COVID-19 Epidemic Analysis based on clinical and user mobility data with Machine and Deep Learning Algorithms

The coronavirus is a family of viruses that take their name from their spiked crown (they form a "crown" in their shell). Coronavirus, also known as SARS-CoV-2, is a respiratory virus that was first reported, in Wuhan China. On 11/2/2020, the World Health Organization (WHO) named it COVID-19 for the disease caused by the new coronavirus.

This work for the "Decision Theory" class aims at the evaluation of COVID-19 through data analysis and projections using the mathematical theories taught in the lectures and all the well-known tools of the machine learning area.

The work is developing into two parts. The first part strand on predicting the dispersion of confirmed cases, and deaths, in a time frame of 10 days after the referenced day, based on the prediction models which will be trained, for specific characteristics the researcher will choose. The second part looks forward to the connection of citizens' mobility through the mobility date, with the existing and projected dispersion.

Date are as follows:

COVID-19 Case Data provided by Johns Hopkins University
Mobility data provided by Apple

Part 1; Following Steps:

1) Useful Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import pandas as pd
import random
import math
import time
from sklearn.linear_model import LinearRegression, BayesianRidge
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, mean_absolute_error
import datetime
import operator
plt.style.use('fivethirtyeight')
%matplotlib inline
from IPython.display import set_matplotlib_formats
set_matplotlib_formats('retina')
import warnings
warnings.filterwarnings("ignore")
```

2) Import the data (make sure you update this on a daily basis)

```
A) For example:

confirmed_df = 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')

#same for deaths, recoveries, latest_data, us_medical_data(best known
data), apple_mobility
```

```
B) Some prints(shows) as: latest_data.head()

#same for confirmed, us_medical
```

3) Get all the dates for the ongoing coronavirus pandemic

(You may use `cols = confirmed_df.keys()`)

You probably need a for loop for confirmed_sum (as for death and recovered). Also you must print the confirmed, deaths, recovered, and active. Compute the rates death_sum/confirmed_sum and recovered_sum/confirmed_sum

4) Getting daily increases and moving averages

You probably need a for loop for daily increase and for moving average. Also setup the window size and set the daily_confirmed cases through world daily cases. Do the same for avg and moving avg

You must repeat the steps above for deaths, recoveries and active cases

5) Future forecasting

This is all yours

6) Convert integer into daytime for better visualization.

This part there's because of googleness

```
start = '1/22/2020'
start_date = datetime.datetime.strptime(start, '%m/%d/%Y')
future_forcast_dates = []
for i in range(len(future_forcast)):
                         future_forcast_dates.append((start_date       +
datetime.timedelta(days=i)).strftime('%m/%d/%Y'))
```

```
# slightly modify the data to fit the model better (regression models
cannot pick the pattern)
X_train_confirmed, X_test_confirmed, y_train_confirmed, y_test_confirmed
=         train_test_split(days_since_1_22[50:],         world_cases[50:],
test_size=0.05, shuffle=False)
```

7) Model for predicting # of confirmed cases.

You can do this using support vector machine, bayesian ridge , and linear regression. You must show the results in the later section.

```
# Great hint
# # use this to find the optimal parameters for SVR
# c = [0.01, 0.1, 1]
# gamma = [0.01, 0.1, 1]
# epsilon = [0.01, 0.1, 1]
# shrinking = [True, False]

# svm_grid = {'C': c, 'gamma' : gamma, 'epsilon': epsilon, 'shrinking' : shrinking}

# svm = SVR(kernel='poly', degree=3)
#        svm_search        =        RandomizedSearchCV(svm,        svm_grid, scoring='neg_mean_squared_error',      cv=3,      return_train_score=True, n_jobs=-1, n_iter=30, verbose=1)
# svm_search.fit(X_train_confirmed, y_train_confirmed)
```

```
A) # check against testing data
[]:
```

```
B) # transform our data for polynomial regression
[]:
```

```
C) # polynomial regression
[]:
```

```
D) # bayesian ridge polynomial regression
[]:
```

8) Worldwide Overview

Graphing the number of confirmed cases, active cases, deaths, recoveries, mortality rate (CFR), and recovery rate worldwide. This gives you a big picture of the ongoing pandemic.

```
# helper method for flattening the data, so it can be displayed
on a bar graph such as

def flatten(arr):
    a = []
    arr = arr.tolist()
    for i in arr:
        a.append(i[0])
    return a
```

[]:

9) Country Specific Graphs

Unlike the previous section, we are taking a look at specific countries. This allows us to examine the pandemic at a local level.

[]:

```
# Country Comparison  # removed redundant code
```

[]:

10) Predictions for confirmed coronavirus cases worldwide

These three models you've constructed predict future covid cases on a global level. The prediction models include

- Support Vector Machine
- Polynomial Regression
- Bayesian Ridge Regression

[]:

```
# Future predictions using SVM
```

[]:

```
# Future predictions using polynomial regression
```

[]:

```
# Future predictions using Bayesian Ridge
```

[]:

11) Mortality Rate (worldwide) **susceptible to change* *

[]:

12) Recovery Rate (worldwide) **suceptible to change* *

[]:

13)Graphing deaths against recoveries

[]:

14) Plotting the number of deaths against the number of recoveries.

[]:

15) Getting information about countries/regions that have confirmed coronavirus cases

16) Data table;

Show covid data for several countries. The table will include the number of confirmed cases, deaths, recoveries, active cases, incidence rate, and mortality rate.

17) Getting the latest information about provinces/states that have confirmed coronavirus cases

[]:

```
# remove areas with no confirmed cases
```

```
# number of cases per province/state/city top 100
```

```
# number of cases per country/region
```

```
# return the data table with province/state info for a given
country
```

```
# remove areas with no confirmed cases
```

```
# number of cases per country/region
```

18) Data table for the United States

19) Data table for Brazil

20) Data table for India

21) Data table for Russia

22) Data table for China

23) Data table for United Kingdom

24) Data table for France

25) Data table for Italy

26) Data table for Spain

27) Data table for Germany

28) Data table for the Netherlands

29) Data table for Colombia

30) Data table for Colombia

31) Data table for Mexico

32) Data table for Greece

33) Bar Chart Visualizations for COVID-19.

This offers us some insights for how different countries/regions compare in terms of covid cases.

34) Pie Chart Visualizations for COVID-19.

Although pie charts are not necessarily the most informative visualizations. Add them to increase some variety for the visualizations.

35) US Medical Data on Testing.

This gives us some information on US coronavirus testing, which is important for getting a clear picture of the pandemic.

Part 2; Following Steps:

1) Taking a look at Apple's mobility data.

It can help you understand the relationship between mobility and daily increases in coronavirus cases. Use graphs, subgraph or anything related you wish to construct a relation between the clinical and mobility data.