Report Date: 11/18/2022
To: ematson@purdue.edu, ahsmith@purdue.edu and lee3450@purdue.edu
From: Chasing FOX

- Daye Kim (rlaek4793@khu.ac.kr)
- Hyun Roh (yeshyun@khu.ac.kr)
- Heewon Jeong (jhjmo0719h@chungbuk.ac.kr)
- Jihu Yang (zihooy@handong.ac.kr)
- Juwon Baek (21700340@handong.ac.kr)

Summary

- In the algorithm part, the risk score that is the result of the data analysis was applied to the graph.
- In the data analysis part, the problem of feature's scale difference and multicollinearity was solved through robust scaler.
- In the Back-end part, API related to the user was updated. The API related to running python code was connected with the Google Drive API and the MySQL DB.
- In the Front-end part, The search page and a route page have been developed. Redux/Encrypted storage were recreated to communicate effectively with the backend server.

What Chasing FOX completed this week:

- **Data analysis**
  - Two problems for defining the riskiness score using regression models were solved.
    - Problem 1: Scale of each factor was different.
    - Problem 2: Multicollinearity issue, which means the feature for the regression model has high correlation although all the features are assumed to be independent in regression analysis.
    - To solve the problems, we rescaled the feature with a robust scaler, which can minimize the effect of outlier and checked Variance Inflation Factors(VIF) value of features.
  - Check the result to predict 2019 crime density data using 2018 crime density data and urban factors that have high correlation with 2019 crime density data (Correlation coefficient value over 0.5).
    - Compared to the result using only 2018 crime density data, the result using 2018 crime data and urban factor has higher performance. The highest former result has 128.36 MAE value and the highest latter result has 4.72 MAE value.
  - Data needed to show the statistic value to the user is created
    - Zip Code number
    - Polygon vertex of Zip Code
    - Crime type that has high crime density then other places
      - Compared to the average of all point's crime density of each crime type where the point is included in Zip Code polygon, the crime type that has higher crime density will be added.
    - Number of the crime occurred in the Zip Code for each crime type.
- **Algorithm**
  - The risk score calculated from the data analysis part was imputed in the algorithm. By considering the mean value of risk score of the starting and end node, risk score in edge was calculated.
  - 9 or fewer nodes were selected with the consideration of the number and length of each edge.
- **Back-end**

    - About the API related to user, sign in API, sign up API and user account update API was developed. The exception and error message was also developed to check existence and correct password.
    - The Flask API related to running python code was updated. The Google Drive API was connected with the Flask API to upload [1] and download [2] the user's weight graph through the Google Drive.
    - The Flask API connected with the MySQL DB. The API needs to control the file information table including file ID of the weight graph to do PUT(in profile page) and POST(in sign-up page) requests.
- **Front-end**
    - The main page and server were connected, and the current location and destination were saved using Redux.
    - The values retrieved from the server were saved using Redux, EncryptedStorage repository.
    - The structure of the data has been redefined in a format that facilitates communication with the backend.
    - In communication with the data analysis part, Data for drawing polygons has been preprocessed.
    - The Bottom Sheet Library has been added. Because it is convenient to reuse.

Things to do by next week

- The code related to data analysis has to be combined with the Back-end API for the application.
- Analyze the result of the algorithm using the total riskiness and distance of route.
- Combine the riskiness score and distance and make the new weight to each edge.
- The Polygon function will be developed, and the entire ux/ui will be redesigned.

Problems or challenges:

- In the algorithm part, the decision about selecting the 9 nodes for the Google Map API is still progressing between considering the uniform intervals or numbers.
- In the data analysis part, the way to combine the risk and distance is not decided. Therefore, searching the material is needed.
- Since there are two back-end servers, there was a problem of when to connect to the server at the front-end.

References

[1] "Upload file data." Google Drive for Developers.
https://developers.google.com/drive/api/guides/manage-uploads
[Accessed: Nov 16, 2022].
[2] "Download files." Google Drive for Developers.
https://developers.google.com/drive/api/guides/manage-downloads#python [Accessed: Nov 16, 2022]