============================] - 0s 39ms/step - loss: 9.7417e-04 - mean_squared_error: 9.7417e-04 - val_loss:
Epoch 84/100
3/3 [==============================] - 0s 33ms/step - loss: 6.7494e-04 - mean_squared_error: 6.7494e-04 - val_loss:
Epoch 85/100
3/3 [==============================] - 0s 39ms/step - loss: 6.1139e-04 - mean_squared_error: 6.1139e-04 - val_loss:
Epoch 86/100
3/3 [==============================] - 0s 35ms/step - loss: 9.6156e-04 - mean_squared_error: 9.6156e-04 - val_loss:
Epoch 87/100
3/3 [==============================] - 0s 35ms/step - loss: 6.7156e-04 - mean_squared_error: 6.7156e-04 - val_loss:
Epoch 88/100
3/3 [==============================] - 0s 35ms/step - loss: 6.3919e-04 - mean_squared_error: 6.3919e-04 - val_loss:
Epoch 89/100
3/3 [==============================] - 0s 33ms/step - loss: 0.0010 - mean_squared_error: 0.0010 - val_loss: 0.0338 -
Epoch 90/100
3/3 [==============================] - 0s 35ms/step - loss: 5.8181e-04 - mean_squared_error: 5.8181e-04 - val_loss:
Epoch 91/100
3/3 [==============================] - 0s 34ms/step - loss: 6.1355e-04 - mean_squared_error: 6.1355e-04 - val_loss:
```

```
Epoch 92/100
3/3 [==============================] - 0s 39ms/step - loss: 9.8794e-04 - mean_squared_error: 9.8794e-04 - val_loss:
Epoch 93/100
3/3 [==============================] - 0s 40ms/step - loss: 0.0010 - mean_squared_error: 0.0010 - val_loss: 0.0387 -
Epoch 94/100
3/3 [==============================] - 0s 34ms/step - loss: 7.1144e-04 - mean_squared_error: 7.1144e-04 - val_loss:
Epoch 95/100
3/3 [==============================] - 0s 35ms/step - loss: 7.0980e-04 - mean_squared_error: 7.0980e-04 - val_loss:
Epoch 96/100
3/3 [==============================] - 0s 34ms/step - loss: 5.3634e-04 - mean_squared_error: 5.3634e-04 - val_loss:
Epoch 97/100
3/3 [==============================] - 0s 34ms/step - loss: 8.8627e-04 - mean_squared_error: 8.8627e-04 - val_loss:
Epoch 98/100
3/3 [==============================] - 0s 35ms/step - loss: 9.5610e-04 - mean_squared_error: 9.5610e-04 - val_loss:
Epoch 99/100
3/3 [==============================] - 0s 35ms/step - loss: 9.0988e-04 - mean_squared_error: 9.0988e-04 - val_loss:
```

```
learning_plot(history)
```



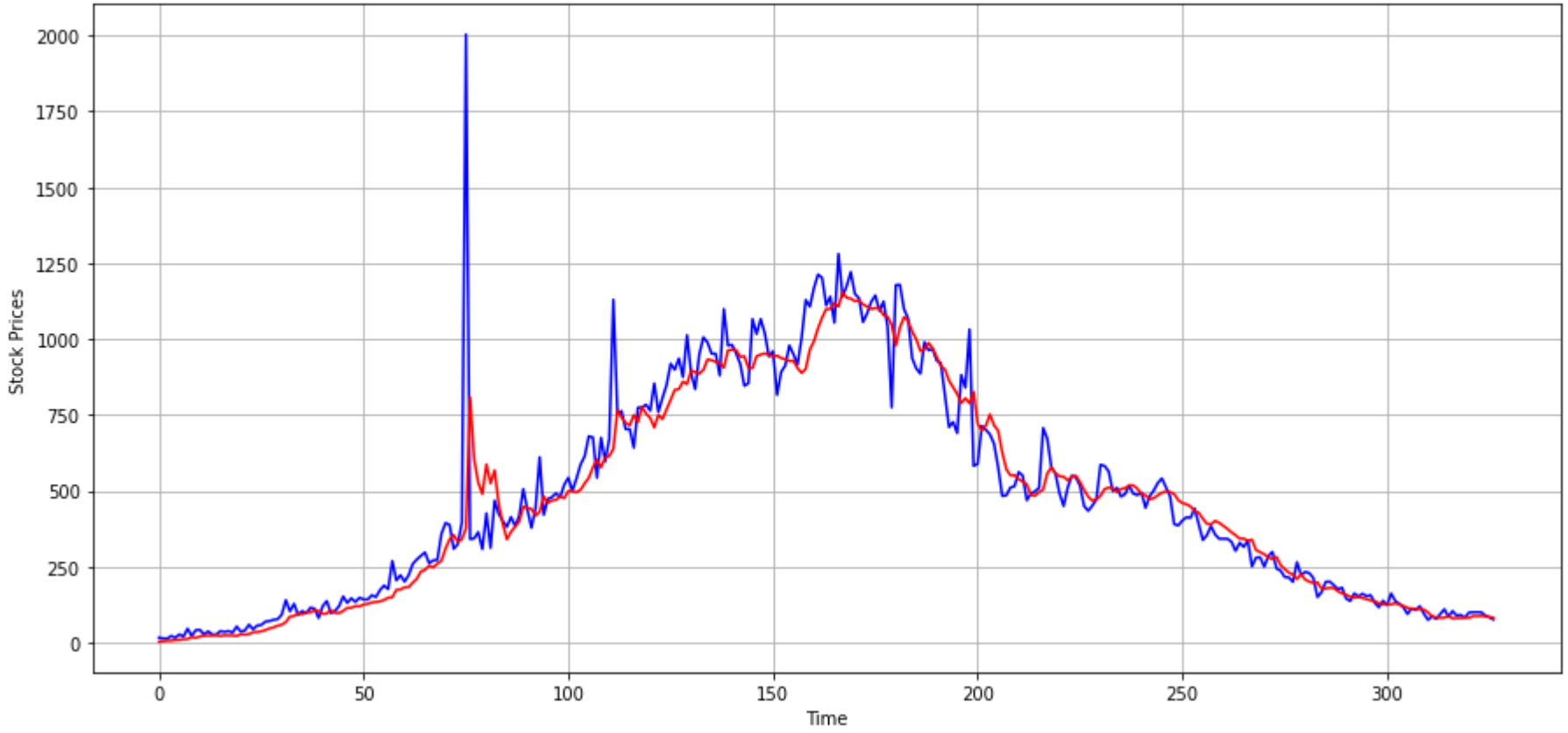

```
model_score(model, X_train, Y_train , X_test, Y_test)
```

```
Train Score: 0.00069 MSE (0.02618 RMSE)
Test Score: 0.03026 MSE (0.17394 RMSE)
```

```
(0.0006852579535916448, 0.030256839469075203)
```

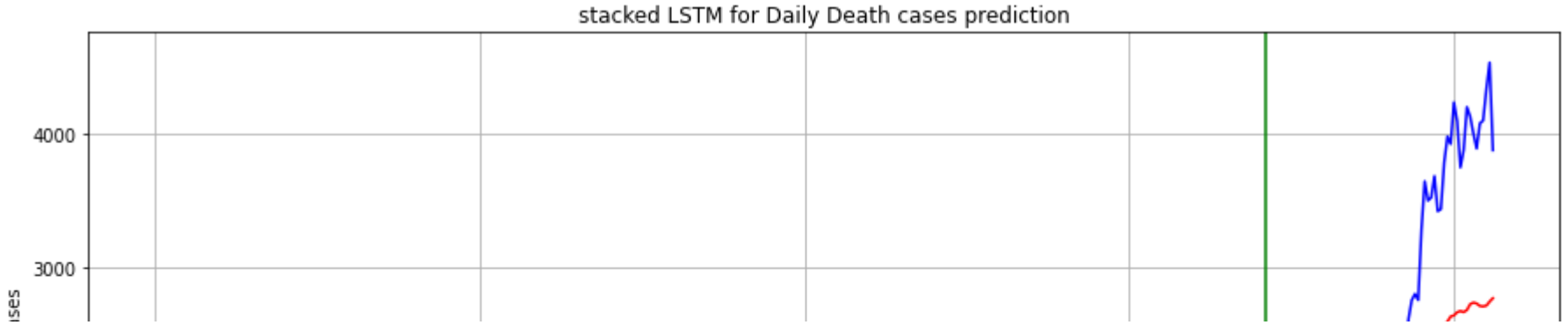
```
pred = model.predict(X_train)
pred = scaler.inverse_transform(pred)
y_test = Y_train.reshape(Y_train.shape[0] , 1)
y_test = scaler.inverse_transform(y_test)
print("Red - Predicted,  Blue - Actual")
plt.rcParams["figure.figsize"] = (15,7)
plt.plot(y_test , 'b')
plt.plot(pred , 'r')
plt.xlabel('Time')
plt.ylabel('Stock Prices')
plt.grid(True)
plt.show()
```

```
Red - Predicted,  Blue - Actual
```



```
from matplotlib.dates import DateFormatter
import matplotlib.dates as mdates


def model_graph(X,model,Y):
    pred = model.predict(X)
    pred = scaler.inverse_transform(pred)
    y_test = Y.reshape(Y.shape[0] , 1)
    y_test = scaler.inverse_transform(y_test)
    print("Red - Predicted,  Blue - Actual")
    plt.rcParams["figure.figsize"] = (15,7)
    plt.plot(y_test , 'b')
    plt.plot(pred , 'r')
    plt.axvline(train.shape[0], color='g')
    plt.xlabel('Time in days')
    plt.ylabel('Daily Death Cases')
    # myFmt = mdates.DateFormatter("%d-%m-%Y")
    # plt.gca().xaxis.set_major_formatter(myFmt)
    plt.title("stacked LSTM for Daily Death cases prediction")
    plt.grid(True)
    plt.show()
    return
model_graph(X,model,Y)
```

```
Red - Predicted,  Blue - Actual
```



Model B

```
model_b = Sequential()
model_b.add(Bidirectional(LSTM(256, activation='relu',return_sequences = True), input_shape=(1,14)))
model_b.add(Dropout(0.3))
model_b.add(Dense(128 ,  activation = 'relu'))

model_b.add(Dense(1,activation='relu'))

print(model_b.summary())
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional (Bidirectional (None, 1, 512)            555008
_________________________________________________________________
dropout_2 (Dropout)          (None, 1, 512)            0
_________________________________________________________________
dense_2 (Dense)              (None, 1, 128)            65664
_________________________________________________________________
dense_3 (Dense)              (None, 1, 1)              129
=================================================================
Total params: 620,801
Trainable params: 620,801
Non-trainable params: 0
```

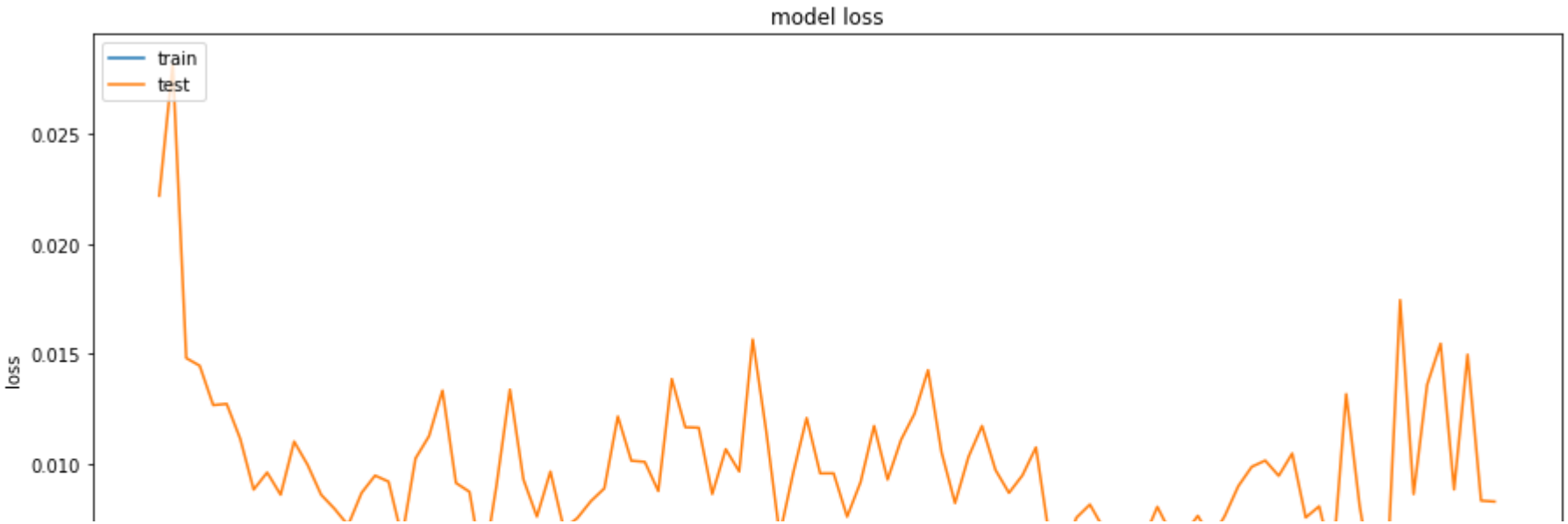
```
_________________________________________________________________
None
```

```
optimizer = optimizers.Adam(lr=0.001)
model_b.compile(loss='mean_squared_error', optimizer=optimizer , metrics = ['mean_squared_error'])
history_b = model_b.fit(X_train, Y_train, epochs=100 , batch_size = 32 ,
          validation_data = (X_test,Y_test))
```

```
Epoch 1/100
11/11 [==============================] - 3s 49ms/step - loss: 0.0080 - mean_squared_error: 0.0080 - val_loss: 0.0222
Epoch 2/100
11/11 [==============================] - 0s 13ms/step - loss: 0.0014 - mean_squared_error: 0.0014 - val_loss: 0.0281
Epoch 3/100
11/11 [==============================] - 0s 13ms/step - loss: 7.3582e-04 - mean_squared_error: 7.3582e-04 - val_loss
Epoch 4/100
11/11 [==============================] - 0s 13ms/step - loss: 8.9296e-04 - mean_squared_error: 8.9296e-04 - val_loss
Epoch 5/100
11/11 [==============================] - 0s 13ms/step - loss: 0.0015 - mean_squared_error: 0.0015 - val_loss: 0.0127
Epoch 6/100
11/11 [==============================] - 0s 12ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 0.0127
Epoch 7/100
11/11 [==============================] - 0s 13ms/step - loss: 7.6632e-04 - mean_squared_error: 7.6632e-04 - val_loss
Epoch 8/100
11/11 [==============================] - 0s 14ms/step - loss: 5.2618e-04 - mean_squared_error: 5.2618e-04 - val_loss
Epoch 9/100
11/11 [==============================] - 0s 13ms/step - loss: 7.6828e-04 - mean_squared_error: 7.6828e-04 - val_loss
Epoch 10/100
11/11 [==============================] - 0s 13ms/step - loss: 0.0014 - mean_squared_error: 0.0014 - val_loss: 0.0086
Epoch 11/100
11/11 [==============================] - 0s 13ms/step - loss: 8.2338e-04 - mean_squared_error: 8.2338e-04 - val_loss
Epoch 12/100
11/11 [==============================] - 0s 13ms/step - loss: 9.4706e-04 - mean_squared_error: 9.4706e-04 - val_loss
Epoch 13/100
11/11 [==============================] - 0s 13ms/step - loss: 4.8972e-04 - mean_squared_error: 4.8972e-04 - val_loss
Epoch 14/100
11/11 [==============================] - 0s 13ms/step - loss: 5.0747e-04 - mean_squared_error: 5.0747e-04 - val_loss
Epoch 15/100
11/11 [==============================] - 0s 15ms/step - loss: 6.0098e-04 - mean_squared_error: 6.0098e-04 - val_loss
Epoch 16/100
11/11 [==============================] - 0s 13ms/step - loss: 0.0014 - mean_squared_error: 0.0014 - val_loss: 0.0087
Epoch 17/100
11/11 [==============================] - 0s 13ms/step - loss: 7.5975e-04 - mean_squared_error: 7.5975e-04 - val_loss
Epoch 18/100
```

```
11/11 [==============================] - 0s 13ms/step - loss: 6.2154e-04 - mean_squared_error: 6.2154e-04 - val_loss
Epoch 19/100
11/11 [==============================] - 0s 13ms/step - loss: 3.3991e-04 - mean_squared_error: 3.3991e-04 - val_loss
Epoch 20/100
11/11 [==============================] - 0s 13ms/step - loss: 9.9044e-04 - mean_squared_error: 9.9044e-04 - val_loss
Epoch 21/100
11/11 [==============================] - 0s 13ms/step - loss: 5.1115e-04 - mean_squared_error: 5.1115e-04 - val_loss
Epoch 22/100
11/11 [==============================] - 0s 15ms/step - loss: 8.5036e-04 - mean_squared_error: 8.5036e-04 - val_loss
Epoch 23/100
11/11 [==============================] - 0s 12ms/step - loss: 4.5738e-04 - mean_squared_error: 4.5738e-04 - val_loss
Epoch 24/100
11/11 [==============================] - 0s 13ms/step - loss: 5.0343e-04 - mean_squared_error: 5.0343e-04 - val_loss
Epoch 25/100
11/11 [==============================] - 0s 13ms/step - loss: 3.9268e-04 - mean_squared_error: 3.9268e-04 - val_loss
Epoch 26/100
11/11 [==============================] - 0s 13ms/step - loss: 0.0017 - mean_squared_error: 0.0017 - val_loss: 0.0090
Epoch 27/100
11/11 [==============================] - 0s 12ms/step - loss: 8.5845e-04 - mean_squared_error: 8.5845e-04 - val_loss
Epoch 28/100
11/11 [==============================] - 0s 12ms/step - loss: 5.1936e-04 - mean_squared_error: 5.1936e-04 - val_loss
Epoch 29/100
11/11 [==============================] - 0s 14ms/step - loss: 4.2840e-04 - mean_squared_error: 4.2840e-04 - val_loss
```

```
plt.plot(history_b.history['loss'])
plt.plot(history_b.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```




```
model_score(model_b, X_train, Y_train , X_test, Y_test)
```

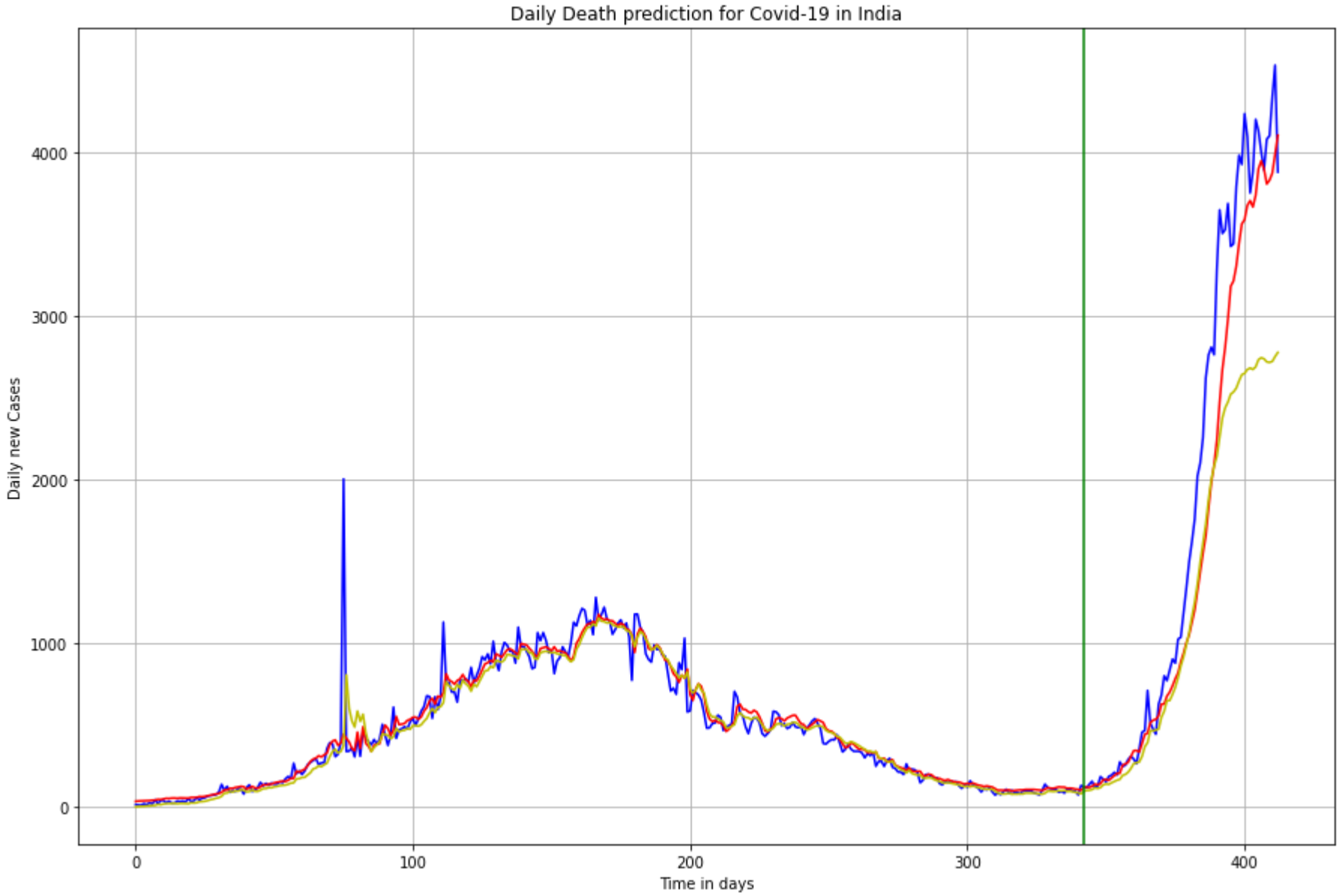
```
Train Score: 0.00056 MSE (0.02375 RMSE)
Test Score: 0.00832 MSE (0.09121 RMSE)
(0.0005639234441332519, 0.008319833315908909)
```

0 20 40 60 80 100

```
pred = model_b.predict(X)
pred1 = model.predict(X)
pred1 = scaler.inverse_transform(pred1)
y_test1 = Y.reshape(Y.shape[0] , 1)
y_test1 = scaler.inverse_transform(y_test1)
pred = pred.reshape(pred.shape[0],1)
pred = scaler.inverse_transform(pred)
y_test = Y.reshape(Y.shape[0] , 1)
y_test = scaler.inverse_transform(y_test)
print("Blue - Actual , Red -Bidirectional Predicted , Yellow- Stacked Prediction")
plt.rcParams["figure.figsize"] = (15,10)
plt.plot(y_test , 'b')
plt.plot(pred , 'r')
plt.plot(pred1 , 'y')
plt.xlabel('Time in days')
plt.axvline(train.shape[0], color='g')
```

```
plt.ylabel('Daily new Cases')
plt.title("Daily Death prediction for Covid-19 in India")
plt.grid(True)
plt.show()
```

Blue - Actual , Red -Bidirectional Predicted , Yellow- Stacked Prediction

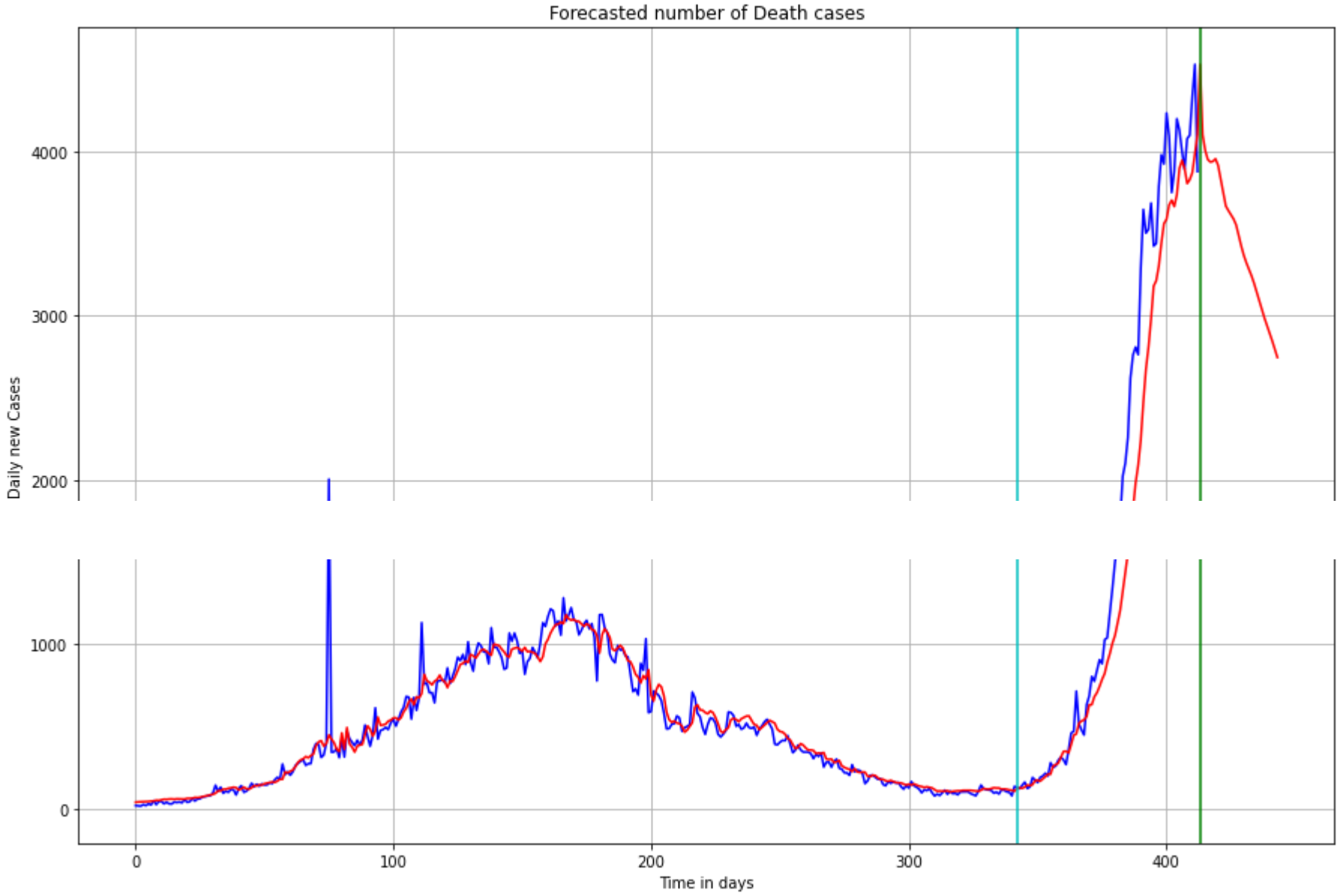


Forecasting

```
close_data = X.reshape((-1))
look_back = 14

def predict(num_prediction, model):
    prediction_list = close_data[-look_back:]
    for _ in range(num_prediction):
        x = prediction_list[-look_back:]
        x = x.reshape((1, look_back, 1))
        out = model.predict(x.reshape((1,1,14)))[0][0]
        prediction_list = np.append(prediction_list, out)
    prediction_list = prediction_list[look_back-1:]

    return prediction_list

def predict_dates(num_prediction):
    last_date = df2['Date'].values[-1]
    prediction_dates = pd.date_range(last_date, periods=num_prediction+1).tolist()
    return prediction_dates

num_prediction = 30
forecast_dates = predict_dates(num_prediction)
forecast = predict(num_prediction, model_b).reshape(-1,1)
forecast1 = scaler.inverse_transform(forecast)


from matplotlib.dates import DateFormatter
import matplotlib.dates as mdates

pred = model_b.predict(X)
pred = pred.reshape(pred.shape[0],1)
pred = np.concatenate((pred, forecast), axis = 0)
pred = scaler.inverse transform(pred)
```

```
# Yforecast = np.concatenate((Y, forecast), axis = 0)
y_test = Y.reshape(Y.shape[0] , 1)
y_test = scaler.inverse_transform(y_test)
print("Red - Predicted,  Blue - Actual")
plt.rcParams["figure.figsize"] = (15,10)
plt.plot(y_test , 'b')
plt.plot(pred , 'r')
plt.xlabel('Time in days')
plt.axvline(train.shape[0], color='c')
plt.axvline(Y.shape[0], color='g')
# plt.axvline(x=53, color='y', label='lockdown 1-23/03/2020')
# plt.axvline(x=76, label='lockdown 2-15/04/2020', color='y')
# plt.axvline(x=95, label='lockdown 3-04/05/2020', color='y')
# plt.axvline(x=109, label='lockdown 4-18/05/2020', color='y')
# plt.axvline(x=123, label='Unlock 1-01/06/2020', color='y')
# plt.axvline(x=153, label='Unlock 2-01/07/2020', color='y')
# plt.axvline(x=184, label='Unlock 3-01/08/2020', color='y')
# myFmt = mdates.DateFormatter("%d-%m-%Y")
# plt.gca().xaxis.set_major_formatter(mdates.DateFormatter("%d-%m-%Y"))
plt.ylabel('Daily new Cases')
plt.title("Forecasted number of Death cases")
plt.grid(True)
plt.show()
```

```
Red - Predicted,  Blue - Actual
```



13s completed at 21:52

Untitled33

May 23, 2021

[1]:
```
# importing all imp libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import os
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
import keras.backend as K
from keras.optimizers import Adam
from keras.models import load_model
from keras.layers import LSTM
np.random.seed(7)
```

[2]:
```
!pip install plotly>=4.9.0
!wget https://github.com/plotly/orca/releases/download/v1.2.1/orca-1.2.1-x86_64.
 ↪AppImage -O /usr/local/bin/orca
!chmod +x /usr/local/bin/orca
!apt-get install xvfb libgtk2.0-0 libgconf-2-4
```

```
--2021-05-23 15:54:58--  https://github.com/plotly/orca/releases/download/v1.2.1
/orca-1.2.1-x86_64.AppImage
Resolving github.com (github.com)... 140.82.112.4
Connecting to github.com (github.com)|140.82.112.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-releases.githubusercontent.com/99037241/9dc3a580-286a-
11e9-8a21-4312b7c8a512?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210523%2Fus-east-1%2Fs3%2Faws4_request&X
-Amz-Date=20210523T155458Z&X-Amz-Expires=300&X-Amz-
Signature=50bcd33c31283c162ab31f8cd21bec9779aa942834d39362219a8e4cc68c7cda&X
-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=99037241&response-content-
disposition=attachment%3B%20filename%3Dorca-1.2.1-x86_64.AppImage&response-
content-type=application%2Foctet-stream [following]
--2021-05-23 15:54:58--  https://github-
releases.githubusercontent.com/99037241/9dc3a580-286a-11e9-8a21-4312b7c8a512?X
```

```
-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20210523
%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20210523T155458Z&X-Amz-Expires=300&X
-Amz-
Signature=50bcd33c31283c162ab31f8cd21bec9779aa942834d39362219a8e4cc68c7cda&X
-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=99037241&response-content-
disposition=attachment%3B%20filename%3Dorca-1.2.1-x86_64.AppImage&response-
content-type=application%2Foctet-stream
Resolving github-releases.githubusercontent.com (github-
releases.githubusercontent.com)... 185.199.108.154, 185.199.109.154,
185.199.110.154, ...
Connecting to github-releases.githubusercontent.com (github-
releases.githubusercontent.com)|185.199.108.154|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 51607939 (49M) [application/octet-stream]
Saving to: /usr/local/bin/orca

/usr/local/bin/orca 100%[===================>]  49.22M  76.0MB/s    in 0.6s

2021-05-23 15:54:59 (76.0 MB/s) - /usr/local/bin/orca saved
[51607939/51607939]

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libnvidia-common-460
Use 'apt autoremove' to remove it.
The following additional packages will be installed:
  gconf-service gconf-service-backend gconf2-common libdbus-glib-1-2
  libgail-common libgail18 libgtk2.0-bin libgtk2.0-common
Suggested packages:
  gvfs
The following NEW packages will be installed:
  gconf-service gconf-service-backend gconf2-common libdbus-glib-1-2
  libgail-common libgail18 libgconf-2-4 libgtk2.0-0 libgtk2.0-bin
  libgtk2.0-common xvfb
0 upgraded, 11 newly installed, 0 to remove and 34 not upgraded.
Need to get 3,715 kB of archives.
After this operation, 17.2 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 libdbus-glib-1-2 amd64
0.110-2 [58.3 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/universe amd64 gconf2-common all
3.2.6-4ubuntu1 [700 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libgconf-2-4 amd64
3.2.6-4ubuntu1 [84.8 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/universe amd64 gconf-service-
backend amd64 3.2.6-4ubuntu1 [58.1 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/universe amd64 gconf-service amd64
```

```
3.2.6-4ubuntu1 [2,036 B]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgtk2.0-common all
2.24.32-1ubuntu1 [125 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgtk2.0-0 amd64
2.24.32-1ubuntu1 [1,769 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgail18 amd64
2.24.32-1ubuntu1 [14.2 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgail-common amd64
2.24.32-1ubuntu1 [112 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgtk2.0-bin amd64
2.24.32-1ubuntu1 [7,536 B]
Get:11 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 xvfb amd64
2:1.19.6-1ubuntu4.9 [784 kB]
Fetched 3,715 kB in 1s (3,601 kB/s)
Selecting previously unselected package libdbus-glib-1-2:amd64.
(Reading database ... 160706 files and directories currently installed.)
Preparing to unpack .../00-libdbus-glib-1-2_0.110-2_amd64.deb ...
Unpacking libdbus-glib-1-2:amd64 (0.110-2) ...
Selecting previously unselected package gconf2-common.
Preparing to unpack .../01-gconf2-common_3.2.6-4ubuntu1_all.deb ...
Unpacking gconf2-common (3.2.6-4ubuntu1) ...
Selecting previously unselected package libgconf-2-4:amd64.
Preparing to unpack .../02-libgconf-2-4_3.2.6-4ubuntu1_amd64.deb ...
Unpacking libgconf-2-4:amd64 (3.2.6-4ubuntu1) ...
Selecting previously unselected package gconf-service-backend.
Preparing to unpack .../03-gconf-service-backend_3.2.6-4ubuntu1_amd64.deb ...
Unpacking gconf-service-backend (3.2.6-4ubuntu1) ...
Selecting previously unselected package gconf-service.
Preparing to unpack .../04-gconf-service_3.2.6-4ubuntu1_amd64.deb ...
Unpacking gconf-service (3.2.6-4ubuntu1) ...
Selecting previously unselected package libgtk2.0-common.
Preparing to unpack .../05-libgtk2.0-common_2.24.32-1ubuntu1_all.deb ...
Unpacking libgtk2.0-common (2.24.32-1ubuntu1) ...
Selecting previously unselected package libgtk2.0-0:amd64.
Preparing to unpack .../06-libgtk2.0-0_2.24.32-1ubuntu1_amd64.deb ...
Unpacking libgtk2.0-0:amd64 (2.24.32-1ubuntu1) ...
Selecting previously unselected package libgail18:amd64.
Preparing to unpack .../07-libgail18_2.24.32-1ubuntu1_amd64.deb ...
Unpacking libgail18:amd64 (2.24.32-1ubuntu1) ...
Selecting previously unselected package libgail-common:amd64.
Preparing to unpack .../08-libgail-common_2.24.32-1ubuntu1_amd64.deb ...
Unpacking libgail-common:amd64 (2.24.32-1ubuntu1) ...
Selecting previously unselected package libgtk2.0-bin.
Preparing to unpack .../09-libgtk2.0-bin_2.24.32-1ubuntu1_amd64.deb ...
Unpacking libgtk2.0-bin (2.24.32-1ubuntu1) ...
Selecting previously unselected package xvfb.
Preparing to unpack .../10-xvfb_2%3a1.19.6-1ubuntu4.9_amd64.deb ...
Unpacking xvfb (2:1.19.6-1ubuntu4.9) ...
```

```
Setting up gconf2-common (3.2.6-4ubuntu1) ...

Creating config file /etc/gconf/2/path with new version
Setting up libgtk2.0-common (2.24.32-1ubuntu1) ...
Setting up libdbus-glib-1-2:amd64 (0.110-2) ...
Setting up xvfb (2:1.19.6-1ubuntu4.9) ...
Setting up libgconf-2-4:amd64 (3.2.6-4ubuntu1) ...
Setting up libgtk2.0-0:amd64 (2.24.32-1ubuntu1) ...
Setting up libgail18:amd64 (2.24.32-1ubuntu1) ...
Setting up libgail-common:amd64 (2.24.32-1ubuntu1) ...
Setting up libgtk2.0-bin (2.24.32-1ubuntu1) ...
Setting up gconf-service-backend (3.2.6-4ubuntu1) ...
Setting up gconf-service (3.2.6-4ubuntu1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
/sbin/ldconfig.real: /usr/local/lib/python3.7/dist-
packages/ideep4py/lib/libmkldnn.so.0 is not a symbolic link

Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

[3]:
```
# data is of netflix from date-(1-aug-2003)_to_(28-aug-2020) from yahoo finance
df = pd.read_csv('case_time_series (4).csv', header=0)
df = df.sort_index(ascending=True, axis=0)
df
```

[3]:

	Date	Date_YMD	...	Daily Deceased	Total Deceased
0	30 January 2020	2020-01-30	...	0	0
1	31 January 2020	2020-01-31	...	0	0
2	1 February 2020	2020-02-01	...	0	0
3	2 February 2020	2020-02-02	...	0	0
4	3 February 2020	2020-02-03	...	0	0
..	...	...	...	...	...
472	16 May 2021	2021-05-16	...	4098	273826
473	17 May 2021	2021-05-17	...	4334	278160
474	18 May 2021	2021-05-18	...	4529	282689
475	19 May 2021	2021-05-19	...	3877	286566
476	20 May 2021	2021-05-20	...	4208	290774

[477 rows x 8 columns]

[4]:
```
df.drop(['Total Confirmed','Total Recovered','Total Deceased'], axis=1,␣
↪inplace=True)
```

[5]:
```
df.head()
```

[5]:

	Date	Date_YMD	...	Daily Recovered	Daily Deceased
0	30 January 2020	2020-01-30	...	0	0
1	31 January 2020	2020-01-31	...	0	0
2	1 February 2020	2020-02-01	...	0	0
3	2 February 2020	2020-02-02	...	0	0

```
4  3 February 2020  2020-02-03  ...                 0                 0

[5 rows x 5 columns]
```

[6]:
```
df  = df[df['Date_YMD']>'2020-03-18']
df.reset_index(drop=True, inplace=True)
df.head()
```

[6]:

	Date	Date_YMD	Daily Confirmed	Daily Recovered	Daily Deceased
0	19 March 2020	2020-03-19	27	5	1
1	20 March 2020	2020-03-20	58	3	0
2	21 March 2020	2020-03-21	78	0	0
3	22 March 2020	2020-03-22	69	0	3
4	23 March 2020	2020-03-23	94	2	2

[7]:
```
df.shape
```

[7]:
```
(428, 5)
```

[8]:
```
df.describe()
```

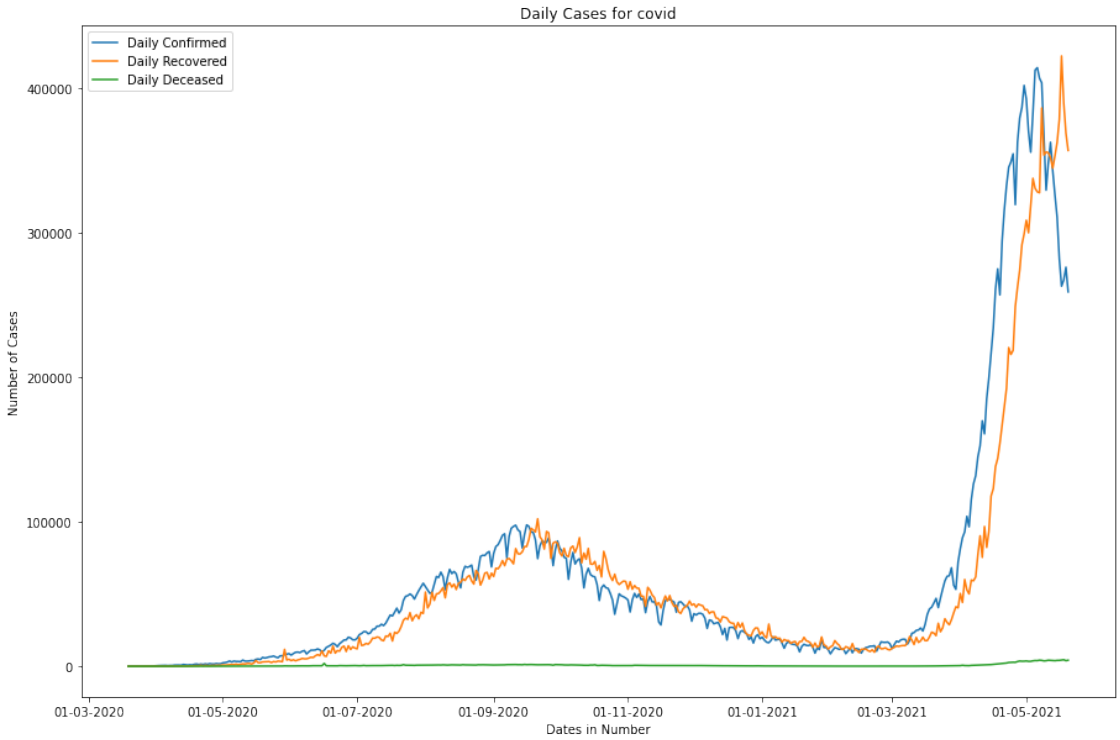
[8]:

	Daily Confirmed	Daily Recovered	Daily Deceased
count	428.000000	428.000000	428.000000
mean	60818.425234	53050.595794	679.371495
std	89594.173976	79248.069067	904.680613
min	27.000000	0.000000	0.000000
25%	12182.750000	11750.750000	137.750000
50%	29664.000000	26810.500000	419.000000
75%	65057.000000	60149.750000	854.000000
max	414280.000000	422391.000000	4529.000000

[9]:
```
from matplotlib.dates import DateFormatter
import matplotlib.dates as mdates

df["Date_YMD"]= pd.to_datetime(df["Date_YMD"])

plt.figure(figsize = (15,10))
plt.plot(df['Date_YMD'], df['Daily Confirmed'], label='Daily Confirmed')
plt.plot(df['Date_YMD'], df['Daily Recovered'], label='Daily Recovered')
plt.plot(df['Date_YMD'], df['Daily Deceased'], label='Daily Deceased')
plt.xlabel("Dates in Number")
plt.ylabel("Number of Cases")
plt.title("Daily Cases for covid")
myFmt = mdates.DateFormatter("%d-%m-%Y")
plt.gca().xaxis.set_major_formatter(myFmt)
plt.legend()
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.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
```



[10]:
```
df.isnull().any()
```

[10]:
```
Date                False
Date_YMD            False
Daily Confirmed     False
Daily Recovered     False
Daily Deceased      False
dtype: bool
```

[11]:
```
fig = px.line(df, x='Date', y='Daily Confirmed')
fig.show()
```

[12]:
```
fig = px.line(df, x=df['Date'], y='Daily Confirmed',title='cases in India')
fig.add_scatter(x=df['Date'], y=df['Daily Recovered'],mode='lines')
fig.show()
```

[13]:
```
pip install -U kaleido
```

```
Collecting kaleido
  Downloading https://files.pythonhosted.org/packages/ae/b3/a0f0f4faac229b
0011d8c4a7ee6da7c2dca0b6fd08039c95920846f23ca4/kaleido-0.2.1-py2.py3-none-
manylinux1_x86_64.whl (79.9MB)
     || 79.9MB 49kB/s
Installing collected packages: kaleido
Successfully installed kaleido-0.2.1
```

[14]:
```
if not os.path.exists("images"):
    os.mkdir("images")
```

[15]:
```
crd_data = df.iloc[:, 2:5]
crd_avg = crd_data.mean(axis=1)
cr_avg = df[['Daily Confirmed', 'Daily Recovered']].mean(axis=1)
close = df['Daily Deceased']
```

[16]:
```
fig1 = go.Figure()
fig1.add_trace(go.Scatter(x = df.index, y = crd_avg,name='CRD avg'))
fig1.add_trace(go.Scatter(x = df.index, y = cr_avg,name='CR avg'))
fig1.add_trace(go.Scatter(x = df.index, y = close,name='deaths column data'))
fig1.show()
```

[17]:
```
fig1.write_image("diff_btwn_diff_avgs.png")
```

[18]:
```
# we will create a new df which has only 2 column which is useful to predict␣
 ↪data
new_data = pd.DataFrame(index=range(0,len(df)), columns=['Date', 'crd_avg'])
for i in range(0, len(df)):
  new_data['Date'][i] = df['Date'][i]
  new_data['crd_avg'][i] = cr_avg[i]
```

[19]:
```
new_data.head()
```

[19]:

	Date	crd_avg
0	19 March 2020	16
1	20 March 2020	30.5
2	21 March 2020	39
3	22 March 2020	34.5
4	23 March 2020	48

[20]:
```
# setting index
new_data.index = new_data.Date
new_data.drop('Date', axis=1, inplace=True)
```

[21]:
```
print(len(new_data))
```

```
428
```

[22]:
```
ds = new_data.values
```

[23]:
```
# we will take 80% data in train and remaining in test
train = int(len(new_data)*0.8)
test = len(new_data) - train
train, test = new_data.iloc[0:train,:], new_data.iloc[train:len(new_data),:]
```

[24]:
```
train.shape
```

[24]:
```
(342, 1)
```

[25]:
```
test.shape
```

[25]:
```
(86, 1)
```

[26]:
```
# we have normalize the data cuz data is like 149...., 488..something like that
# so we have to normalize betwwen 0 and 1
scalar = MinMaxScaler(feature_range=(0, 1))
scaled_data = scalar.fit_transform(ds)
```

[27]:
```
# splitting the data to x_train, y_train
# we will first train upto 60 and then predict on 61 and then
# we will train from 61 to 120 then predict on 121 likewise we will go
x_train, y_train = [], []
for i in range(60, len(train)):
  x_train.append(scaled_data[i-60:i,0])
  y_train.append(scaled_data[i,0])

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

[28]:
```
# now we have reshape the array to 3-d to pass the data into lstm [number of␣
 ↪samples, time steps/batch_size, features]
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
```

[29]:
```
x_train.shape
```

[29]:
```
(282, 60, 1)
```

[30]:
```
model = Sequential()
model.add(LSTM(units=100, return_sequences=True, input_shape=(x_train.shape[1],␣
 ↪1)))
model.add(Dropout(0.25))
model.add(LSTM(units=50))
model.add(Dense(1))
model.add(Activation('relu'))
model.compile(loss='mean_squared_error', optimizer='adam')
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 60, 100)           40800
_________________________________________________________________
```

```
dropout (Dropout)            (None, 60, 100)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 50)                30200
_________________________________________________________________
dense (Dense)                (None, 1)                 51
_________________________________________________________________
activation (Activation)      (None, 1)                 0
=================================================================
Total params: 71,051
Trainable params: 71,051
Non-trainable params: 0
_________________________________________________________________
None
```

[31]:
```
model.fit(x_train, y_train, epochs=100, batch_size=32, verbose=1)
```

```
Epoch 1/100
9/9 [==============================] - 4s 72ms/step - loss: 0.0136
Epoch 2/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0133
Epoch 3/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0132
Epoch 4/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0127
Epoch 5/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0137
Epoch 6/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0125
Epoch 7/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0141
Epoch 8/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0136
Epoch 9/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0128
Epoch 10/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0148
Epoch 11/100
9/9 [==============================] - 1s 75ms/step - loss: 0.0133
Epoch 12/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0135
Epoch 13/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0138
Epoch 14/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0128
Epoch 15/100
9/9 [==============================] - 1s 68ms/step - loss: 0.0140
Epoch 16/100
```

```
9/9 [==============================] - 1s 71ms/step - loss: 0.0138
Epoch 17/100
9/9 [==============================] - 1s 77ms/step - loss: 0.0133
Epoch 18/100
9/9 [==============================] - 1s 77ms/step - loss: 0.0136
Epoch 19/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0140
Epoch 20/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0133
Epoch 21/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0147
Epoch 22/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0131
Epoch 23/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0138
Epoch 24/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0135
Epoch 25/100
9/9 [==============================] - 1s 69ms/step - loss: 0.0129
Epoch 26/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0135
Epoch 27/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0142
Epoch 28/100
9/9 [==============================] - 1s 69ms/step - loss: 0.0146
Epoch 29/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0142
Epoch 30/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0136
Epoch 31/100
9/9 [==============================] - 1s 69ms/step - loss: 0.0135
Epoch 32/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0143
Epoch 33/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0147
Epoch 34/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0123
Epoch 35/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0130
Epoch 36/100
9/9 [==============================] - 1s 68ms/step - loss: 0.0131
Epoch 37/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0147
Epoch 38/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0126
Epoch 39/100
9/9 [==============================] - 1s 69ms/step - loss: 0.0130
Epoch 40/100
```

```
9/9 [==============================] - 1s 72ms/step - loss: 0.0146
Epoch 41/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0139
Epoch 42/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0141
Epoch 43/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0155
Epoch 44/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0147
Epoch 45/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0136
Epoch 46/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0138
Epoch 47/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0134
Epoch 48/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0127
Epoch 49/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0129
Epoch 50/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0137
Epoch 51/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0143
Epoch 52/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0126
Epoch 53/100
9/9 [==============================] - 1s 70ms/step - loss: 0.0127
Epoch 54/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0126
Epoch 55/100
9/9 [==============================] - 1s 76ms/step - loss: 0.0142
Epoch 56/100
9/9 [==============================] - 1s 77ms/step - loss: 0.0132
Epoch 57/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0146
Epoch 58/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0130
Epoch 59/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0140
Epoch 60/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0142
Epoch 61/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0147
Epoch 62/100
9/9 [==============================] - 1s 71ms/step - loss: 0.0140
Epoch 63/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0124
Epoch 64/100
```

```
9/9 [==============================] - 1s 76ms/step - loss: 0.0141
Epoch 65/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0143
Epoch 66/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0128
Epoch 67/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0138
Epoch 68/100
9/9 [==============================] - 1s 74ms/step - loss: 0.0140
Epoch 69/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0133
Epoch 70/100
9/9 [==============================] - 1s 75ms/step - loss: 0.0136
Epoch 71/100
9/9 [==============================] - 1s 76ms/step - loss: 0.0137
Epoch 72/100
9/9 [==============================] - 1s 78ms/step - loss: 0.0138
Epoch 73/100
9/9 [==============================] - 1s 81ms/step - loss: 0.0133
Epoch 74/100
9/9 [==============================] - 1s 76ms/step - loss: 0.0133
Epoch 75/100
9/9 [==============================] - 1s 74ms/step - loss: 0.0123
Epoch 76/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0133
Epoch 77/100
9/9 [==============================] - 1s 74ms/step - loss: 0.0134
Epoch 78/100
9/9 [==============================] - 1s 80ms/step - loss: 0.0132
Epoch 79/100
9/9 [==============================] - 1s 72ms/step - loss: 0.0144
Epoch 80/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0137
Epoch 81/100
9/9 [==============================] - 1s 74ms/step - loss: 0.0135
Epoch 82/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0137
Epoch 83/100
9/9 [==============================] - 1s 78ms/step - loss: 0.0141
Epoch 84/100
9/9 [==============================] - 1s 74ms/step - loss: 0.0136
Epoch 85/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0134
Epoch 86/100
9/9 [==============================] - 1s 73ms/step - loss: 0.0138
Epoch 87/100
9/9 [==============================] - 1s 76ms/step - loss: 0.0137
Epoch 88/100
```

```
9/9 [==============================] - 1s 77ms/step - loss: 0.0145
Epoch 89/100
9/9 [==============================] - 1s 83ms/step - loss: 0.0129
Epoch 90/100
9/9 [==============================] - 1s 82ms/step - loss: 0.0129
Epoch 91/100
9/9 [==============================] - 1s 78ms/step - loss: 0.0139
Epoch 92/100
9/9 [==============================] - 1s 80ms/step - loss: 0.0135
Epoch 93/100
9/9 [==============================] - 1s 82ms/step - loss: 0.0149
Epoch 94/100
9/9 [==============================] - 1s 78ms/step - loss: 0.0148
Epoch 95/100
9/9 [==============================] - 1s 75ms/step - loss: 0.0141
Epoch 96/100
9/9 [==============================] - 1s 77ms/step - loss: 0.0136
Epoch 97/100
9/9 [==============================] - 1s 77ms/step - loss: 0.0129
Epoch 98/100
9/9 [==============================] - 1s 77ms/step - loss: 0.0135
Epoch 99/100
9/9 [==============================] - 1s 77ms/step - loss: 0.0139
Epoch 100/100
9/9 [==============================] - 1s 80ms/step - loss: 0.0136
```

[31]:
```
<tensorflow.python.keras.callbacks.History at 0x7fa7ce3a8450>
```

[32]:
```
# predicting 920 values, using past 60 from the train data
inputs = new_data[len(new_data)-len(test) - 60:].values
inputs = inputs.reshape(-1,1)
inputs = scalar.transform(inputs)
```

[33]:
```
inputs.shape
```

[33]:
```
(146, 1)
```

[34]:
```
x_test = []
for i in range(60,inputs.shape[0]):
    x_test.append(inputs[i-60:i,0])
x_test = np.array(x_test)
```

[35]:
```
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
```

[36]:
```
predicted_price = model.predict(x_test)
# inverse transform for getting back all normal values from scaled values
predicted_price = scalar.inverse_transform(predicted_price)
```

[37]:
```
rms=np.sqrt(np.mean(np.power((test-predicted_price),2)))
rms
```

[37]:
```
crd_avg    208559.349232
dtype: float64
```

[38]:
```
# create a new column of predicted values
test['Prediction'] = predicted_price
test.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2:
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
```

[38]:

Date	crd_avg	Prediction
24 February 2021	14515	15.999999
25 February 2021	14410.5	15.999999
26 February 2021	14676	15.999999
27 February 2021	14257	15.999999
28 February 2021	13452.5	15.999999

[39]:
```
test.tail()
```

[39]:

Date	crd_avg	Prediction
16 May 2021	330182	15.999999
17 May 2021	342706	15.999999
18 May 2021	328502	15.999999
19 May 2021	322599	15.999999
20 May 2021	308068	15.999999

[40]:
```
# Graph for comparing the results of model predicted and original value
fig2 = go.Figure()
fig2.add_trace(go.Scatter(x = train.index, y = train.crd_avg,name='train'))
fig2.add_trace(go.Scatter(x = test.index, y = test.crd_avg,name='test_crd_avg'))
fig2.add_trace(go.Scatter(x = test.index, y = test.Prediction,name='test'))
fig2.show()
```

[ ]:
```
%%capture
!wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
from colab_pdf import colab_pdf
colab_pdf('Untitled33.ipynb')
```

[ ]: