UNIVERSITI MALAYA

Faculty of Computer Science and Information Technology

WOA7001
ADVANCED Algorithms

GROUP ASSIGNMENT

Group 4

Xu Wang S2105012

Zixuan Wang S2122769

Zhang Yinrui S2033281

Table of Contents

1. Assignment of responsibilities 1

2.Project analysis 4

3.Planning and execution based on FILA form 20

4.Overall reflection of project experience 23

5. Conclusion 24

6、Reference 24

1.Roles

No	Matric No	Name	Team Role
1	S2105012	Xu Wang	Data Analyzer
2	S2122769	Zixuan Wang	Algorithm Developer
3	S2033281	Zhang Yinrui	Web Developer

2.Project analysis

Function: the function of this project is to realize the movement of visualization personnel and its relationship with covid-19 communication.

Objective: the goal of the project is to implement algorithms and data structures to solve real-world problems.

It is divided into three modules: data visualization module, web design module and algorithm design module.

1. Data visualization module:

Data preprocessing

This module uses Python language, pandas to process data sets, and MATLAB lib to draw graphics as a tool library.

First, import the data and use the following four data sets: clusters and cases_ state, checkin_ malaysia_ time, trace_ malaysia

```
# import dataset
clusters = pd.read_csv('/content/drive/MyDrive/WOA7001/Project/DataSource/clusters.csv'
cases_state = pd.read_csv('/content/drive/MyDrive/WOA7001/Project/DataSource/cases_stat
checkin_malaysia_time = pd.read_csv('/content/drive/MyDrive/WOA7001/Project/DataSource/
trace_malaysia = pd.read_csv('/content/drive/MyDrive/WOA7001/Project/DataSource/trace_r
```

Then there is data preprocessing, which aims to clean up null values and duplicate values in the data set,

```
clusters = clusters.dropna(subset=['summary_en'])
```

Then use chi square test to test the fields with high correlation with the project. The fields with low correlation will affect the accuracy of the project, so they will not be used.

Define a function to simplify the code of chi square test and avoid code redundancy.

```
from scipy.stats import chi2_contingency
```

```
def B():
    # create contigency table for X vs Y
    info = pd.crosstab(index=clusters[X], columns=clusters[Y])
    # perform chi square
    print(info.values.tolist())
    chi2, p, dof, expctd = chi2_contingency(info.values.tolist())
    print('chi2 value: ', chi2)
    print('p value: ', p)
    print('dof value: ', dof)

    if p < 0.05:
        print(X + ' and ' + Y +' are independent of each other.')
    else :
        print(X + ' and ' + Y +' are NOT independent of each other')
```

After testing, the correlation p value is as follows:

```
# cases_total and recovered
X = 'cases_total'
Y = 'category'
B()
```

```
[[0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 1, 0, 1], [1, 0, 0, 0, 1, 0, 6], [0, 0, 6, 0, 1, 0, 3
chi2 value: 777.4303589876318
p value: 0.23670046006146517
dof value: 750
cases_total and category are NOT independent of each other
```

a) The first figure is line chart to show the change of registration number every 1 / 2 hour in whole Malaysia,

The first step is to set up the horizontal and vertical coordinates

```
#Read row header (hours), set to X-axis
labels = list(checkin_malaysia_time.columns.values)
```

```
#Read the first row of data, not the row header
y = checkin_malaysia_time.iloc[0]
```

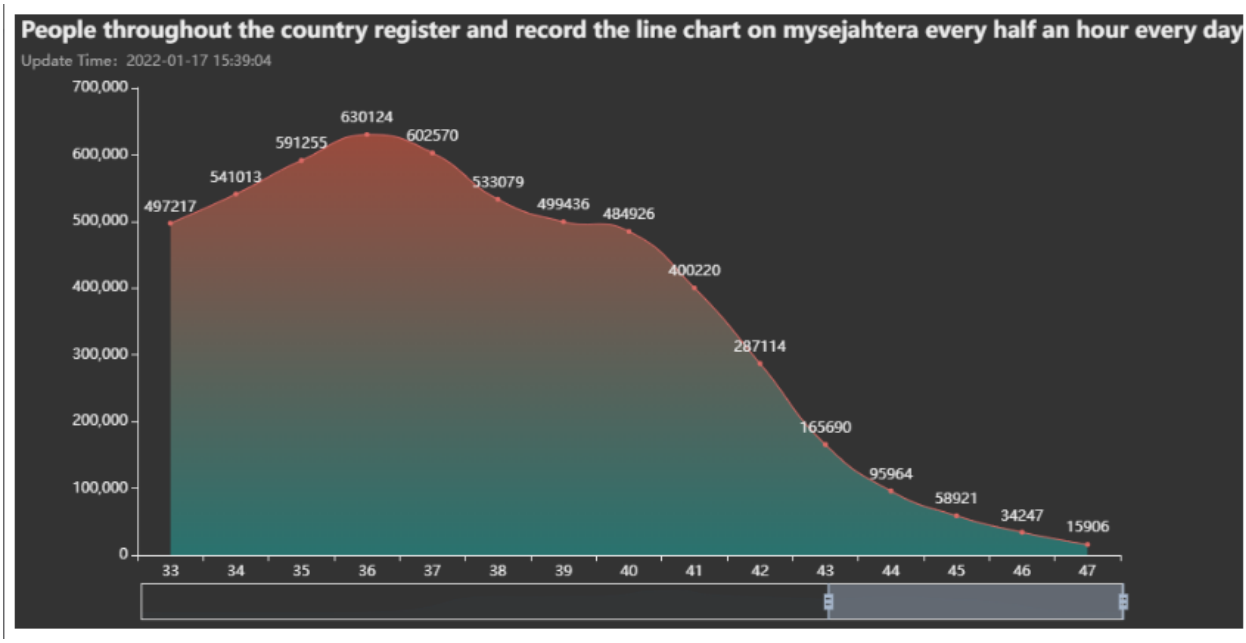
Then use the drawing function in the matlablib library to draw graphics, and input the extracted data,

```
#  checkin_malaysia_time.csv
import  requests
import  pyecharts
from  pyecharts.charts  import  *
from  pyecharts  import  options  as  opts
from  pyecharts.commons.utils  import  JsCode
from  pyecharts.datasets  import  register_url
import  pandas  as  pd
from  datetime  import  date
import  datetime
from  dateutil  import  tz
import  json
update_date  =  datetime.datetime.now(tz=tz.gettz(
        'Asia/Shanghai')).strftime('%Y-%m-%d  %H:%M:%S')
line  =  (Line(init_opts=opts.InitOpts(theme='dark',  width='1000px'))
                .add_xaxis(labels)
                .add_yaxis("check  in",  y,  is_smooth=True,
                areastyle_opts=opts.AreaStyleOpts(opacity=0.5,color=JsCode("""new  echart

                .set_series_opts(label_opts=opts.LabelOpts(is_show=True))
                .set_global_opts(
                        datazoom_opts=opts.DataZoomOpts(range_start=70,  range_end=100),
                        title_opts=opts.TitleOpts(title="People  throughout  the  country
                        xaxis_opts=opts.AxisOpts(
                                splitline_opts=opts.SplitLineOpts(is_show=False)),
                        legend_opts=opts.LegendOpts(is_show=False),
```

After drawing, the figure is as follows:

The abscissa is 0-48, which means every half hour of the day. The ordinate is the number of people checking in in mysejahtera. The time of the data can be seen in the upper left corner



Because there are 48 and a half hours in total, the display will be very bulky, so design a scrollbar that can be dragged and zoomed to observe all half-hour data in a fixed page.

b)Map the total number of confirmed cases in each state (map of Malaysia)
This drawing is based on case_ State dataset,
Install the pyecarts library,

```
pip install pyecharts
```

The extracted dataframe is as follows:

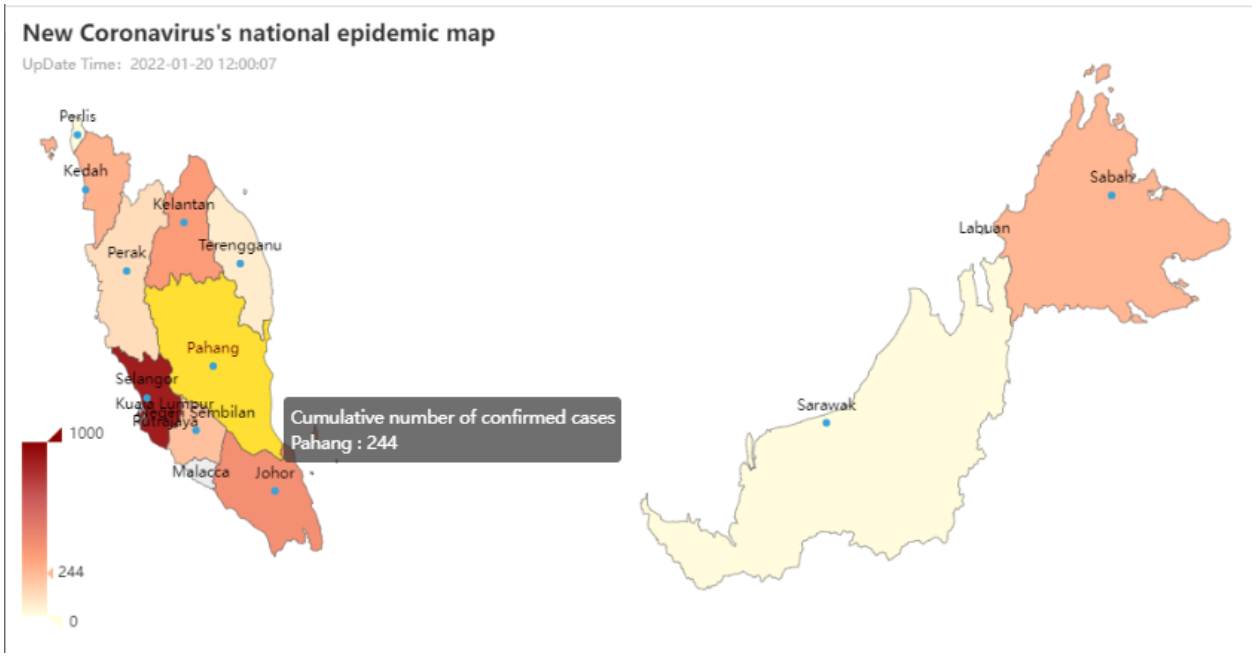
```
cases_state_drop_duplicated = cases_state.drop_duplicates(subset=['state'],keep='first'
# cases_state_drop_duplicated['state']
```

Next, draw graphics. The design idea of this module is: first, since interactive visualization is involved, I think of using the best pyecharts Library in Python, and then based on the requirements, through the data source clusters The field in CSV envisages a national epidemic map of Malaysia, which visually shows the provinces with a large number of infected people, and the color will deepen. Users can move the mouse to preview the total number of infected people in the current state. They can also drag the left adjustment bar to show the distribution of infected people in different sections of the map, and design the clickable function of the map, You can click to enter the designated state and display all the epidemic information of the

current state. The information has been divided into groups through the cluster, so you can more intuitively see the epidemic information under each cluster.

```
# b)Map the total number of confirmed cases in each state
# cases_state.csv
from pyecharts.charts import Map
from pyecharts import options as opt
from datetime import date
import datetime
from dateutil import tz
update_date = datetime.datetime.now(tz=tz.gettz('Asia/Shanghai')).strftime('%Y-%m-%d %
_map = (
        Map(init_opts=opts.InitOpts(theme='light', width='1000px'))
        # is_ Roam = true set mouse zoom and pan roaming is_ map_ symbol_ S
        .add("Cumulative number of confirmed cases", cases_state_drop_duplicated[['s
        .set_series_opts(label_opts=opts.LabelOpts(is_show=True))
        .set_global_opts(
                title_opts=opts.TitleOpts(title="New Coronavirus's national epidemic
                subtitle="UpDate Time: {}".format(update_date)),
                legend_opts=opts.LegendOpts(is_show=False),
                visualmap_opts=opts.VisualMapOpts(is_show=True, max_=1000,is_piecewise=F
        )
)
# _map.render(path='/content/drive/MyDrive/WOA7001/Project/map of Malaysia.html.html')
_map.render(path=r'/content/drive/MyDrive/WOA7001/Project/VisualizationFile/Map of Mala
```

The drawn figure is as follows:



c)The overlapping histogram shows the comparison of new and cured people

The purpose of this figure is to jump to the personnel flow track information page of the corresponding state after the user clicks a state in B).
First, according to the state field in the clusters file, 1-16 in represent different states in Malaysia. I dream to extract the data of the corresponding states according to this code.

```
# Extract the data of each state according to the state number

# 1 Johor
Johor_data = clusters[clusters['state'].isin(['1'])]
# 2 kedah
Kedah_data = clusters[clusters['state'].isin(['2'])]
# 3 Kelantan
Kelantan_data = clusters[clusters['state'].isin(['3'])]
# 4 malacca
Malacca_data = clusters[clusters['state'].isin(['4'])]
# 5 SembilanM
Sembilan_data = clusters[clusters['state'].isin(['5'])]
# 6 Pahang
Pahang_data = clusters[clusters['state'].isin(['6'])]
# 7 Penang
Penang_data = clusters[clusters['state'].isin(['7'])]
# 8 Perak
Perak_data = clusters[clusters['state'].isin(['8'])]
# 9 Perlis
Perlis_data = clusters[clusters['state'].isin(['9'])]
# 10 Selangor
Selangor_data = clusters[clusters['state'].isin(['10'])]
# 11 Terengganu
Terengganu_data = clusters[clusters['state'].isin(['11'])]
# 12 Sabah
Sabah_data = clusters[clusters['state'].isin(['12'])]
```

As flows,Define the function, extract the fields to be displayed and merge them into a new dataframe

```
# Define the function, extract the fields to be displayed and merge them int
def combineData(data):
    def get_df_name(df):
        name =[x for x in globals() if globals()[x] is df][0]
        return name
    dataname = get_df_name(data)

    newdata = zip(data['date_announced'], data['cluster'], data['summary_en'])
    data = pd.DataFrame(data = newdata, columns=['date_announced', 'cluster', 'sur
    return [data, dataname]
```

Define a function to generate the moving track information table of infected persons in each state according to the state data.

```
# Export data to HTML table
def createInfoHTML(data, dataname):
    pd.set_option('display.width', 1000)
    pd.set_option('colheader_justify', 'center')
    np.random.seed(6182018)
    pd.set_option('colheader_justify', 'center')      # FOR TABLE <th>
    html_string = '''
    <html>
        <head><title>HTML Pandas Dataframe with CSS</title></head>
        <link rel="stylesheet" type="text/css" href="df_style.css"/>
        <body>
            {table}
        </body>
    </html>.
    '''
    # OUTPUT AN HTML FILE
    # with open(dataname+'.html', 'w') as f:
    #         f.write(html_string.format(table=data.to_html(classes='mystyle')))

    with open('/content/drive/MyDrive/WOA7001/Project/VisualizationFile/'+dataname+'.htr
            f.write(html_string.format(table=data.to_html(classes='mystyle')))
```

Call the function below and pass in parameters. Because there are many states, only two states are given as examples,

```
# Generate the flow information table of infected persons in Johor
dataTEMP = combineData(Johor_data)
data = dataTEMP[0]
dataname = dataTEMP[1]
createInfoHTML(data, dataname)
```

```
# Generate the flow information table of infected persons in Kedah
dataTEMP = combineData(Kedah_data)
data = dataTEMP[0]
dataname = dataTEMP[1]
createInfoHTML(data, dataname)
```

The generated moving track information table of infected persons in each state is as follows:

	date_announced	cluster	summary_en
0	2021-09-14	KLUSTER SAWIT LEPAR HILIR	This cluster involves workers in an oil palm plantation located at Ladang Lepar Hilir 5, Kuantan, Pahang. The index case for this cluster was a 21 years old male Bangladesh citizen who was reported positive for COVID-19 on 4th September 2021 due to symptomatic individual screening.
1	2021-09-19	KLUSTER INDUSTRI BUKIT KOMAN	The cluster involves workers of a factory located in Bukit Koman, Raub. The index case for this cluster was a 27-year-old male Malaysian citizen who was reported positive for COVID-19 on 16th September 2021 due to symptomatic individual screening. The case started began to develop symptoms on 12th September 2021.
2	2021-10-07	KLUSTER PAYA SENDAYAN	The index case for this cluster is Case No. 2267315, a female Malaysian citizen aged 32 years who reported positive COVID-19 on 2nd October 2021 due to symptomatic individual screening. Cases began to appear symptomatic on 28th September 2021. The name of this cluster refers to the locality where the spread of cases was reported to occur, namely in Kampung Paya Sendayan, Lipat Kajang, Temerloh.
3	2021-10-07	KLUSTER KAMPUNG PANTOS	The index case for this cluster is Case No. 2284338, a male Malaysian citizen aged 18 years who were reported positive for COVID-19 on 4th October 2021 due to close contact screening. The name of this cluster refers to the locality where the spread of cases is reported to occur, namely in Kampung Pantos, Kuala Lipis.
4	2021-10-08	KLUSTER KAMPUNG SUNGAI UBI	The index case for this cluster is Case No.2293611, a male Malaysian citizen aged 34 years who reported positive for COVID-19 on 5th October 2021 due to screening of symptomatic individuals. The case began to develop symptoms on 4th October 2021. The name of this cluster refers to the locality where the spread of cases was reported to occur, namely in Kampung Sungai Ubi, Tanah Rata, Cameron Highlands.
5	2021-10-08	KLUSTER RUMPUN MAKMUR	The index case for this cluster is Case No. 2276059, a male Malaysian citizen aged 42 years who reported positive COVID-19 on 3rd October 2021 due to symptomatic individual screening. The case began to develop symptoms on 26th September 2021. The name of this cluster refers to the locality where the spread of cases was reported to occur, namely in Kampung Rumpun Makmur, Mukim Jenderak, Temerloh.

Of course, this is only a small part of the massive data, which can be shown as a sample.

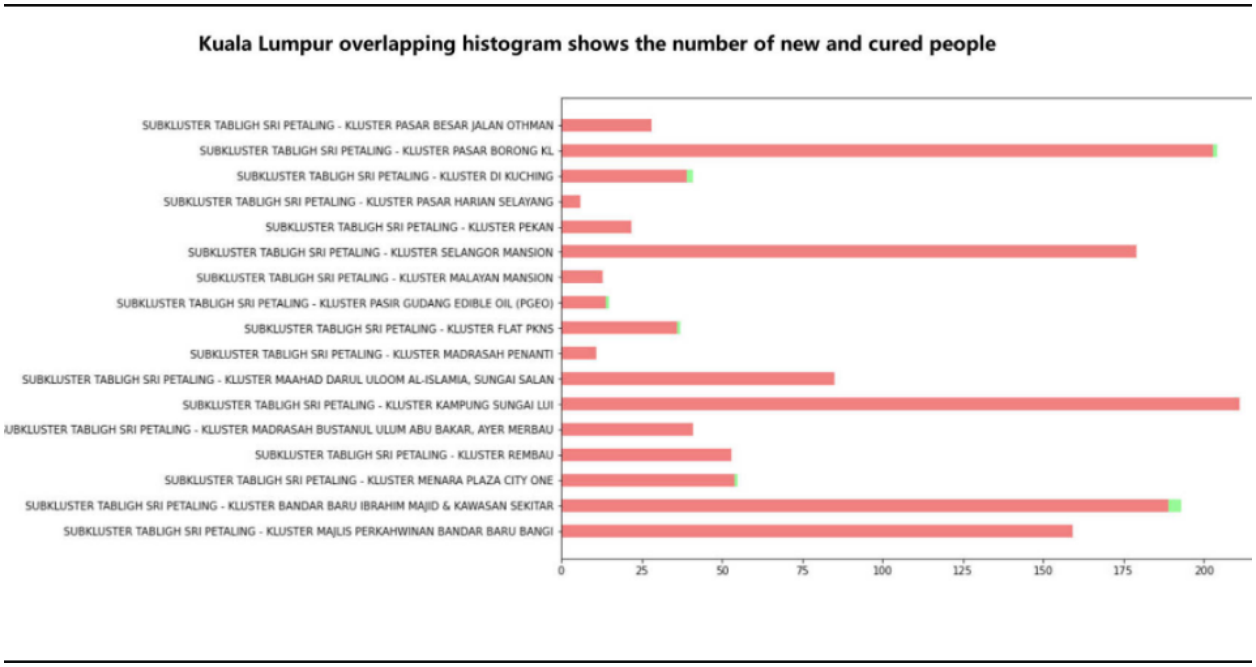
d)Indicates the flow of confirmed cases - generate the flow information table of infected persons in each state (with map)

This graph is to show the comparison of new and cured people in Kuala Lumpur. The horizontal overlapping histogram is used to make the data comparison more intuitive

```
# Make horizontal bar chart
cases_total = KualaLumpur_data['cases_total']
recovered = KualaLumpur_data['recovered']
cluster = KualaLumpur_data['cluster']

plt.figure(figsize=(12, 8))
plt.barh(width=cases_total, y=cluster, height= 0.5, align='center', color="palegreer
plt.barh(width=recovered, y=cluster, height= 0.5, color="lightcoral")
plt.savefig("/content/drive/MyDrive/WOA7001/Project/VisualizationFile/Kuala Lumpur ove
```

The generated figure is as follows: the abscissa is the number of people, and the ordinate is the names of various regions in Kuala Lumpur. From the figure, you can see red and green, representing the number of new people and the number of cured people respectively.



e)Flow diagram

This figure is a flow diagram based on dijestra algorithm. The idea is as follows. From this figure, we can see the path of each state through which the infection case passes. Among them, this line is the shortest path calculated by dijestra algorithm, and then we draw it. It could have two choices. One is to pass through Pahang and then Sembilan. The other is the one drawn on

the way. It passes through Sembilan and then to Pahang. According to the distance between various states provided by us, the algorithm calculates the shortest path.

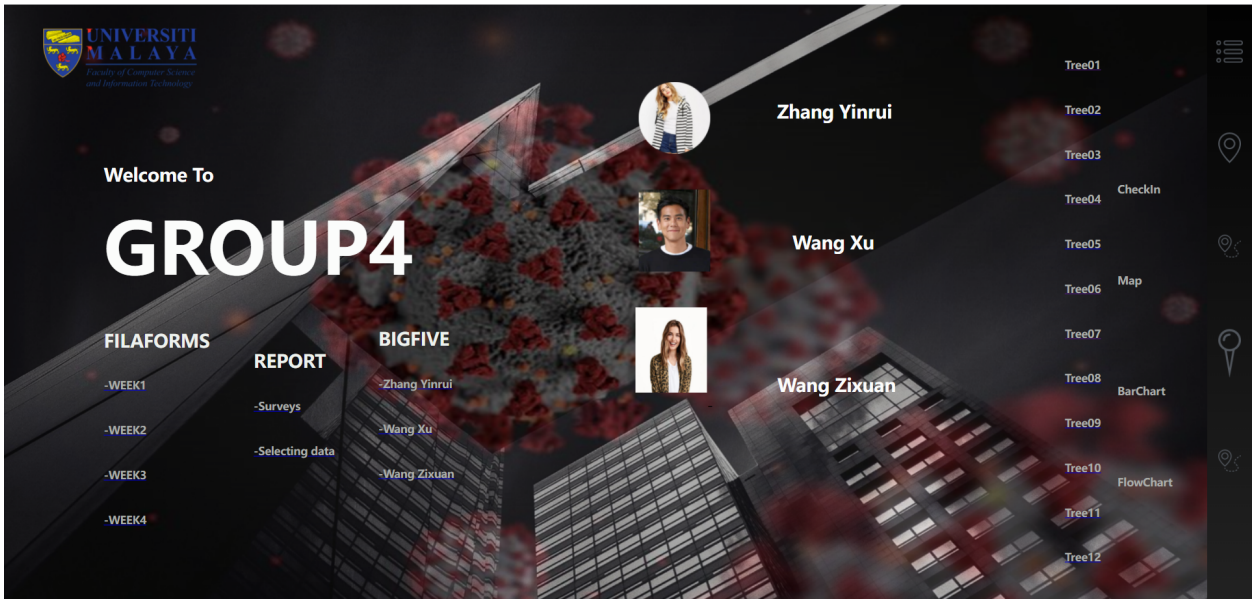
First, use Google map to find out the GPS position of each state, and then find out the distance (km) between each state according to the data on Google. According to the states through which the cases pass in the clusters file, call the dijestra algorithm to calculate the shortest path

```
from pyecharts.charts import Geo
from pyecharts import options as opts
from pyecharts.globals import ChartType, SymbolType
update_date = datetime.datetime.now(tz=tz.gettz('Asia/Shanghai')).strftime('%Y-%m-%d %
geo = Geo()
geo.add_coordinate(name="Johor", longitude=103.7618, latitude=1.4854)
geo.add_coordinate(name="kedah", longitude=100.3685, latitude=6.1184)
geo.add_coordinate(name="Kelantan", longitude=102.2381, latitude=6.1254)
geo.add_coordinate(name="malacca", longitude=102.2501, latitude=2.1896)
geo.add_coordinate(name="Sembilan", longitude=101.9424, latitude=2.7258)
geo.add_coordinate(name="Pahang", longitude=103.2202, latitude=3.7634)
geo.add_coordinate(name="Perak", longitude=101.0901, latitude=4.5921)
geo.add_coordinate(name="Perlis", longitude=100.2048, latitude=6.4449)
geo.add_coordinate(name="Selangor", longitude=101.5183, latitude=3.0738)
geo.add_coordinate(name="Sabah", longitude=116.0753, latitude=5.9788)
geo.add_coordinate(name="Sarawak", longitude=110.3593, latitude=1.5533)
```

Then draw it through the directed line segment, and the effect is as follows:

2. Web design module

The web design results are as follows



On the main interface, we designed the group number, the group members, the background is New Coronavirus pictures, the upper left corner is logo of University of Malaya Computer Institute, set up the Fila form button, users can directly download FILA form by clicking the corresponding week, the middle is the report, the surveys collation after interview, the following is the introduction of the dataset we selected. On the right is the biglive file. Tree01-tree12 on the right side of the page is the tree generated according to Huffman code, corresponding to the branch diagram of infected personnel flow in Pahang state, Malaysia every month in 2021. On the far right are the four buttons for data visualization, Respectively corresponding,a)Line chart to show the change of registration number every 1 / 2 hour in whole Malaysia, b)Map the total number of confirmed cases in each state (map of Malaysia),d)Indicates the flow of confirmed cases - generate the flow information table of infected persons in each state (with map),e)Flow diagram.

3. Algorithm design module

Dijestra algorithm:

```
def Dijkstra(network,s,d):
        print("Start Dijstra Path:")
        path=[]
        n=len(network)
        #print(n)
        #print(network)
        fmax=999
        w=[[0 for i in range(n)]for j in range(n)]
        book=[0 for i in range(n)]
        dis=[fmax for i in range(n)]
        book[s-1]=1
        midpath=[-1 for i in range(n)]
        for i in range(n):
                for j in range(n):
                        if network[i][j]!=0:
                                w[i][j]=network[i][j]
                        else:
                                w[i][j]=fmax
                        if i==s-1 and network[i][j]!=0:
                                dis[j]=network[i][j]
        for i in range(n-1):
                min=fmax
                for j in range(n):
                        if book[j]==0 and dis[j]<min:
                                min=dis[j]
                                u=j
                book[u]=1
                for v in range(n):
                        if dis[v]>dis[u]+w[u][v]:
```

```
                                dis[v]=dis[u]+w[u][v]
                                midpath[v]=u+1
        j=d-1
        path.append(d)
        while(midpath[j]!=-1):
                path.append(midpath[j])
                j=midpath[j]-1
        path.append(s)
        path.reverse()
        print(path)
        #print(midpath)
        print(dis)
        #return  path

network=[[0, 188, 231, 241, 0, 0, 0],
                  [188, 0, 71, 197, 0, 0, 0],
                  [231, 71, 0, 152, 120, 65, 55],
                  [241, 197, 152, 0, 0, 0, 0],
                  [0, 0, 120, 0, 0, 56, 69],
                  [14, 0, 0, 65, 0, 56, 0, 23],
                  [16, 0, 0, 55, 0, 69, 23, 0]]
Dijkstra(network, 3, 7)
```

Code interpretation:

N represents the number of nodes, that is, the dimension of the adjacency matrix, which is used for circular traversal.

Convert adjacency matrix into dimension matrix with w = [[0 for I in range (n)] for J in range (n)].

Book is used to determine whether it is already the smallest tag.

DIS is used to store the minimum distance to other nodes.

Midpath means to jump up the search list.

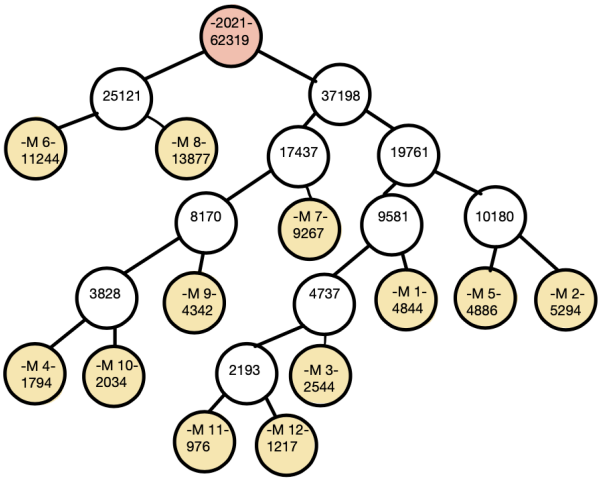
Then enter the loop traversal to store and update the shortest distance of the node.

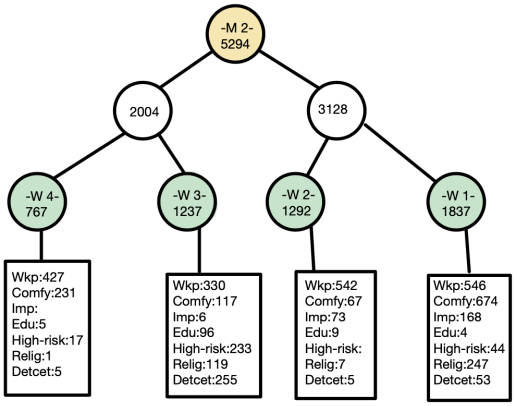
Finally, make a network table to store the weight of each node. In this project, weight is the distance between states.

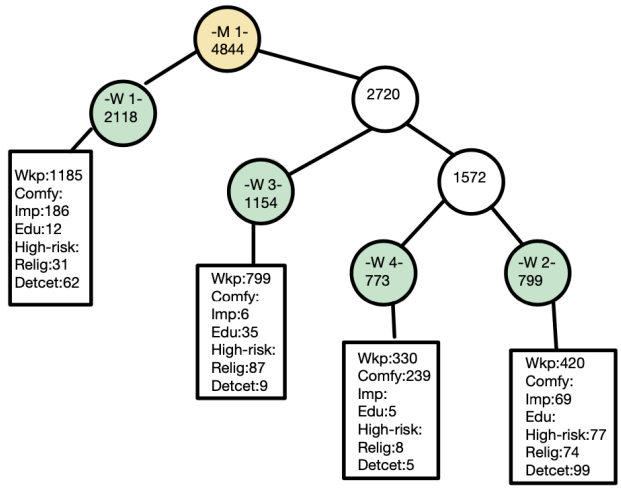
A person wants to get from one place to another. Theoretically, there are many routes to choose, but in fact, most people tend to choose the nearest route.

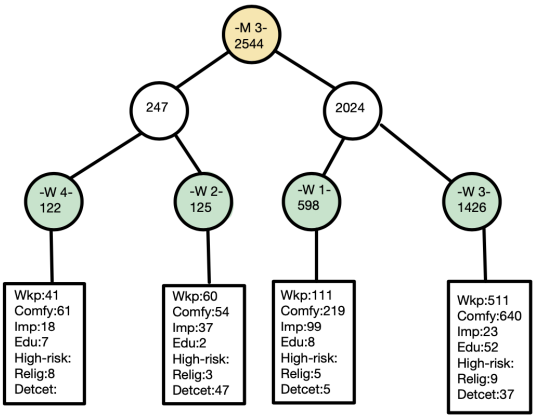
So we use dijestra algorithm to plan the shortest route between each two states. In this way, we can maximize the possibility of each state being threatened by the virus. That is, states on the shortest route are more likely to be infected by the virus carried by passing cases.Because a person can't cross the middle state from one state to another, we can complete the shortest distance planning only by considering the distance relationship between adjacent states.

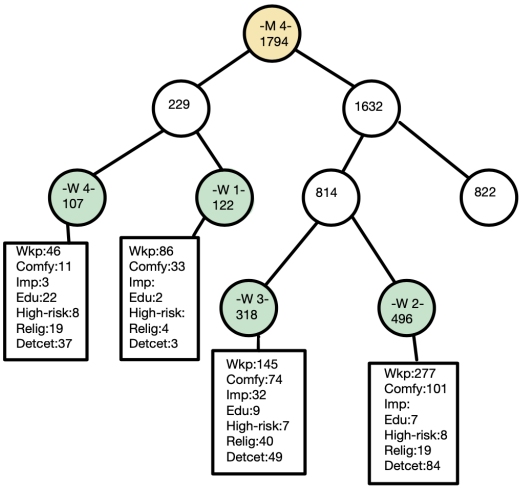
Huffman Code:

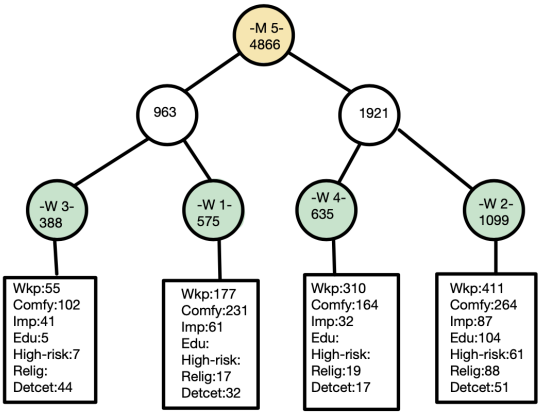


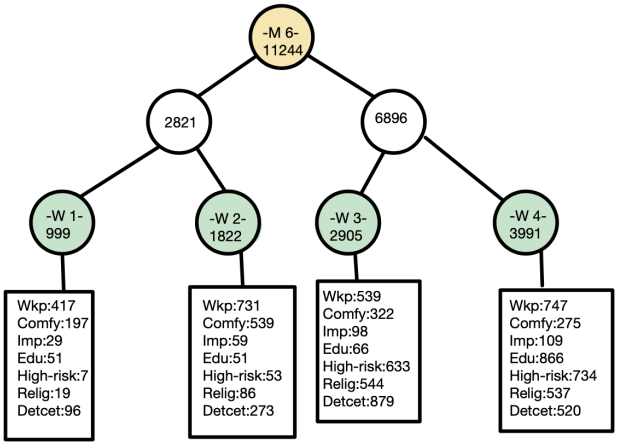


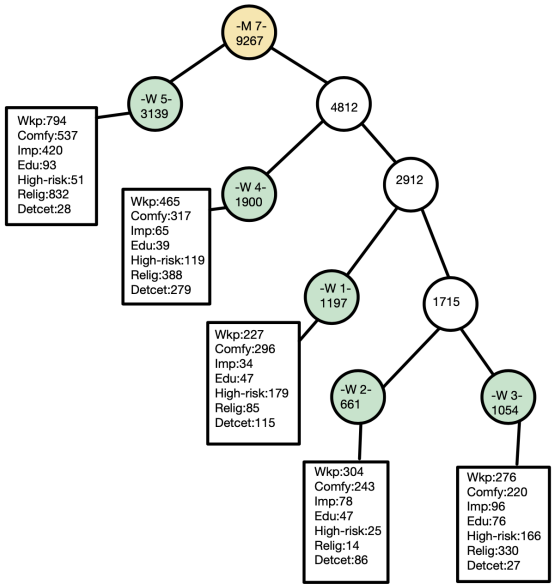


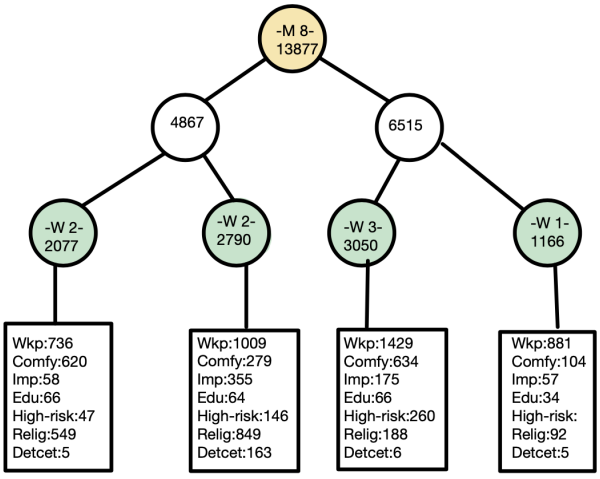


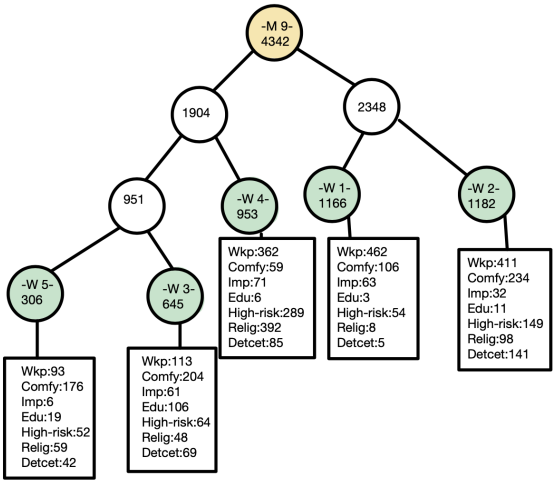


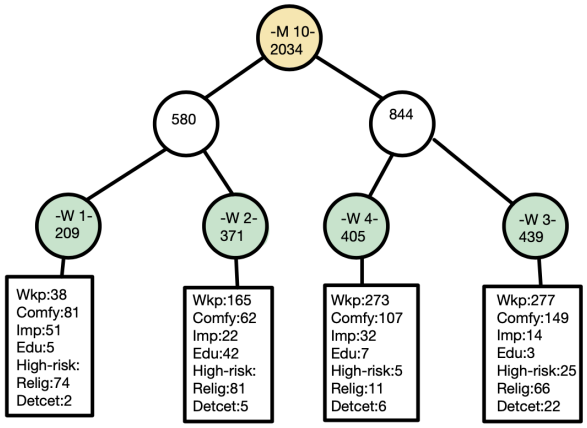


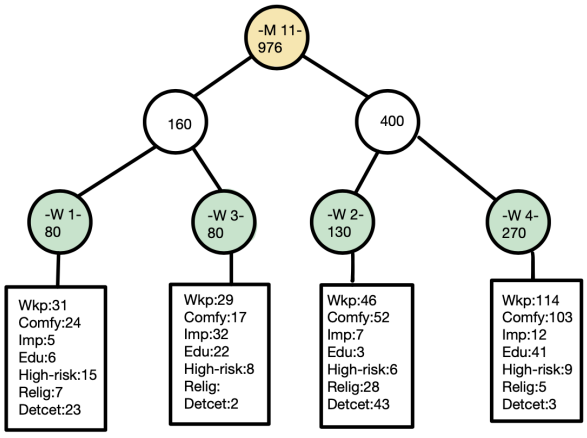

-M 6-
11244
2821
6896
-W 1-
999
-W 2-
1822
-W 3-
2905
-W 4-
3991
Wkp:417
Comfy:197
Imp:29
Edu:51
High-risk:7
Relig:19
Detcet:96
Wkp:731
Comfy:539
Imp:59
Edu:51
High-risk:53
Relig:86
Detcet:273
Wkp:539
Comfy:322
Imp:98
Edu:66
High-risk:633
Relig:544
Detcet:879
Wkp:747
Comfy:275
Imp:109
Edu:866
High-risk:734
Relig:537
Detcet:520

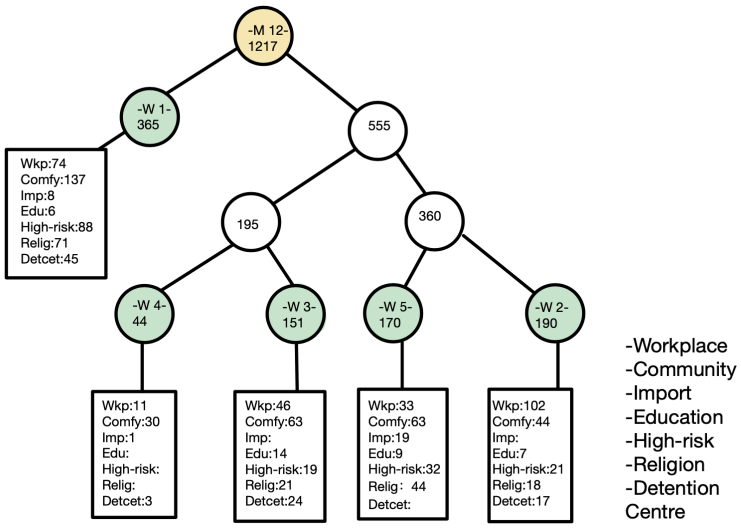

-M 7-
9267
-W 5-
3139
4812
Wkp:794
Comfy:537
Imp:420
Edu:93
High-risk:51
Relig:832
Detcet:28
-W 4-
1900
2912
Wkp:465
Comfy:317
Imp:65
Edu:39
High-risk:119
Relig:388
Detcet:279
-W 1-
1197
1715
Wkp:227
Comfy:296
Imp:34
Edu:47
High-risk:179
Relig:85
Detcet:115
-W 2-
661
-W 3-
1054
Wkp:304
Comfy:243
Imp:78
Edu:47
High-risk:25
Relig:14
Detcet:86
Wkp:276
Comfy:220
Imp:96
Edu:76
High-risk:166
Relig:330
Detcet:27


-M 8-
13877
4867
6515
-W 2-
2077
-W 2-
2790
-W 3-
3050
-W 1-
1166
Wkp:736
Comfy:620
Imp:58
Edu:66
High-risk:47
Relig:549
Detcet:5
Wkp:1009
Comfy:279
Imp:355
Edu:64
High-risk:146
Relig:849
Detcet:163
Wkp:1429
Comfy:634
Imp:175
Edu:66
High-risk:260
Relig:188
Detcet:6
Wkp:881
Comfy:104
Imp:57
Edu:34
High-risk:
Relig:92
Detcet:5


-M 9-
4342
1904
2348
951
-W 4-
953
-W 1-
1166
-W 2-
1182
-W 5-
306
-W 3-
645
Wkp:362
Comfy:59
Imp:71
Edu:6
High-risk:289
Relig:392
Detcet:85
Wkp:462
Comfy:106
Imp:63
Edu:3
High-risk:54
Relig:8
Detcet:5
Wkp:411
Comfy:234
Imp:32
Edu:11
High-risk:149
Relig:98
Detcet:141
Wkp:93
Comfy:176
Imp:6
Edu:19
High-risk:52
Relig:59
Detcet:42
Wkp:113
Comfy:204
Imp:61
Edu:106
High-risk:64
Relig:48
Detcet:69


-M 10-
2034
580
844
-W 1-
209
-W 2-
371
-W 4-
405
-W 3-
439
Wkp:38
Comfy:81
Imp:51
Edu:5
High-risk:
Relig:74
Detcet:2
Wkp:165
Comfy:62
Imp:22
Edu:42
High-risk:
Relig:81
Detcet:5
Wkp:273
Comfy:107
Imp:32
Edu:7
High-risk:5
Relig:11
Detcet:6
Wkp:277
Comfy:149
Imp:14
Edu:3
High-risk:25
Relig:66
Detcet:22


-M 11-
976
160
400
-W 1-
80
-W 3-
80
-W 2-
130
-W 4-
270
Wkp:31
Comfy:24
Imp:5
Edu:6
High-risk:15
Relig:7
Detcet:23
Wkp:29
Comfy:17
Imp:32
Edu:22
High-risk:8
Relig:
Detcet:2
Wkp:46
Comfy:52
Imp:7
Edu:3
High-risk:6
Relig:28
Detcet:43
Wkp:114
Comfy:103
Imp:12
Edu:41
High-risk:9
Relig:5
Detcet:3



3. Planning and execution based on FILA form

FILA Form Week 1

FACTS	IDEAS	LEARNING ISSUES	ACTION	DATELINE
What we know about the task	What do we need to find out?		Who is going to do it?	
Identify suitable visualization approach incorporating graphs that users prefer to visualize the movement of people and how it is related to the spread of covid-19. Make a survey, interview to at least 5 people regarding various visualization techniques to complete this	start data visualization coding. Finished Group Contract,		Xu Wang	29/12/2022
	Finished FILA FORM		Zixuan Wang	
	Make a survey, finish the survey results Plan how to design the web page, Determine the data needed for the project, Write the report to survey result.		Yinrui Zhang	
	finished Big 5 personality test,		ALL	

FILA Form Week 2

FACTS	IDEAS	LEARNING ISSUES	ACTION	DATELINE
What we know about the task	What do we need to find out?		Who is going to do it?	
Design The Idea of how the visualization will be implemented Design the sketch and the algorithm for interface interactivity	Create github Project, start data visualization coding.		Xu Wang	10/1/2022
	Algorithm analysis 1)Understanding the 5 stages of Design Thinking process 2)add new and innovative ideas 3)collect Graph algorithm		Zixuan Wang	
	begin design web coding		Yinrui Zhang	

FILA Form Week 3

FACTS	IDEAS	LEARNING ISSUES	ACTION	DATELINE
What we know about the task	What do we need to find out?		Who is going to do it?	
A working coded graph algorithm and almost complete visualization	1.Draw Map the total number of confirmed cases in each state (map of Malaysia) 2.draw The overlapping histogram shows the comparison of new and cured people 3.draw Kuala Lumpur overlapping histogram shows the number of new and cured people		Xu Wang	17/1/2022
	Algorithm analysis 1)Connect new ideas to data sets. 2)Colab-Graph algorithm		Zixuan Wang	
	finished the web designing upload films and some other tasks on the website.		Yinrui Zhang	

FILA Form Week 4

FACTS	IDEAS	LEARNING ISSUES	ACTION	DATELINE
What we know about the task	What do we need to find out?		Who is going to do it?	
Presentation of completed project Expected output: working code and updated resources in repository, presentation slide The completed interactive visualization to visualize the movement of people and how it is related to the spread of covid-19.	1.Mapped the flow chart of infected people through the States 2.Do data statistics for the drawing of Huffman tree and Dijkstra algorithm		Xu Wang	24/1/2022
	Algorithm analysis 1)Huffman Tree-data process-Sorting 2) Colab-CompleteHuffman tree		Zixuan Wang	
	Complete the coding of Dijkstra algorithm. Through the state column in the clusters file, we can know the several states the infected person passes through, and then we plan the shortest path for his action track		Yinrui Zhang	

2.Design

First of all, I think of using the best pyecharts Library in Python, since interactive visualization is involved. And then based on the requirements, through the data source clusters The field in CSV envisages a national epidemic map of Malaysia, which visually shows the provinces with a large number of infected people, and the color will deepen. Users can move the mouse to preview the total number of infected people in the current state. They can also drag the left adjustment bar to show the distribution of infected people in different sections of the map, and design the clickable function of the map, You can click to enter the designated state and display all the epidemic information of the current state. The information has been divided into groups through the cluster, so you can more intuitively see the epidemic information under each cluster.

Design a broken line chart of data statistics of people's check-in and check-out every half hour on mysejahtera platform every day. Because there are 48 and a half hours in total, it will be very bloated. Therefore, design a scroll bar that can be dragged and zoomed to observe all half-hour data in a fixed page,

4.Overall reflection of project experience

We have gained a lot of practical experience from this project,
Data visualization:
For example, in the data visualization map, a large number of dataframes are processed, and the extracted information cannot be displayed on the map. We replaced one visualization library after another. Finally, we determined to use pyechards to the most visualization library, in which the parameters of the drawing function are not clear at a glance, You need to consult the source code to understand what type of parameters you need, and you also spend a lot of time on data format conversion.
In the horizontal overlapping histogram, the overlapping columns cannot be displayed at the same time. After reading many forums, the final solution is to draw the graph on the basis of one column.

5、Conclusion

The visualization platform novel coronavirus pneumonia novel coronavirus pneumonia epidemic database is built on the basis of the national new crown pneumonia epidemic database. Raw data from

The existing data of relevant government departments and the data published by state organs are filtered, cleaned and integrated to obtain the current detailed epidemic data of various regions. The database includes the national epidemic situation details database and user details database. The national epidemic situation details database includes the statistical epidemic situation data sheet of each province, the detailed situation data sheet of each province, etc. Combined with the general requirements of database content and platform display, the system mainly visualizes the following four information objects.

Big data visualization aims to enable the audience to have intuitive visual thinking on problems by combining with maps, thermal maps, scatter charts and other chart forms, so that the public can "read and understand". The front-end web page adopts the combination of map and visual graphics. With the help of the rich visual graphics library and map elements in echarts technology, it displays the epidemic analysis data in a diversified way, and dynamically visualizes the epidemic data with a draggable time axis, which dynamically shows the content of epidemic information big data and enriches the expression forms of epidemic data. The visualization of the novel coronavirus pneumonia epidemic data has played a vital role in improving the awareness of self-protection and preventing and controlling epidemic situation.

6.Reference

1. Tebé, C., Valls, J., Satorra, P. et al. COVID19-world: a shiny application to perform comprehensive country-specific data visualization for SARS-CoV-2 epidemic. BMC Med Res Methodol 20, 235 (2020). https://doi.org/10.1186/s12874-020-01121-9.

2.[1] Chen Ying et al "Bibliometrics based visualization analysis of New Coronavirus pneumonia (COVID-19) study." Chinese Journal of traditional Chinese medicine 45.10 (2020): 10

3.[1] Fang Zhixiang, Pei Tao, and Yin Ling "Global New Coronavirus pneumonia (COVID-19) epidemic dynamics modeling and visual decision analysis" introduction. Journal of Earth Information Science 23.2 (2021): 1

4.[1] Hu Jun, Teng Shaohua, Zhang Wei, & Liu Dongning (2014). Research on multi classification based on support vector machine and Huffman tree Journal of Guangdong University of Technology.