

COVID-19 Website Main Contents:

This website is used to exhibit our hierarchical risk assessment of COVID-19 over the map. On the website, a dynamic map is preferred to be shown through map APIs, e.g., Google Map API or others. Certain markers may need to be inserted on the map to exhibit the risk level for certain areas. Other than this, a heat map needs to be given to show the risk distribution on the map. The website should be daily updated based on the up-to-date official dataset. Below are several potential map tools and potential programming packages may be needed for the website.

Google Map API: <https://cloud.google.com/maps-platform>

Google Map Heatmap Instructions: <https://developers.google.com/maps/documentation/javascript/heatmaplayer>

eSpatial (Heatmap creator): <https://www.espatial.com/how-to-guides/create-a-hotspot-heat-map>

Potential programming package needed:

Django: <https://www.djangoproject.com/>

BeautifulSoup4: <https://www.crummy.com/software/BeautifulSoup/>

MySQL: <https://www.mysql.com/>

Others: <https://www.hongkiat.com/blog/python-packages/>

There are a total of 7 main functions should be achieved in this website, which are listed in the following. For each function, we list the required input and output of the function. Most functions' inputs can be acquired from our risk assessment model or online data source. For function 2-7, the outputs are the content needed to be shown on the website.

Main Functions:

1. **Data Crawler**

Fetch COVID-19 data online from authoritative sources, e.g., CDC, WHO. The main required data for website are confirmed case, death case for metropolitan areas in U.S. Each metropolitan area is assigned by a coordinate (Latitude, Longitude). The function output is as follows.

Output: Data list of [*Confirmed_Case_Number*, *Coordinate*(*Latitude*, *Longitude*)]; Data list of [*Death_Case_Number*, *Coordinate*(*Latitude*, *Longitude*)].

Note that the fetched data is used in the following functions. However, the following functions can be finished first, since sample data can be used for testing. Below are some recommended links for data fetching.

US Metropolitan Daily Cases with Basemap:

<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/5B8YM8>

United States Cities Database:

<https://simplemaps.com/data/us-cities>

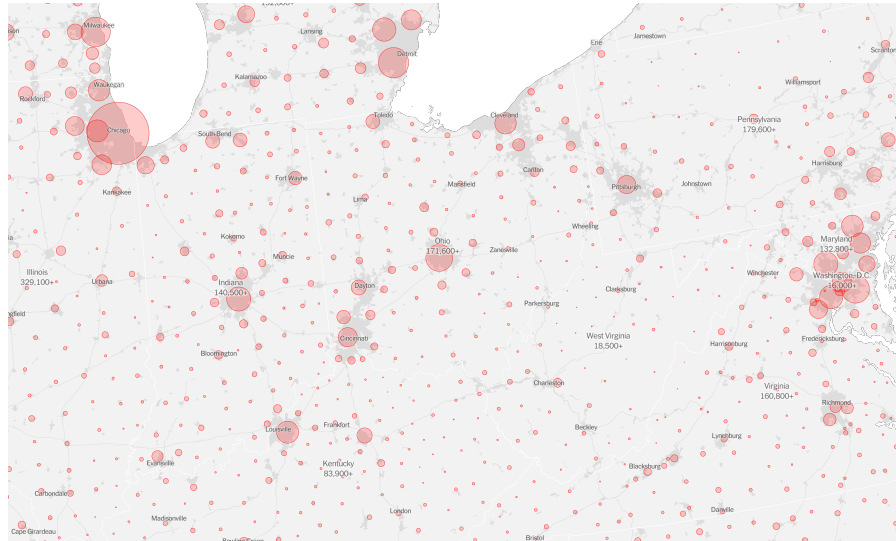
2. **Confirmed Case Map**

The confirmed case map function basically shows the confirmed case number for each city (metroplitan) area given truth data crawled from CDC and WHO.

Input: Data list of [*Confirmed_Case_Number*, *Coordinate(Latitude, Longitude)*].

This data list is truth data crawled from authorized source.

Output: A map showing the confirmed case number of cities. There is a sample is as follows.



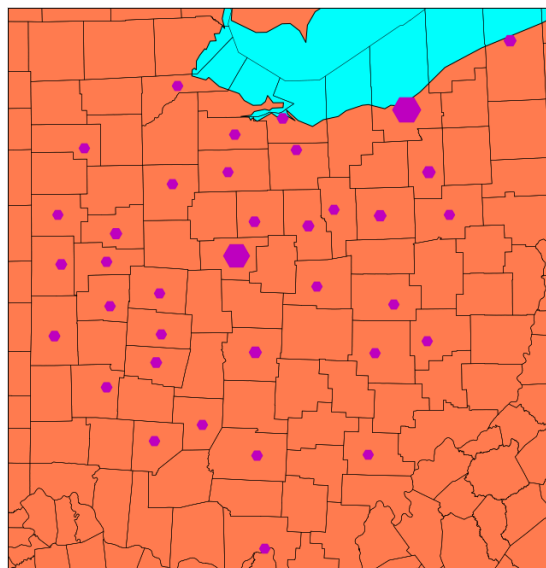
The size of circle represents the number of confirmed case.

3. **Labeled Risk Map**

This map function exhibits our manually labeled risk for each city.

Input: Data list of [*Risk_Label*, *Coordinate(Latitude, Longitude)*].

Output: A map showing the labeled risk of cities. A toy sample is as follows.



The size of the marker represents the labeled risk.

4. **High-level Risk Prediction Map**

This map function exhibits our predicted risk for each city.

Input: Data list of $[Predicted_Risk, Coordinate(Latitude, Longitude)]$.

Output: A map showing the labeled risk of cities. This map is in the same form of labeled risk map. The size of the marker represents the predicted risk.

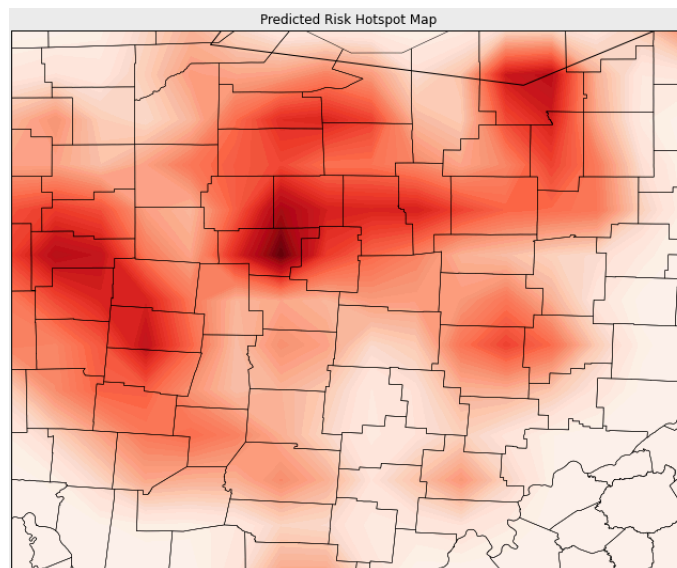
5. **Low-level Risk Assessment Map (Heat Map)**

This map function exhibits a predicted risk over the whole map. In this part, the map is segmented into a $n \times n$ grid, where each block of the grid is assigned by a coordinate. The front-end website need to query the back-end low-level prediction function for the predicted risk for each coordinate.

Query sent to back-end: List of $[Coordinate(Latitude, Longitude)]$.

Return: List of $[Predicted_Risk, Coordinate(Latitude, Longitude)]$.

Output: A map showing the predicted risk over the grid. A sample is as follows.



The color depicts the value of predicted risk.

6. **High-Level Validation Score**

A curve shows the high-level validation score.

Input: List of $[High_Val_Score]$.

Output: A curve of high-level validation score.

7. **Low-Level Validation Score**

A curve shows the low-level validation score.

Input: List of $[Low_Val_Score]$.

Output: A curve of low-level validation score.

Basic Web Layout:

Main Page:

Number of Confirmed Case
(Button)

Risk Labels
(Button)

Risk Estimation (High Level)
(Button)

Risk Estimation (Heat Map)
(Button)

Map View
(Confirmed Case / Risk Labels / High-level Estimation / Heat Map)

Risk Score Estimation:

Location:

Coordinates

(Showing the coordinates of the pin on the map)

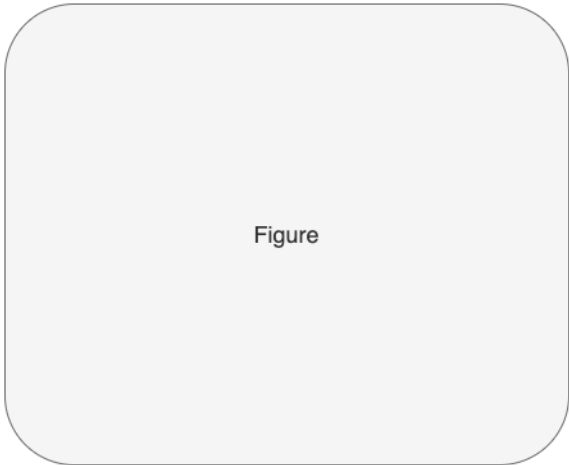
Risk Score:

Float Number

(Display)

Performance Validation Page:

Model Learning Performance:

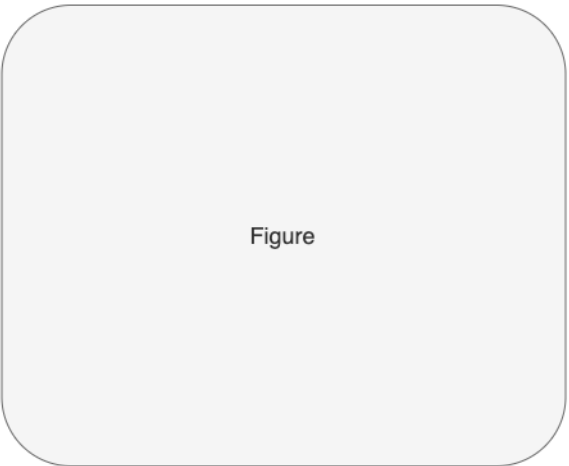


Score:

Training: Float Number
Testing: Float Number

[\(Display\)](#)

High-level Validation:



Score:

Float Number

[\(Display\)](#)

Low-level Validation:

Score:

Float Number

[\(Display\)](#)