```
# -*- coding: utf-8 -*-
"""
Created on Sat May  9 23:45:11 2020

@author: user
"""

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import random
import math
import time
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, mean_absolute_error
import datetime
import operator
import seaborn as sns
#modul supervised learning python
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from datetime import datetime,timedelta
from sklearn.metrics import mean_squared_error
from scipy.optimize import curve_fit
from scipy.optimize import fsolve

#dataset
data_province = pd.read_csv('province_data.csv')
data_regional = pd.read_csv('regional_data.csv')

#pemisahan dataframe per region code
total1 = data_regional[(data_regional.Region_Code == 1)]
total2 = data_regional[(data_regional.Region_Code == 2)]
total3 = data_regional[(data_regional.Region_Code == 3)]
total4 = data_regional[(data_regional.Region_Code == 4)]
total5 = data_regional[(data_regional.Region_Code == 5)]
total6 = data_regional[(data_regional.Region_Code == 6)]
total7 = data_regional[(data_regional.Region_Code == 7)]
total8 = data_regional[(data_regional.Region_Code == 8)]
total9 = data_regional[(data_regional.Region_Code == 9)]
total10 = data_regional[(data_regional.Region_Code == 10)]
total11 = data_regional[(data_regional.Region_Code == 11)]
total12 = data_regional[(data_regional.Region_Code == 12)]
total13 = data_regional[(data_regional.Region_Code == 13)]
```

```
total14 = data_regional[(data_regional.Region_Code == 14)]
total15 = data_regional[(data_regional.Region_Code == 15)]
total16 = data_regional[(data_regional.Region_Code == 16)]
total17 = data_regional[(data_regional.Region_Code == 17)]
total18 = data_regional[(data_regional.Region_Code == 18)]
total19 = data_regional[(data_regional.Region_Code == 19)]
total20 = data_regional[(data_regional.Region_Code == 20)]


#data frame untuk mengetahui rata-rata penambahan kasus
total1_total_cases1 = total1.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
total2_total_cases2 = total2.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
total3_total_cases3 = total3.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
total4_total_cases4 = total4.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
total5_total_cases5 = total5.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
total6_total_cases6 = total6.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
total7_total_cases7 = total7.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
total8_total_cases8 = total8.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
total9_total_cases9 = total9.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
total10_total_cases10 = total10.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
total11_total_cases11 = total11.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
total12_total_cases12 = total12.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()
```

```
total13_total_cases13 =
total13.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()

total14_total_cases14 =
total14.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()

total15_total_cases15 =
total15.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()

total16_total_cases16 =
total16.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()

total17_total_cases17 =
total17.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()

total18_total_cases18 =
total18.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()

total19_total_cases19 =
total19.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()

total20_total_cases20 =
total20.groupby(['Date'])['Total_Cases','New_Actually_Positive'].sum().reset_index()


total1_total_cases1["New_Actually_Positive"].mean()

total2_total_cases2["New_Actually_Positive"].mean()

total3_total_cases3["New_Actually_Positive"].mean()

total4_total_cases4["New_Actually_Positive"].mean()

total5_total_cases5["New_Actually_Positive"].mean()

total6_total_cases6["New_Actually_Positive"].mean()

total7_total_cases7["New_Actually_Positive"].mean()

total8_total_cases8["New_Actually_Positive"].mean()

total9_total_cases9["New_Actually_Positive"].mean()

total10_total_cases10["New_Actually_Positive"].mean()

total11_total_cases11["New_Actually_Positive"].mean()

total12_total_cases12["New_Actually_Positive"].mean()

total13_total_cases13["New_Actually_Positive"].mean()

total14_total_cases14["New_Actually_Positive"].mean()

total15_total_cases15["New_Actually_Positive"].mean()
```

```
total16_total_cases16["New_Actually_Positive"].mean()
total17_total_cases17["New_Actually_Positive"].mean()
total18_total_cases18["New_Actually_Positive"].mean()
total19_total_cases19["New_Actually_Positive"].mean()
total20_total_cases20["New_Actually_Positive"].mean()

#dataframe eksternal, data Indonesia, sumber Kawal COVID-19
data_indonesia = pd.read_csv('Kasus Harian Ina.csv')
data_indonesia["Kasus Baru"].mean()

#gunakan data yang ke 9 untuk memprediksi corona di indonesia
total9_total_cases = total9.groupby(['Date'])['Total_Cases'].sum().reset_index()

#logistic regression
FMT = '%Y-%m-%d %H:%M:%S'
date = total9_total_cases['Date']
total9_total_cases['Date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime('2020-02-24 18:00:00' , FMT)).days)

def logistic_model(x,a,b,c) :
   return c/(1+np.exp(-(x-b)/a))

x = list(total9_total_cases.iloc[1:,0])
y = list(total9_total_cases.iloc[1:,1])
fit = curve_fit(logistic_model,x,y)

A,B = fit
A
```

```
errors = [np.sqrt(fit[1][i][i]) for i in [0,1,2]]
errors

a=A[0]+errors[0]
b=A[1]+errors[1]
c=A[2]+errors[2]

#jumlah hari maksimal terhitung dari 24 februari
sol = int(fsolve(lambda x : logistic_model(x,a,b,c) - int(c),b))
sol

#pembuatan kurva
pred_x = list(range(max(x),sol))
plt.rcParams['figure.figsize']=[7,7]
plt.rc('font', size=14)
plt.scatter(x,y,label='Real data', color='red')
#logistic curve
plt.plot(x+pred_x, [logistic_model(i,a,b,c) for i in x+pred_x], label='Logistic model')

plt.legend()
plt.xlabel('Days since 24 feb 2020')
plt.ylabel('Total cases')
plt.ylim((min(y)*0.9,c*1.1))
plt.show()

#mencari mse dan r^2
y_pred_logistic = [logistic_model(i,a,b,c) for i in x]
p=mean_squared_error(y,y_pred_logistic)
```

```
s1=(np.subtract(y,y_pred_logistic)**2).sum()

s2=(np.subtract(y,np.mean(y))**2).sum()

r=1-s1/s2

print('R^2 {}'.format(r))

print('Mean square error {}'.format(p))


print('Puncak wabah adalah {:.0f} hari setelah 24 feb 2020 {}'. format(sol,x))
```