MERCURY GROUP, PROJECT 4

CRISTIAN SALGUEIRO, HARSHA MAHAJAN, SAHIL KUMAR

DECEMBER 14, 2020

ECE 59500 SOCIAL NETWORKS W/ MACHINE LEARNING

INDIANA UNIVERSITY PURDUE UNIVERSITY INDIANAPOLIS

The objective of this project is to analyze COVID-19 data and identify trends. COVID-19 case growth rate was determined for several countries. The growth rate was used to predict future cases, and the prediction model was validated against data. A correlation coefficient was then calculated for growth rates of different countries and the relationship between this coefficient and a friendliness factor calculated using international trade data.

Figure 1 and Figure 2 shows the error between predicted cases and actual cases for each m value. In order to get these values, we considered the last m values (in our case 1 to 14) of x_i series and the moving average growth rate is calculated according to

$$G_n = \frac{1}{m}\sum_{i=n-m}^{n-1}\left(\frac{x_i - x_{i-1}}{x_{i-1}}\right) \tag{1}$$

Using the last m days, the simple iteration provided predicted confirmed cases beyond i=n-1.

$$\hat{x}_{n+1} = x_n(1 + G_n). \tag{2}$$

From equation 3 the predicted cases and actual cases were compared on the basis of calculating the error between two values. We have calculated the overall error of each m value for all the predicted i=n-1 days, in our case we predicted 10 values. For USA and Brazil data, we found that m=11 yields less error between predicted values and actual values with respect to other values of m. If we consider values beyond m=14 the error increases drastically.

$$Error = \sum \hat{x}_{n+1} - x_{n+1}. \tag{3}$$

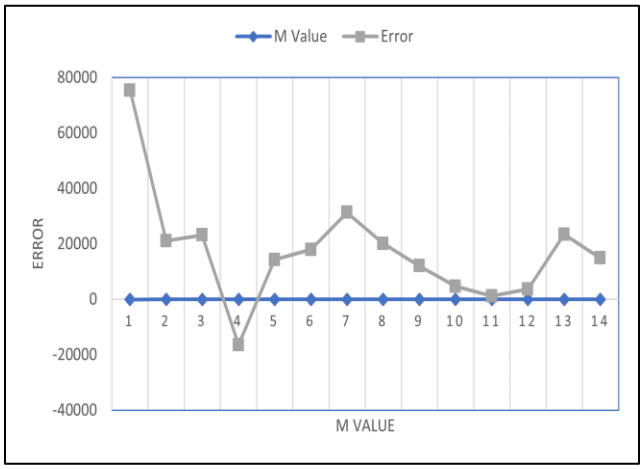

Figure 1USA Overall Error in predictions for each m value

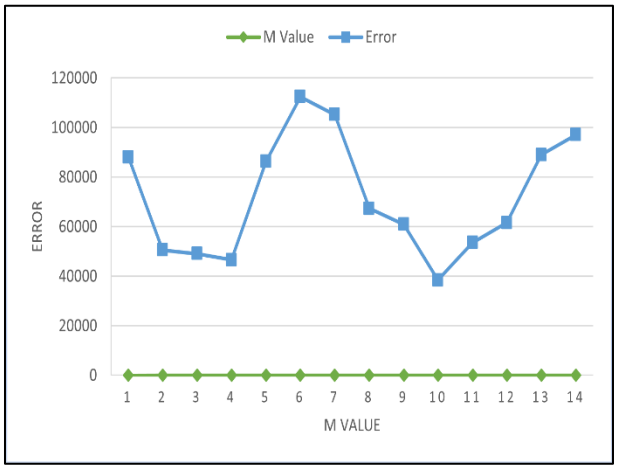

Figure 2Brazil Overall Error in predictions for each m value

Figure 3 below shows the moving average of the growth rate for 6 countries. Moving average of growth rate is the preferred value to predict the value of the COVID-19 cases over a period of time as it is more

efficient in predicting the values. Moving Average is defined as Sum of Growth rate of last 'N' days divide by 'N'.

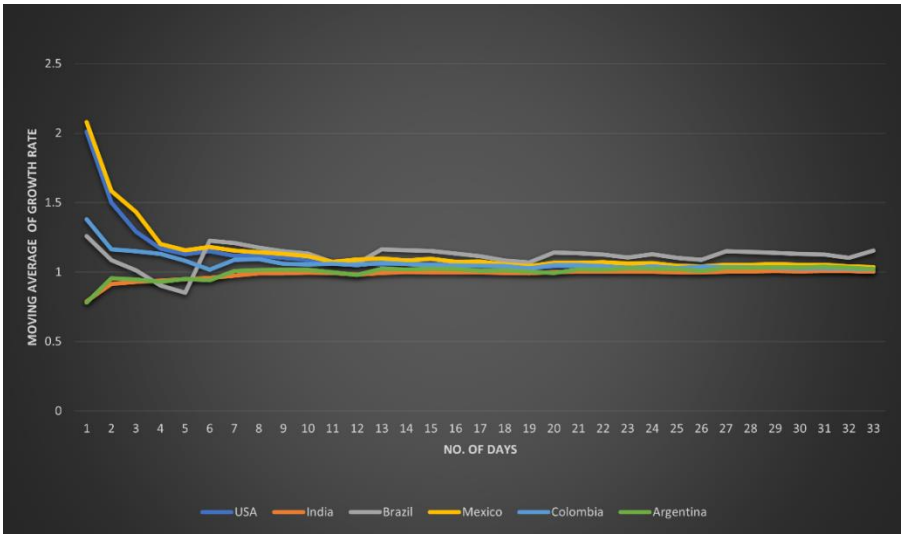

Figure 3. Moving Average of Growth Rate for 6 Countries

Figure 4 below shows a trade network of several of the top countries for total COVID-19 cases. The friendliness factor used to determine the coloring of the network edges is based on 2018 import data for the countries shown. For each country in the network, the export value of each outgoing edge was normalized by the total country export value to the other countries in the network. Even normalized, the graph demonstrates the significant economic relationship the United States, which has the most COVID-19 cases of any country, has with the other countries in the network. The United States has numerous high friendliness edges with other countries. It can also be seen that Brazil's economic presence may be a factor in driving COVID-19 spread in South America. Despite the comparatively low value trade interactions Argentina and Colombia have with the other top countries, they both share a significant economic relationship with Brazil.

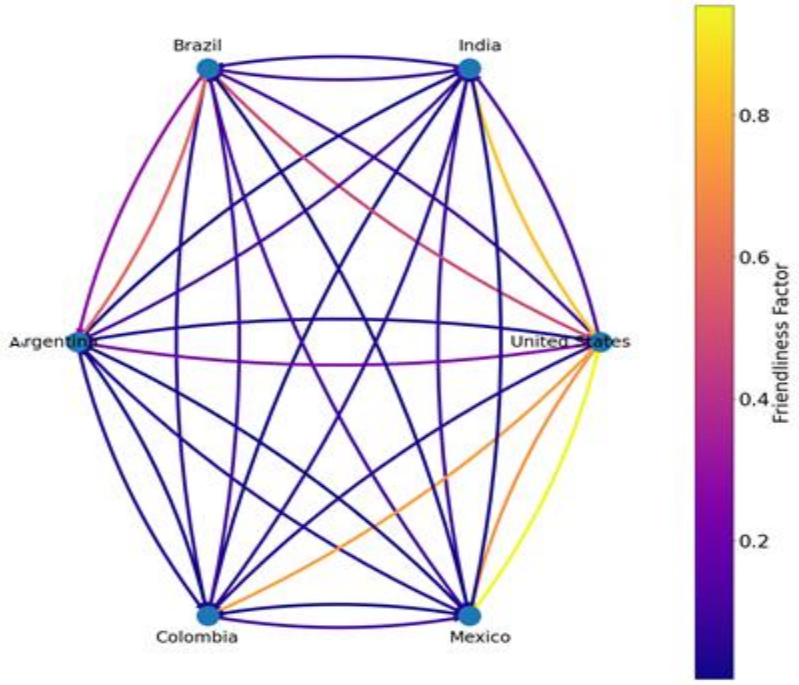

Figure 4. International Trade Network for Top COVID-19 Countries

Using the growth rates over time calculated for each country in the trade network, the correlation coefficient was obtained for each relationship. For any two countries in the network, a vector of each country's growth rate date was input into the NumPy corrcoef function to calculate the correlation. These values were then plotted against the friendliness factors for each relationship. Because each relationship contains two unique friendliness values and one correlation coefficient, the correlation values are plotted against the maximum friendliness value in the relationship.

The resulting plot in Figure 5 below does not appear to show any relationship between the maximum friendliness between two countries and the correlation coefficient of their growth rate data over time. The plot also shows there is no negative growth rate correlation between countries. These countries are all in the top 10 for active cases and their growth rates are positive and generally stable.

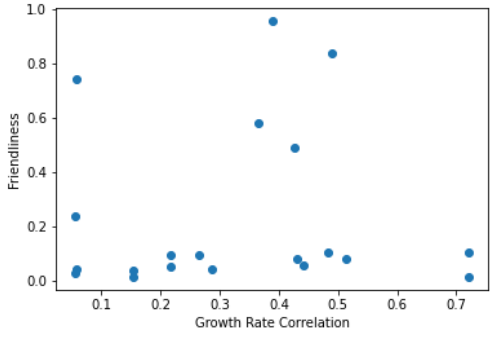

Figure 5. Friendliness vs. Growth Rate Correlation

Because the United States has such a strong economic relationship with all other countries in the network, its friendliness values appear to be outliers. A second network excluding the United States was created to assess if outlier data was affecting the relationship between growth correlation and friendliness. Figure 6 below shows the updated network. Despite the absence of the high centrality U.S node, there is still no clear relationship between friendliness and growth rate correlation.

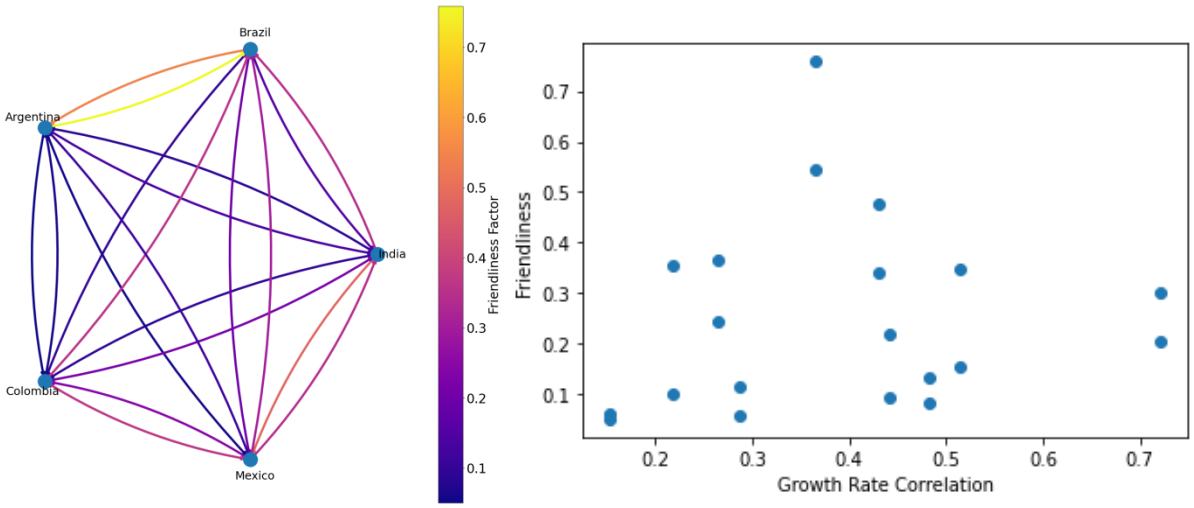

Figure 3. International Friendliness Network Without U.S

The lack of a trend between friendliness and growth rate correlation suggests the COVID-19 case count growth is not due to international relationships. Therefore, once the virus reaches a country via international travel it typically spreads through community. As demonstrated below, higher population density does not correspond to high case count. Instead, growth rate of each country is more likely driven by lack of mitigation efforts in local policy.

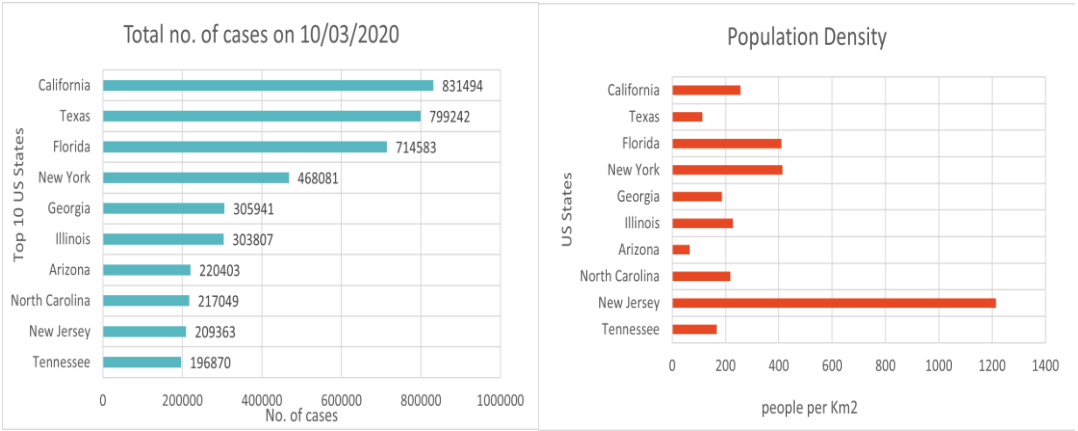

Figure 4. Case Count and Population Density Comparison

The presented analysis on m values and growth rate shows that taking last 10-12 days gives the most accurate prediction of number of cases. USA has the highest number of covid-19 cases, still has a significant economic relation with another countries. Also, Brazil's economic presence can be the reason for spread of virus in the South America. By plotting the friendliness with growth rate correlation, the graph shows there is no negative growth rate correlation.

References:

1. https://www.wto.org/english/res_e/statis_e/trade_datasets_e.htm
2. https://ourworldindata.org/coronavirus-source-data
3. www.Kaggle.com

Code:

Code for creating network graph and correlation plots:

```
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np
data = pd.read_csv("WTOdata.csv",encoding="latin-1")
top = ["United States of America","India","Brazil","Argentina","Colombia","Mexico"]#,"Peru","South
Africa","Italy","Iran","Chile","Germany","Iraq","Bangladesh","Indonesia","Phillippines"]
data = data[(data['Reporting Economy'].isin(top))&data(['Partner Economy'].isin(top))]
data = data.loc[(data['Year'] == 2018) & (data['Product/Sector Classification'] == 'Harmonized
System')]
data.to_csv("topTrade.csv")
colname = ['Reporting Economy','Partner Economy', 'Import Value','Friendliness']
out = pd.DataFrame(columns = colname)
for i in top:
   total = 0
   for j in top:
      trade = sum(data[(data['Partner Economy']==i) & (data['Reporting Economy']==j)]['Value'])
      trade = int(round(trade/1000000)) #final value in millions USD
      total = total + trade
      out = out.append({'Partner Economy':i,'Reporting Economy':j,'Import
Value':trade,'Friendliness':trade}, ignore_index=True)
   if total == 0:
      out['Friendliness'].loc[out['Partner Economy']==i] = 0
   else:
      out['Friendliness'].loc[out['Partner Economy']==i] = out['Friendliness'].loc[out['Partner
Economy']==i]/total
out = out.loc[out['Import Value'] !=0]
out.to_csv("Network_Data.csv")
#out = pd.read_csv('Network_Data.csv')
out['Correlation'] = np.nan
growth = pd.read_csv('growth_rate_data.csv')
index = list(growth.drop_duplicates('location')['location'])
for i in index:
   for j in index:
      correlation = np.corrcoef(growth['Growth Rate'].loc[growth['location']==i],growth['Growth
Rate'].loc[growth['location']==j])
      out['Correlation'].loc[(out['Reporting Economy']==i) & (out['Partner Economy']==j)] =
correlation[0,1]
plt.figure(figsize = (20,20))
G = nx.from_pandas_edgelist(out,source = 'Partner Economy',target = 'Reporting
Economy',edge_attr='Friendliness',create_using=nx.DiGraph())
pos = nx.circular_layout(G)
label_pos = {}
countries = list(pos.keys()) #create list of countries to offset label position
for k in countries:
   label_pos[k] = pos[k]*1.08 #node label offset
```

```
nx.draw_networkx_labels(G,label_pos,font_size=25)
imports = nx.get_edge_attributes(G,'Friendliness').values()
cval = np.array(list(imports))
vmin = min(cval)
vmax = max(cval)
cmap = plt.cm.plasma
nx.draw_networkx_nodes(G,pos,node_size = 1000)
edges = nx.draw_networkx_edges(G,pos,width = 5,arrowsize = 15, connectionstyle = 'arc3, rad
=0.1',alpha = 1,edge_color = cval,edge_cmap = cmap,vmin=vmin,vmax=vmax)
plt.axis('off')
sm = plt.cm.ScalarMappable(cmap = cmap,norm = plt.Normalize(vmin = vmin,vmax = vmax)) #take
exponential to map colorbar to real values
clb = plt.colorbar(sm)
clb.ax.tick_params(labelsize = 28)
clb.set_label('Friendliness Factor',size = 28)
out = out[(out['Friendliness']!=0) & (out['Partner Economy']!='Italy')]
out = out.sort_values('Friendliness',axis = 0,ascending = False,inplace = False)
out = out.drop_duplicates('Correlation')
out2 = out[(out['Reporting Economy']!='United States of America')& (out['Partner
Economy']!='United States of America')]
plt.show()
plt.figure()
x = out['Correlation']
y = out['Friendliness']
plt.ylabel('Friendliness')
plt.xlabel('Growth Rate Correlation')
plt.scatter(x,y)
plt.show()
plt.figure()
x2 = out2['Correlation']
y2 = out2['Friendliness']
m2,b2 = np.polyfit(x2,y2,1)
plt.scatter(x2,y2)
plt.ylabel('Friendliness')
plt.xlabel('Growth Rate Correlation')
```

Code for calculating the moving average of Growth Rate:-

```
def growt_rate() :
        global y

y_df=df.drop(['new_deaths','total_cases','total_deaths','weekly_cases','weekly_deaths','biweekly_c
ases','biweekly_deaths'],axis =1 )
        #for i in date_array :
        for j in range(0,len(country_array)) :

            y=y_df.loc[df['date'].isin(date_array) & (df['location']==country_array[j])  ]
            new_cases= y['new_cases']
            new_case=new_cases.tolist()
```

```
a= len(new_case)
print(type(new_case))
growth=[]
moving_average=[]
for x in range(0,a) :

   if (new_case[x]!=0) and (new_case[x-1]!=0) :
      gnumbers=round(new_case[x]/new_case[x-1],2)
      growth.append(gnumbers)
   else:
      gnumbers=0
      growth.append(gnumbers)
#for i in range(0,len(growth)):

y['Growth']=growth
sum=0
for i in range(0,a):
   sum=sum+growth[i]
   moving_average.append(sum/(i+1))
y['Moving_Average']=moving_average

print(y)
print(f'Growth List For Country {country_array[j]} ..')
print(list(growth))
print('\n')
print(len(growth))
print('\n')
```

Code for calculating Error for each m value:-

```
new_case=df['new_cases_x'].tolist()
def calculate_m(m,n):
   sum=0
   for i in range(n-m,n):
      sum=sum+((new_case[i]-new_case[i-1])/new_case[i-1])
   return sum/m
m=int(input())
actual_cases=[]
error=0
growth=[]
for n in range(m,m+11):
   growth_average=calculate_m(m,n)
   predict_cases=new_case[n]*(1 + growth_average)
   error=error+(new_case[n]*(1 + growth_average)-new_case[n+1])
print(m)
print(error)
```