Chicago Urban Planning Project Report

Hanna Yukhymenko
ETH Zürich
Zürich, Switzerland
hyukhymenko@student.ethz.ch

Mihaela Demireva
ETH Zürich
Zürich, Switzerland
mdemireva@student.ethz.ch

Olga Ovcharenko
ETH Zürich
Zürich, Switzerland
oovcharenko@student.ethz.ch

ABSTRACT

This report presents *Chicago Urban Planning*[1], a dashboard for exploration of the current greening situation in Chicago and for predicting the area of new parks to be built. The dashboard is meant to be used by urban planners and the general public. Lay users can interact with the Chicago community areas map and explore chosen community areas, and compare life quality metrics for similar community areas. The future part is related to urban planners. It can help them in their decision-making process regarding where and how many parks or what size of green area to build.

1 INTRODUCTION

More and more people want to live in places with many green areas where the life quality indicators are corresponding to their preferences. Our application is for Chicago community areas, and it is meant to help urban planners to build parks where needed. A dashboard is intended to help general users to find the best community area for living in Chicago based on their preferences.

2 USERS AND TASKS

We consider two types of users - urban planners, who are domain experts, and general users, so-called lay users. The main goal for urban planners is to build parks where needed based on different life quality and area metrics. The tasks that our application solves for this type of user are the following:

(1) Future Tab - How many parks should be build considering change of life quality?
(2) Future Tab - What is the expected ratio of total green area to the total area considering the % -improvement of a chosen life quality indicator(s)?
(3) Current Tab - How big is the total green area in each community area?
(4) Current Tab - How did diverse life quality metrics change with years, and how are they related to other metrics?
(5) Current Tab - What are the values for a life quality metric for a chosen community compared to other communities?

On the other hand, the main goal for lay users is to explore the community area or to find a community area that matches their preferences. Also, a goal for the lay user might be to see if it is worth it to move to another community area by simply comparing the life quality indicators and the park data for similar areas. Lay users are mainly intended to use the Current Tab. The tasks that we are trying to solve for them are similar to the ones for the Domain Experts in the Current Tab:

(1) How big is the total area of green parks in the current/desired community area of living for the lay user?
(2) How many parks are there in the current/desired community area of living for the lay user?
(3) How different is the desired community area from the current community area of living in terms of life quality indicators and number/total area of parks?

3 ARCHITECTURE

In this section, the data, data collection, pre-processing, and the model are discussed.

3.1 Frontend backend

Our dashboard is designed as a two-tier architecture. We have server (backend and data) and client (frontend) that communicate with each other. The server represents data access logic and data, client - presentation logic. The client sends HTTP POST requests to request data on demand or requests it once and stores it as in the map case. The server gets a request and returns data. Once the backend is started, data is read into pandas.DataFrame (from CSVs) and is stored using a DataCollection class instance. For each data file, we implement a few resources: Classes that inherit flask_restful.Resource. Each resource returns serialized JSON object. Below is an example of a life quality feature resource:

```
class LifeQualityResourceByFeature(Resource):
    def get(self, feature_name):
        """GET request handler"""
        return json.loads(lq_df[feature_name]
                .to_json(orient='records'))
```

3.2 Data

Based on the data available online, we collected datasets from the City of Chicago's open data portal [1]. One can access city data, find facts about their neighborhoods, visualize this information, or download the data for further analysis.

For our interactive task, we collected data about green areas [2] [5] (parks and green roofs), and diverse life quality metrics [3, 4] that characterize community areas and their inhabitants. Different metrics are for one year or within a time interval, this depends on the feature. Table 1 shows a sample of features.

As a pre-processing step, we detect missing values (MV) such as NAs and replace them with sklearn.SimpleImputer. After exploring data distributions (see Figure 1), we decided not to remove outliers but we performed clustering to group samples.

Figure 2 shows clusters with marked *Rogers Park*. We use Principal Component Analysis (PCA) to reduce data to two dimensions, normalize principal components, and apply K-Means with K=5 to assign samples to clusters. Neither UMAP nor DBSCAN did improve clusters. Clusters are later used for visualization and comparison

[1] Authors are sorted alphabetically.

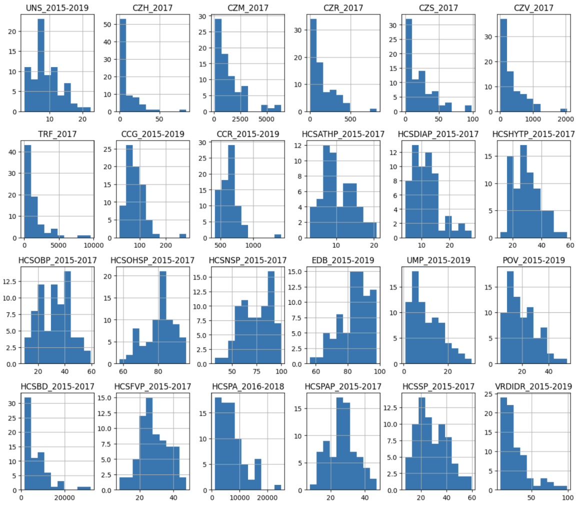


Figure 1: Data distribution of a few samples

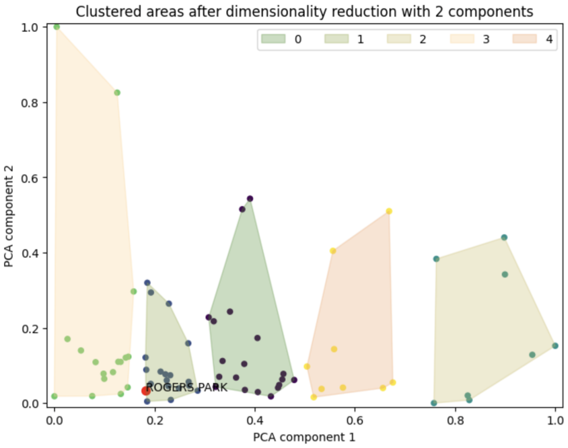


Figure 2: Data clusters with marked *Rogers Park*

Table 1: Life quality features sample

Feature name	Description
HCSOBP_2015-2017	Adult obesity rate, 2015-2017
HCSPAP_2015-2017	Adult physical inactivity rate, 2015-2017
CZD_2017	Drug abuse (crimes), 2017
UMP_2015-2019	Unemployment rate, 2015-2019
VRDIAR_2015-2019	Diabetes mortality rate , 2015-2019
VRCAR_2015-2019	Cancer mortality rate , 2015-2019

of samples. Therefore, we create a new feature with cluster assignments. We tried normalizing and smoothing data, but it did not lead to any improvements in terms of clustering and ML model performance. Additionally, for the ML model, we engineer new features: We calculate new dimensions which describe the count of parks and their average area from existing data.

3.3 The ML Pipeline

In this part we start with preparing the data for model training and visualization - this includes data imputation, clustering, and feature engineering as described in Subsection 3.2. For missing values imputation, we use SIMPLEIMPUTER from SKLEARN and KMeans for clustering similar parks, which is used later for better visual data analysis. We also calculate new variables which describe the count of green objects and their average area from existing data. In the end, we have pre-processed datasets with new features which allow prediction variability and flexibility for FUTURE TAB of the dashboard.

For predicting the future greening of Chicago areas we are using a XGBOOST model. We choose this model because it is explainable and computationally efficient for training and predicting on the fly. First, we find the optimal model parameters using 5-fold GRIDSEARCHCV with mean squared error as optimization goal, then the best parameters are saved and used for training the model on a full training set. The new model is saved and afterward loaded from the backend to the frontend during the user prediction part. In the dashboard, the user inputs an improvement ratio for desired metric(s) (in %) in a specific community area. User input is used as test data for the model. We get a prediction value from the backend. On the backend side, user input and the entry for the chosen community are extracted, and life quality features that the user wants to improve with greening are replaced. This row is treated as a test sample for prediction and afterward, the output values from the model are sent back to the frontend (see Figure 3).

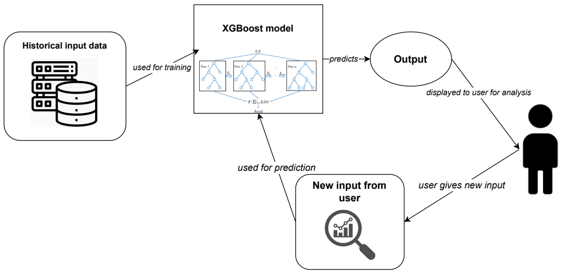


Figure 3: Machine learning pipeline structure

As independent variables, we use 16 life quality metrics. We use the newly calculated ratio between the total and green area of the community area as a target variable. The reason is that it is a relative metric, and we can calculate several output values for the user out of this predicted feature, such as the number of parks/green roofs required to build, and the expected total green area to achieve desirable life quality metrics.

For transparency and explainability, we calculate Shapley values in the backend after predicting with new data. It describes each feature's contribution to the predicted output value. Many life quality features are actually having a negative impact, and it is good to decrease them. Therefore, after predicting we display Shapley values for features, so the user would gain deeper insight into green area contribution to different parts of life quality in the city (see Figure 4).

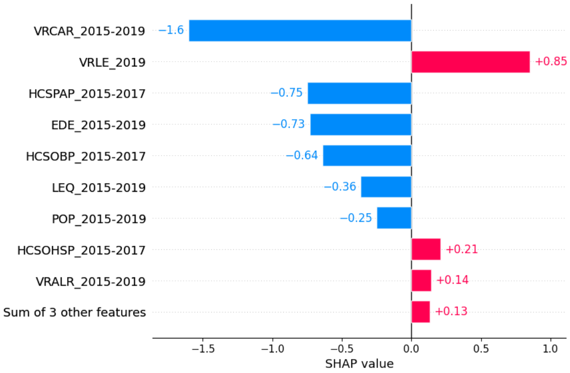


Figure 4: Shapley values for new prediction

4 INTERACTIVE DASHBOARD

In this section, we describe the workflow and design choices.

4.1 Story line & Workflow explanation

We have two main user types - Urban Planners and General Users. The Urban Planners can explore the community areas by clicking on each of them on the map. A pop-up will show up with information about the total area of green areas and the number of parks.

If the Urban Planners want to further explore, they can choose the community area, in which they are interested, either by choosing it from the dropdown menu or by clicking on the map. From the checkbox menu users can choose up to three life quality metrics to explore. If more than three indicators are chosen, an alert message shows up. The line plot for the chosen metric is shown to compare the chosen community area with similar areas within a time interval. Similar areas are chosen randomly from the same cluster. Therefore, the same chosen area can be plotted with different "similar" areas.

If data is available for one year only, stacked bar plots are shown instead. They present the chosen life quality indicator compared to relatively important life quality metrics.

The grouped bar plot presents life expectancy, traffic intensity, and overall health status in the chosen and similar community areas.

To choose more or less metrics visualizations, user can click or unclick checkboxes respectively. It is possible to change metrics dynamically, and the plots update accordingly. In the CURRENT TAB the workflow is similar for both types of users. The only difference is that the lay users are intended to explore a particular area, and FUTURE TAB is not in the area of their interest. An example of a user workflow in the Current tab is shown in Figure 5. The visualization is intended to help to understand the workflow [6].

The intended user of the FUTURE TAB is the Urban Planner. They can interact with the map by clicking on it for basic information. Furthermore, the Urban Planners can define what % of improvement they want to achieve in a particular life quality metric by using the blank space in front of every life quality metric. As shown in Figure 6. They can choose the community area of interest again either by clicking on the map or by choosing it from the dropdown menu. Based on their input the Urban Planners will see the prediction: How many parks should be built, and what is the expected ratio of green area to total area in the chosen community area. Together with the prediction, the bar plot with Shapley values appears. It shows the feature importance and contribution of certain features to the model.

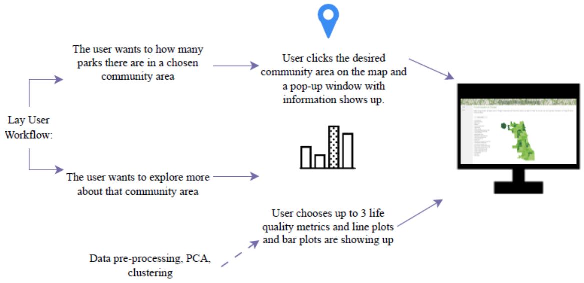

The user wants to how many parks there are in a chosen community area
User clicks the desired community area on the map and a pop-up window with information shows up.
Lay User Workflow:
The user wants to explore more about that community area
User chooses up to 3 life quality metrics and line plots and bar plots are showing up
Data pre-processing, PCA, clustering

Figure 5: Urban Planner Workflow in the Current Tab

There is an INTROJS tutorial for both tabs. Every user sees it once opening the tab for the first time. It can be skipped. A tutorial walks the user through every possible action and informs how a particular part of the dashboard can be used.

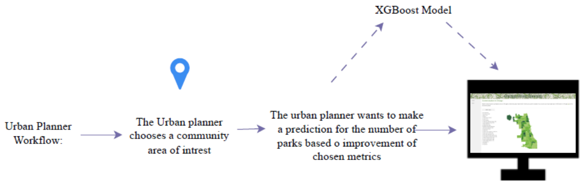

XGBoost Model
Urban Planner Workflow:
The Urban planner chooses a community area of intrest
The urban planner wants to make a prediction for the number of parks based o improvement of chosen metrics

Figure 6: Urban Planner Workflow in the Future Tab

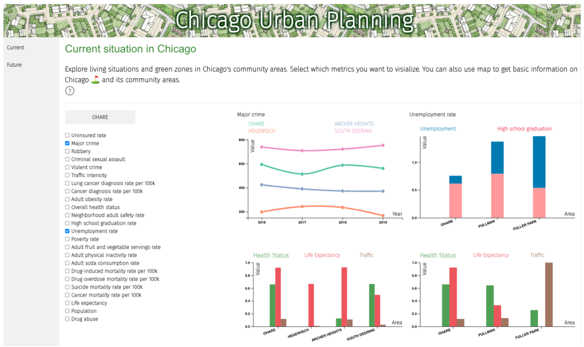
4.2 Design choices & Contributions

In this section, we describe why we made certain design choices.

- **Welcome Tab** is shown as a user opens the page. It gives a general idea of what the purpose of the dashboard is.
- **Two Tabs for different purposes** are designed to separate the workflow into two parts: Exploration and Decision Making. One tab is responsible for each. It makes it easier for the user to choose what one wants to do based on personal preferences.
- **On-boarding tutorial** is provided since we want to ensure that the user will not be lost in the dashboard. It smoothens the overall user experience and avoids confusion. Additionally, the user can hover the mouse over the question icon to get more details.
- **Map of Chicago** is at the community areas granularity (see Figure 8). It improves the decision making process of the user. At first look, as the map is choropleth the user can quickly notice what is the green area size in each community area. Since the application is for Urban Planning, we decided that the most intuitive color for that is green. If a user is having the dilemma of whether to move to a particular community area, one can see where it is on the map, and how far away it is from the current community of living.
- **Interaction between the dropdown menu and the map** is made for the user's convenience. Whenever a user chooses

a community area from the map, it will automatically be selected on the dropdown menu as well.

- **Diverse plots** are generated as life quality indicators are chosen: Line, stacked bar, and grouped bar plots (see Figure 7). We believe that plots are the easiest way for users to perceive and analyze information. We compare metrics with each other, but also between community areas, such that the user can make the best decision possible.
- **Checkboxes** are used to choose which metrics to visualize. They are intuitive, and several metrics can be chosen in a short time.
- **Only three life quality metrics** can be chosen. The amount of information a user can perceive and can be presented on the screen is limited. We do not want the user to be confused and overwhelmed with 20 different plots. The dashboard looks better if there are not too many elements. Moreover, we wanted to avoid scrolling as much as possible. To make it easier for the user, the pop-up alert window shows up if a user chooses more than three features.
- **Limited text boxes input** helps the user to give the correct input: Floating point number in [0, 1].
- **Glitter as an entertainment** is a small "surprise" for the user. Whenever the user is tired from exploring community areas or has trouble making the decision where to move, one can always click on the header, which will (de)activate glitter on the whole tab.

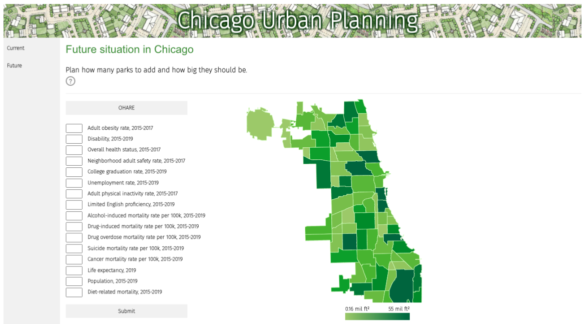


Explore living situations and green zones in Chicago's community areas. Select which metrics you want to visialize. You can also use map to get basic information on Chicago ⛳ and its community areas.

OHARE

☐ Uninsured rate
☑ Major crime
☐ Robbery
☐ Criminal sexual assault
☐ Violent crime
☐ Traffic intensity
☐ Lung cancer diagnosis rate per 100k
☐ Cancer diagnosis rate per 100k
☐ Adult obesity rate
☐ Overall health status
☐ Neighborhood adult safety rate
☐ High school graduation rate
☑ Unemployment rate
☐ Poverty rate
☐ Adult fruit and vegetable servings rate
☐ Adult physical inactivity rate
☐ Adult soda consumption rate
☐ Drug-induced mortality rate per 100k
☐ Drug overdose mortality rate per 100k
☐ Suicide mortality rate per 100k
☐ Cancer mortality rate per 100k
☐ Life expectancy
☐ Population
☐ Drug abuse

Figure 7: Plots of two selected metrics



Plan how many parks to add and how big they should be.

OHARE

☐ Adult obesity rate, 2015-2017
☐ Disability, 2015-2019
☐ Overall health status, 2015-2017
☐ Neighborhood adult safety rate, 2015-2017
☐ College graduation rate, 2015-2019
☐ Unemployment rate, 2015-2019
☐ Adult physical inactivity rate, 2015-2017
☐ Limited English proficiency, 2015-2019
☐ Alcohol-induced mortality rate per 100k, 2015-2019
☐ Drug-induced mortality rate per 100k, 2015-2019
☐ Drug overdose mortality rate per 100k, 2015-2019
☐ Suicide mortality rate per 100k, 2015-2019
☐ Cancer mortality rate per 100k, 2015-2019
☐ Life expectancy, 2019
☐ Population, 2015-2019
☐ Diet-related mortality, 2015-2019

Figure 8: Future Tab with map

5 FUTURE WORK

As the FUTURE TAB is meant to be used mostly from Domain Experts, we would like to use active learning such that ML model learns from user if it is feasible to construct more parks. This can be achieved by adding a button to label predictions as good or bad. In case the user confirms the quality of the prediction, features could be modified to retrain to obtain more accurate model. ML model is a black box from the user's perspective. Therefore, visualization of how XGBOOST trains would be useful. Additionally, the map could be further improved by adding more fine grained information once user clicks on a community area. For better explainability, selection of the feature on which choropleth map is based on would be useful. The major improvement could be the prediction of the area to construct the new park (coordinates and shape). Unfortunately, the necessary data is not available for Chicago. Although, we have an idea that the prediction could be done with Voronoi diagram and convex hulls.

6 CONCLUSIONS

This report describes *Chicago Urban Planning* project, a dashboard that provides support for urban planners during their decision-making process and can also be used by the general public for exploration purposes. The final product has two tabs - CURRENT and FUTURE. The current tab allows users to get deep insight into life quality in different community areas of Chicago by interacting with a map and plots. FUTURE TAB can help domain experts to decide where and how many parks to build in a specific part of the city. The dashboard uses ML methods, such as XGBoost, PCA, and K-Means to predict how many parks one needs to improve overall life quality. Predictions are supported by an explainability metric to help the user to understand the decision process of the model. The model is trained on historical life quality metrics on the community level in the city of Chicago. The dashboard is designed as a two-tier architecture, defined by continuous interaction between the backend and frontend. The dashboard provides a tutorial to ease the onboarding process for new coming users. Overall, the product is designed in a human-centered way, so that users without a required technical background could use it without new information overload.

7 ACKNOWLEDGMENTS

We thank to our teaching assistant Javier Sanguino Bautiste and lecturers Dr. Mennatallah El-Assady and Dr. Furui Cheng for the time invested into our meetings, helpful advice and suggestions.

REFERENCES

[1] City Of Chicago. 2023. Chicago Data Portal. https://data.cityofchicago.org. [Online].
[2] City Of Chicago. 2023. Chicago Data Portal. https://data.cityofchicago.org/Parks-Recreation/Parks-Chicago-Park-District-Park-Boundaries-curren/ej32-qgdr. [Online].
[3] City Of Chicago. 2023. Chicago Data Portal (Crime Data). https://data.cityofchicago.org/Public-Safety/Crimes-2022/9hwr-2zxp/data. [Online].
[4] City Of Chicago. 2023. Chicago Data Portal (Life Quality Indicators). https://chicagohealthatlas.org/indicators. [Online].
[5] City Of Chicago. 2023. Chicago Data Portal (Parks). https://data.cityofchicago.org/Parks-Recreation/Parks-Locations-deprecated-November-2016-/wwy2-k7b3/data. [Online].
[6] S. van der Linden et al. 2023. MediCoSpace: Visual DecisionSupport for Doctor-Patient. Consultations using Medical. Concept Spaces from EHRs. https://doi.org/10.1145/3564275. [Online].