import pandas as pd

import numpy as np

import seaborn as sns

import glob

import matplotlib.pyplot as plt

import matplotlib as mpl

mpl.matplotlib\_fname()

import plotly.graph\_objects as go

import matplotlib.dates as mdates

from matplotlib.dates import DateFormatter

import seaborn as sns

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import mean\_squared\_error

from keras.models import Sequential

from keras.layers import Dense, LSTM, Dropout

from datetime import date

from datetime import datetime as dt

import datetime

import random

DT = pd.concat(map(pd.read\_excel, glob.glob('data/\*.xls')))

DT = DT.sort\_values(by='Tgl Pendaftaran', ascending=True)

DT = DT.reset\_index(drop=True)

DT

medic\_rec = DT.loc[:, ['Tgl Pendaftaran','No Identitas Pasien','Tanggal Lahir','Nama Ruangan','Jenis Kelamin']]

medic\_rec.rename(columns = {'Tgl Pendaftaran':'Tanggal Pendaftaran','No Identitas Pasien':'NIK','Tanggal Lahir':'Tanggal Lahir','Nama Ruangan':'Penyakit'}, inplace = True)

medic\_rec['Tanggal Pendaftaran'] = medic\_rec['Tanggal Pendaftaran'].dt.strftime('%Y-%m-%d')

medic\_rec['Tanggal Pendaftaran'] = pd.to\_datetime(medic\_rec['Tanggal Pendaftaran'], format='%Y-%m-%d')

medic\_rec

daftar\_penyakit = [

"Klinik Penyakit Dalam","Klinik Penyakit Syaraf",

"Klinik Orthopedi","Klinik Penyakit Paru","Klinik Penyakit Jantung",

"Klinik Anak","Klinik THT","Klinik Urology","Klinik Bedah Umum",

"Klinik Kebidanan dan Penyakit Kandungan","Klinik Gigi dan Mulut",

"Klinik Penyakit Kulit dan Kelamin","Klinik Kesehatan Jiwa","Klinik Reumatologi",

"Klinik Bedah Onkology","Klinik Penyakit Mata","Klinik Endokrin",

"Klinik Gastroentro Hepatologi","Klinik Bedah Digestif","Klinik Onkology Kebidanan",

"Klinik VCT","Klinik Bedah Plastik","Klinik Ginjal dan Hipertensi"

]

daftar\_penyakit\_eng = [

'Internal Disease Clinic', 'Nervous Disease Clinic', 'Orthopedic Clinic',

'Pulmonary Disease Clinic', 'Heart Disease Clinic', "Children's Clinic",

'THT Clinic', 'Urology Clinic', 'General Surgery Clinic', 'Obstetrics and Gynecology Clinic',

'Dental and Oral Clinic', 'Skin and Genital Disease Clinic', 'Mental Health Clinic',

'Rheumatology Clinic', 'Surgical Oncology Clinic', 'Eye Disease Clinic', 'Endocrine Clinic',

'Gastroenterology Hepatology Clinic', 'Digestive Surgery Clinic', 'Obstetric Oncology Clinic',

'VCT Clinic', 'Plastic Surgery Clinic', 'Kidney and Hypertension Clinic'

]

warna\_penyakit = ['#63EA71',

'#CE3F0D',

'#ACAD5E',

'#F07843',

'#BE7FD8',

'#B596F3',

'#AF6AC7',

'#0AE9CC',

'#8F3032',

'#D84C0B',

'#EA37BE',

'#6D0F01',

'#814B13',

'#7AC82A',

'#81B48F',

'#01B465',

'#622F24',

'#A57A48',

'#B5AED8',

'#F9B227',

'#AFFE30',

'#EE211F',

'#4D1737']

# Total

data\_penyakit\_final = []

data\_tmp = []

tahun = pd.date\_range("2018-01-01", periods=5, freq="Y")

nama\_penyakit = daftar\_penyakit

data\_tmp = medic\_rec

# Looping per Nama Penyakit

y=0

while y < len(nama\_penyakit) :

data\_tmp.loc[data\_tmp.Penyakit == nama\_penyakit[y], nama\_penyakit[y]] = 'True'

y+=1

data\_tmp = pd.DataFrame(data\_tmp.groupby('Tanggal Pendaftaran', as\_index=False)[nama\_penyakit].count())

data\_penyakit\_final=pd.DataFrame(data\_tmp)

data\_train = pd.to\_datetime(data\_penyakit\_final['Tanggal Pendaftaran'])

data\_penyakit\_final = data\_penyakit\_final.iloc[:-1,:]

data\_training = []

y\_pred\_future = []

i=1

while i <= len(daftar\_penyakit) :

cols = list(data\_penyakit\_final)[i:i+1]

data\_training = data\_penyakit\_final[cols].astype(float)

scaler = StandardScaler()

scaler = scaler.fit(data\_training)

data\_training\_scaled = scaler.transform(data\_training)

trainX = []

trainY = []

n\_future = 0

n\_past = 14

for y in range(n\_past, len(data\_training\_scaled) - n\_future +1):

trainX.append(data\_training\_scaled[y - n\_past:y, 0:data\_training.shape[1]])

trainY.append(data\_training\_scaled[y + n\_future - 1:y + n\_future, 0])

trainX, trainY = np.array(trainX), np.array(trainY)

model = Sequential()

model.add(LSTM(64, activation='relu', input\_shape=(trainX.shape[1], trainX.shape[2]), return\_sequences=True))

model.add(LSTM(32, activation='relu', return\_sequences=False))

model.add(Dropout(0.2))

model.add(Dense(trainY.shape[1]))

model.compile(optimizer='adam', loss='mse' )

model.summary()

history = model.fit(trainX, trainY, epochs=12, batch\_size=16, validation\_split=0.1, verbose=1)

n\_future=730

forecast\_periode\_dates = pd.date\_range(list(data\_train)[-1], periods=n\_future, freq='1D').tolist()

forecast = model.predict(trainX[-n\_future:])

forecast\_copies = np.repeat(forecast, data\_training.shape[1], axis=1)

y\_pred\_future.append(scaler.inverse\_transform(forecast\_copies)[:,0])

i+=1

tes = pd.DataFrame(y\_pred\_future)

tes = tes.T

tes.columns = daftar\_penyakit

forecast\_dates = []

for time\_i in forecast\_periode\_dates:

forecast\_dates.append(time\_i.date())

new\_col = forecast\_dates

tes.insert(loc=0, column='Tanggal Pendaftaran', value=new\_col)

data\_forecast = tes

data\_forecast['Tanggal Pendaftaran'] = pd.to\_datetime(data\_forecast['Tanggal Pendaftaran'])

data\_forecast = data\_forecast.groupby(pd.Grouper(key='Tanggal Pendaftaran', freq='1M')).sum().reset\_index()

# data\_actual = data\_penyakit\_final.groupby(pd.Grouper(key='Tanggal Pendaftaran', freq='1M')).sum().reset\_index()

actual = data\_penyakit\_final

# new\_col = data\_penyakit\_final['Tanggal Pendaftaran']

# actual.insert(loc=0, column='Date', value=new\_col)

actual = actual.groupby(pd.Grouper(key='Tanggal Pendaftaran', freq='1M')).sum().reset\_index()

actual['Tanggal Pendaftaran'] = actual['Tanggal Pendaftaran'].dt.strftime('%b-%y')

actual['Tanggal Pendaftaran'] = pd.to\_datetime(actual['Tanggal Pendaftaran'], format='%b-%y')

actual = actual.iloc[:-1,:]

data\_forecast.loc[[0],:] = list(actual.loc[len(actual)-1])

data\_forecast = data\_forecast.iloc[:-1,:]

fig = plt.figure(figsize=(20,10), facecolor='white')

graph = plt.subplot(111)

# plt.plot(actual['Date'], actual['Klinik Penyakit Dalam'], color='blue', label='Data Sebenarnya',linewidth=1.5)

y=0

while y < len(nama\_penyakit) :

plt.plot(actual['Tanggal Pendaftaran'], actual[nama\_penyakit[y]], color=warna\_penyakit[y], linewidth=1.5)

y+=1

y=0

while y < len(nama\_penyakit) :

plt.plot(data\_forecast['Tanggal Pendaftaran'], data\_forecast[nama\_penyakit[y]], color=warna\_penyakit[y], label=daftar\_penyakit\_eng[y],linewidth=1.5)

y+=1

# plt.xlabel('Year', fontsize=28)

# plt.ylabel('Number of Patient (person)', fontsize=28)

plt.xlabel('Tahun', fontsize=28)

plt.ylabel('Jumlah Pasien (orang)', fontsize=28)

plt.xticks(fontsize=26)

plt.yticks(fontsize=26)

leg = plt.legend (loc=1, fontsize=12);

frame = leg.get\_frame()

frame.set\_facecolor('white')

ax = plt.gca()

ax.set\_facecolor("white")

ax.spines['bottom'].set\_color('0.5')

ax.spines['top'].set\_color('0.5')

ax.spines['right'].set\_color('0.5')

ax.spines['left'].set\_color('0.5')

graph.set\_xlim([datetime.date(2018, 1, 1), datetime.date(2024, 7, 1)])

# plt.ylim(-10,100)

graph.xaxis.set\_major\_locator(mdates.MonthLocator(interval=6))

graph.xaxis.set\_major\_formatter(DateFormatter("%b-%y"))

plt.savefig("grafikeng/grafik\_prediksi/Prediksi 2.jpg", dpi=300)

plt.show()