**COVID FORECASTING ANALYSIS USING RNN**

**SOURCE CODE:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

country = "US"

#country = 'India'

**#Total COVID confirmed cases**

df\_confirmed = pd.read\_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid19\_confirmed\_global.csv")

#df\_confirmed.to\_csv('global.csv')

df\_confirmed\_country = df\_confirmed[df\_confirmed["Country/Region"] == country]

df\_confirmed\_country = pd.DataFrame(df\_confirmed\_country[df\_confirmed\_country.columns[4:]].sum(),columns=["confirmed"])

df\_confirmed\_country.index = pd.to\_datetime(df\_confirmed\_country.index,format='%m/%d/%y')

df\_confirmed\_country.plot(figsize=(10,5),title="COVID confirmed cases")

df\_confirmed\_country.tail(10)

print("Total days in the dataset", len(df\_confirmed\_country))

**#Use data until 14 days before as training**

x = len(df\_confirmed\_country)-14

train=df\_confirmed\_country.iloc[:x]

test = df\_confirmed\_country.iloc[x:]

**##scale or normalize data as the data is too skewed**

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

scaler.fit(train)

train\_scaled = scaler.transform(train)

test\_scaled = scaler.transform(test)

from keras.preprocessing.sequence import TimeseriesGenerator

**#Sequence size has an impact on prediction, especially since COVID is unpredictable!**

seq\_size = 7 ## number of steps (lookback)

n\_features = 1 ## number of features. This dataset is univariate so it is 1

train\_generator = TimeseriesGenerator(train\_scaled, train\_scaled, length = seq\_size, batch\_size=1)

print("Total number of samples in the original training data = ", len(train)) # 271

print("Total number of samples in the generated data = ", len(train\_generator)) # 264 with seq\_size=7

**#Check data shape from generator**

x,y = train\_generator[10] #Check train\_generator

#Takes 7 days as x and 8th day as y (for seq\_size=7)

**#Also generate test data**

test\_generator = TimeseriesGenerator(test\_scaled, test\_scaled, length=seq\_size, batch\_size=1)

print("Total number of samples in the original training data = ", len(test)) # 14 as we're using last 14 days for test

print("Total number of samples in the generated data = ", len(test\_generator)) # 7

**#Check data shape from generator**

x,y = test\_generator[0]

from keras.models import Sequential

from keras.layers import Dense, LSTM, Dropout, Activation

**#Define Model**

model = Sequential()

model.add(LSTM(150, activation='relu', return\_sequences=True, input\_shape=(seq\_size, n\_features)))

model.add(LSTM(64, activation='relu'))

model.add(Dense(64))

model.add(Dense(1))

model.compile(optimizer='adam', loss='mean\_squared\_error')

model.summary()

print('Train...')

history = model.fit\_generator(train\_generator,

validation\_data=test\_generator,

epochs=50, steps\_per\_epoch=10)