Optimizing a Healthcare Network for Improved Service Delivery

Muhammad Ammar Baig

June 2020

Table of Contents

1 Introduction

The Washington State Health Ministry would like to optimally upgrade staff or resources in existing facilities across Washington so that they can allocate resources to where they are most needed, based on demand for services in different geographic regions/areas. The document proposes the following solutions to address this problem

- Optimizing the resource allocation process by increasing the resources capacity in some facilities and decreasing capacity in others to allocate resources where they are most needed, hence making services more accessible to patients.
- Calculating the travel time from one area facility to another, and visualizing the distance on map so that people only travel to their nearby facility, therefore minimizing the time and distance cost.

1.1 Problem statement

The unavailability of staff or resources in an area's facility causes the population of that area to travel to the facilities located in other areas, in order to avail medical services. This results in an increase travel time and distance. Optimally upgrading the staff or resources in existing facilities across Washington so that the resources are allocated to where they are most needed, based on demand for services in different geographic regions/areas will resolve this problem

2 Assumptions

Following are the assumptions made to solve this problem

- 75 percent of the population in an area will require medical services however to only assume a certain percentage of a population is simplistic in nature, therefore an impact of poverty rate in an area and the percentage of population that may need more health services such as young kids under the age of five and old age population over the age of seventy are taken in to account. It is assumed that populations living in poverty have poor health and living conditions and young kids and old age people requires more medical attention.
- It is also assumed according to the rule of thumb that 1:2808 is an acceptable staff to patient ratio in order to not over burden the staff

3 Data gathering, handling, cleaning, processing

Area Zip codes along with it's population data is scraped from the website "Washington Demographics by Cubit" containing US Census data. The data is extracted from csv files and converted into data frames using python Pandas library. The population data consisted of three columns

- Washington Zip codes by Rank
- Zip Code
- Population

The "Washington Zip codes by Rank "column is dropped since this feature is of no importance to us, a data frame is made with only area zip codes given in our problem along with it's respective population. Data frame with columns "Facility ID" and "Facility staff count" is created and then merged with the existing data frame to create a table given in the test example document.

The data for poverty status and age is collected by American Community Survey, it is downloaded from US census bureau website. Poverty rate is extracted for each area from poverty status data and a "poverty rate" feature column is created, similarly different category of age data is extracted for each area, and multiple feature columns of age category are created which are as follows

- Age <5
- Age 70-74
- Age 75-79
- Age 80-84
- Age >85

The above age columns are added to form a feature "priority population(%)"which is a percentage of population that requires extra medical care based on our assumption.The population column was then multiplied by 0.75 to calculate 75% of the total population for each area based on the assumption mentioned in above section.The output of the entire data preprocessing step is a "Structured_data.csv" file containing six features as follows

- Facility ID
- Zip_Code
- Population
- Facility Staff Count
- Poverty rate
- Priority population (%)

4 Proposed Solution

The proposed solution for the given problem is to allocate staff or resources based on the percentage of the population living in an area and using the "Poverty rate" and "Priority population " features to calculate the population in each area which is more vulnerable to health risks and add them to the population of an area in order to take into account poverty status and age factor based on the assumption that they require more medical services.
The above approach can be explained by an example, suppose there are three areas A,B and C with population 20,40 and 30 and staff availability as 6,8 and 5 respectively. We calculate the sum of the population for all three areas as 20+40+30=90 and then find population percentage for each area which is 22.2% for area A,44.4% for area B and 33.3% for area C. Now we calculate total staff which is 6+9+5=20.The staff is allocated according to the population percentage for each area.
Staff allocated to area A=20*22.2/100 $\approx$ 4
Staff allocated to area B=20*44.4/100 $\approx$ 9
Staff allocated to area C=20*33.3/100 $\approx$ 7
The formula used to calculate estimated population at health risk for an area is as follows
Est Pop at health risk $\approx$(poverty rate + priority population)-poverty rate of Americans over age of 70.

The Python library Geopy is used to calculate the distance between facilities, distance pivot table is created to make the data more readable and the distance between facilities is visualized on map using Folium library.

5 Average Travel Time

"Google Maps Distance Matrix API" is used to calculate the travel time that each person would need to travel from one facility to another. The API provides the service to calculate travel distance and time taken to reach the destination along with recommended routes between the start and end points as calculated by the Google Maps API, and returning journey duration and distance values for each pair. An optional parameter mode is specified with values returning for different modes of transportation which are bicycling, bus, driving, rail and walking. Travel time pivot table is created to make the travelling time data more readable

```
# assigning different modes of transit
modes = ['driving', 'walking', 'bicycling', 'transit']
transit_modes = ['bus', 'rail']


url ='https://maps.googleapis.com/maps/api/distancematrix/json?'

matrix = []
# requesting distance matrix responses for different transit modes
for mode in modes:
    transit = None
    # units = None
    response = {}
    print("Mode: " + mode)
    if(mode == "transit"):
        for transit in transit_modes:
            print("Transit: " + transit)
            response = request_url(url, "|".join(source), "|".join(dest), mode, api_key, transit)
            mode_matrix = distance_matrix(response, source, dest, transit)
            matrix.extend(mode_matrix)
    else:
        response = request_url(url, "|".join(source), "|".join(dest), mode, api_key, transit)
        mode_matrix = distance_matrix(response, source, dest, mode)
        matrix.extend(mode_matrix)
    time.sleep(5)
```

5.1 Other APIS

- "Distancematrix.Ai" can also be used to compute travel time and distance, it's trial version is available for free.
- "TravelTime" API testing is free and pricing is much lower than Google distance matrix API as it's charged per search (e.g. travel time from origin to 1000s of points vs. origin-destination pair like Google Distance matrix).It handles driving distances and driving times.
- "Openroute service" API Time-Distance Matrix service can be used , the trial version is free for use.

6 Population density

READING ZIP CODES BY POPULATION

```
import pandas as pd

df = pd.read_html('https://www.washington-demographics.com/zip_codes_by_population',header=0)
population_df = df[0]
population_df.head()
```

	Rank	Zip Code	Population
0	1	99301	78767
1	2	98052	67779
2	3	98012	64417
3	4	98682	58178
4	5	98208	57306

COLLECTING POVERTY STATUS DATA FOR AREA ZIP CODE

```
[ ] #Read 98007 poverty_status.csv
    from google.colab import files
    uploaded = files.upload()
```

```
Choose Files  No file chosen   Upload widget is only available when the cell has been executed in the current brows
Saving 98007 poverty_status.csv to 98007 poverty_status.csv
```

```
import io
povertystatus_df = pd.read_csv(io.BytesIO(uploaded['98007 poverty_status.csv']))
povertystatus_df.head()
```

The population data is collected from "Washington Demographics by Cubit" containing publicly available current US census data .United States Census Bureau website can also be used to avail the publicly available population data for area zip code.

7 Optimization

The optimization technique used to allocate staff or resources is based on allocating resources depending on the percentage of the population living in an area. The sum of population of all areas is calculated along with sum of total staff available. The percentage of population is then calculated for each area and staff is allocated based on that percentage, in this way an area which comprises of large percentage of population will be allotted more staff and area with less population will be allotted less staff in order to maintain the staff to patient ratio in all the facilities.

The "poverty rate" and "priority population (%)" features are used to further optimize the staff allocation process. The percentage of population that is more vulnerable to health risks and may need more health services is calculated

using the below formula
Est Pop at health risk = (poverty rate + priority population)-poverty rate of Americans over age of 70

To solve this problem we assumed that 75% of the total population of each area will require health facilities. The value calculated using the above formula for each area is added to that population in order to take account of impact of poverty rate and age of the population in an area for resource allocation process. The features needed for the optimization technique is listed as below

- Population
- Staff count
- Poverty rate
- Priority population

```
12] US_PR_over_age70=0.092
    df['bias']=df['Population']*((df['Poverty rate']/100)+(df['priority population(%) ']/100)-US_PR_over_age70)
    df.head()
```

```
[14] # find percentage of the population of each area
population_percentage_in_area = []
total_population = df['Population'].sum()
total_staff = df['Facility Staff Count'].sum()
for i in df['Population']:
    p = (i / total_population) * 100
    population_percentage_in_area.append(p)

# convert number of staff according to the population percentage in each area

staff = []
for i in df.index:
    s = (population_percentage_in_area[i] * total_staff) / 100
    staff.append(round(s))
# Display Population Percantage in each area & converted staff according to it
print(population_percentage_in_area)
print(staff)
```

```
[20.894534320824196, 25.10654143737846, 10.237908064048987, 32.40266457031735, 11.358351607431008]
[39.0, 47.0, 19.0, 61.0, 21.0]
```

```
new_facility_with_updated_staff = df.drop(['Facility Staff Count', 'staff to patient ratio'], axis = 1)
new_facility_with_updated_staff['New Staff Count'] = staff
new_facility_with_updated_staff['New staff to patient ratio'] = round(new_facility_with_updated_staff['Population'] / new_facility_with_updated_staff['New Staff Count'])
new_facility_with_updated_staff
```

	Facility ID	Zip_Code	Population	Poverty rate	priority population(%)	New Staff Count	New staff to patient ratio
0	A	98007	25250	10.2	14.4	39.0	647.0
1	B	98290	30340	6.9	14.7	47.0	646.0
2	C	98065	12372	1.3	14.0	19.0	651.0
3	D	98801	39157	11.4	18.4	61.0	642.0
4	E	98104	13726	23.3	15.3	21.0	654.0


```
[14] # find percentage of the population of each area
population_percentage_in_area = []
total_population = df['Population'].sum()
total_staff = df['Facility Staff Count'].sum()
for i in df['Population']:
    p = (i / total_population) * 100
    population_percentage_in_area.append(p)

# convert number of staff according to the population percentage in each area

staff = []
for i in df.index:
    s = (population_percentage_in_area[i] * total_staff) / 100
    staff.append(round(s))
# Display Population Percantage in each area & converted staff according to it
print(population_percentage_in_area)
print(staff)
```

```
[20.894534320824196, 25.10654143737846, 10.237908064048987, 32.40266457031735, 11.358351607431008]
[39.0, 47.0, 19.0, 61.0, 21.0]
```

```
new_facility_with_updated_staff = df.drop(['Facility Staff Count', 'staff to patient ratio'], axis = 1)
new_facility_with_updated_staff['New Staff Count'] = staff
new_facility_with_updated_staff['New staff to patient ratio'] = round(new_facility_with_updated_staff['Population'] / new_facility_with_updated_staff['New Staff Count'])
new_facility_with_updated_staff
```

	Facility ID	Zip_Code	Population	Poverty rate	priority population(%)	New Staff Count	New staff to patient ratio
0	A	98007	25250	10.2	14.4	39.0	647.0
1	B	98290	30340	6.9	14.7	47.0	646.0
2	C	98065	12372	1.3	14.0	19.0	651.0
3	D	98801	39157	11.4	18.4	61.0	642.0
4	E	98104	13726	23.3	15.3	21.0	654.0

8 References

1. A. A. Ansari, S. Chakraborty, S. Aich, B. S. Kim and H. Kim, "Optimization of Healthcare Network for improved service delivery," 2020 22nd International Conference on Advanced Communication Technology (ICACT), Phoenix Park, PyeongChang,, Korea (South), 2020, pp. 314-318, doi: 10.23919/ICACT48636.2020.9061358.
2. https://www.washington-demographics.com/
3. https://developers.google.com/maps
4. https://pypi.org/project/folium/
5. https://pypi.org/project/geopy/
6. https://pandas.pydata.org/
7. https://www.census.gov/
8. https://numpy.org/
9. https://fas.org/

9 Source code link

`https://tinyurl.com/ybs5ohml`