Title: MM802 - Visualization Mini-project

Date: March 15, 2022

Name: Chenggui Sun, Li Bin Song

Project Tile Edmonton Neighborhood Searching Tool

1 Abstract

This mini-project created a searching tool to help Edmontonians to select a good neighborhood to live in. The searching criteria are the average housing price of a neighborhood, the total number of crimes that occurred in a neighborhood, and the price-crime index of a neighborhood defined by this project. This tool can help a user quickly locate a few neighborhoods then start to look for a house listed in those neighborhoods.

2 Introduction

Edmonton is the capital city of the province of Alberta and the major economic, financial, distribution and logistics centers for Alberta. It's also Canada's premier research and education center. There are seven publicly-funded post-secondary institutions located in Edmonton. These institutions are Concordia University of Edmonton, MacEwan University, King's University, NorQuest College, the Northern Alberta Institute of Technology (NAIT), the University of Alberta (U of A) and a campus of Athabasca University [1].

Edmonton has a population of over one million. The residents of Edmonton are known as Edmontonians. More and more people migrate to Edmonton for employment, education, and business from other Canadian provinces and other countries and become Edmontonians every year, due to Edmonton's unique governmental and cultural status, its dynamic economy, and premier research and education resources. For new Edmontonians, they require information to select a neighborhood to live in. And for old Edmontonians, some of them may need to find a new neighborhood to buy a new house to improve his/her familly's living conditions. The objective of this project is to provide a tool to help Edmontonians to select one of the Edmonton neighborhoods to live in based on the residential neighborhood property value and the total number of crimes that occurred from 2010 to 2019.

The original datasets used for this project are the 2022 Edmonton Property Assessment Data and the Edmonton Neighborhood Criminal Occurrence Data, which were cleaned and analyzed for the visualization app. The details about these two datasets are described as follows:

- 2022 Edmonton Property Assessment Data was obtained from the City of Edmonton's Open Data Portal (https://data.edmonton.ca). This dataset provides the address, type, account number, assessed value, latitude, and longitude of each property and the neighborhood that each property belongs to. The average house price of a neighborhood was calculated based on the assessed values of the properties in this neighborhood. Each neighborhood can be located with its latitude and longitude.

– Edmonton Neighborhood Criminal Occurrence was obtained from Edmonton's Open Data Portal (https://dashboard.edmonton.ca). This dataset ranges from 2010 to 2019. The criminal occurrence in each neighborhood is collected quarterly every year. This dataset was cleaned and the total number of crimes in each neighborhood was calculated for our visualization app.

3 Project Status

This "Edmonton Neighborhood Searching Tool" has been completed on time and the tool can be assessed from url:`https://share.streamlit.io/ziyunxiao/mm802_miniproject/app.py`. It provides four functions to help users to locate a neighborhood. These four functions are:

– Search neighborhoods based on users' budget for buying a single house aka the average housing prices of neighborhoods;

– Refine budget/house price search results based on the total numbers of crime occurrence in neighborhoods;

– Refine budget/house price search results based on the price-crime index, which is defined as the average house price of a neighborhood over its total number of crimes occurred from 2010 to 2019;

– Show the locations of the search results on a map.

The key tasks of this project and the contributions of team members are shown in table 1:

Project Tasks	**Contributor**
Project Definition	Chenggui Sun, Li Bin Song
Development Environement Setup	Li Bin Song
Python Codes for Data Processing	Chenggui Sun, Li Bin Song
Visualization App	Li Bin Song, Chenggui Sun
Project Report, PPT slides	Chenggui Sun, Li Bin Song

Table 1: Project tasks and contributors

4 Development Environment

4.1 Choosing Development Framework

The use scenario of our project is for Edmontonians to find a neighborhood to buy a house and live in. The following criteria/issues are considered to design this visualization app:

– The search tool should be easy to access and user-friendly. It's not considered to build a Mobile App for this project, because users need to install a Mobile App to use it.

– This tool should present a good user interface on both computers and mobile devices. Therefore, a web application is the best choice to build this neighborhood search tool. This web application needs to incorporate bootstrap libraries to provide a responsive user interface.

– Java, C#, and PHP are good languages to build a web framework. However, because this web application is relatively simple and python is a popular and powerful language as well, Python is selected to build this web application.

– JS libraries should be utilized to build good user interfaces and deliver good data visualization to users.

Based on the above criteria, a solution that is on top of the Streamlit library is considered to build this web visualization application. Streamlit is an open-source Python library. It has several advantages:

– It includes JS libraries to build beautiful web apps.

– It includes bootstrap, which creates responsive web apps.

– The back-end is using Python, which is popular and powerful.

– The base service layer of Streamlit is a Python Flask framework. It can be customized for more powerful functions.

– It has multiple widgets to handle user inputs and data visualization.

– It has a free cloud service to deploy free apps.

– It is a quick web development tool for data scientists.

4.2 Development Environment Setup

This project is open-source and hosted on GitHub. The source code repo is located at URL: `https://github.com/ziyunxiao/mm802_miniproject.git`

Here are the steps to setup the project. The commands are based on Linux.

– Clone the project: *git clone https://github.com/ziyunxiao/mm802_miniproject.git*

– Get in project folder: *cd mm802_miniproject*

– Setup python virtual environment: *python -m venv .venv*

– Activate virtual envirionment: *. .venv/bin/activate*

– Install required packages: *pip install –upgrade pip && pip install -r requirements.txt*

– Start web server: *streamlit run app.py –server.port 8080*

– Open browser and access: `http://localhost:8080/`

5 Development Work

5.1 Data Preparation

In this project, Edmonton Property Assessment Data and Edmonton Neighborhood Criminal Occurrence are used to prepare the visualization data. The details about these two datasets have been discussed in the introduction section. These two datasets were downloaded as csv files, processed and analyzed using Pandas and Numpy with Jupyter Notebook - *data_preparation.ipynb*. In this notebook, the main tasks are:

- Clean Data: the average pricing of single houses in each neighborhood were calculated and unuseful data were removed;
- Rename columns: columns were named with common names. e.g. keeping all column names as lower case, renaming latitude to lat, longitude to lon.
- Concatenate datasets: house price data and crime data were joined together so the new dataset can provide both average house price and the total number of crimes for each neighborhood.
- The final dataset is saved to pkl (pickle) file for less storage.

More details about the data processing codes are available in the data_preparation Juypter notebook of the Git repository of this project.

5.2 Visualization App Development

This is the main component of this project. Streamlit takes the logic from the beginning to the end like a Python script. First, a file named app.py is created.

The main logic is explained below:

1. Load data using scripts shown in figure 1:

```
# LOADING DATA
@st.experimental_memo
def load_data(fname):
    data = pkl.load(open(fname,'rb'))
    return data

data = load_data('./prop_crime.pkl')
```

Figure 1: Data loading

2. Create user input and visualize the results using scripts shown in figure 2:

The logic is very easy to understand. First, an input widget was created. This widget allows users to change the filter based on their interested values for each step/question. Next, the values

```
# Q1
st.markdown("## Step 1 (Q1): Select Neighborhoods Based on Budget")
price_range = st.slider( 'List neighborhoods in a range of house prices (x 1000$).',\
    | 200, 2000, (300,600), step=10)
df1 = data.loc[(data['value']>=price_range[0] * 1000) & (data['value']<= price_range[1] * 1000),\
    ["neighbourhood","value"]]
st.write(price_range)
st.dataframe(df1)
```

Figure 2: User input creation

will be passed to the back, and Pandas dataframe will filter the data, then st.dataframe will display the search results.

3. Similar logic is applied to create visualization results for Step 2/Q2, Step 3/Q3, and Step 4/Map-View.

5.3 Run the application

During testing, the web app can be started by command *streamlit run app.py –server.port 8080*. This will bring up the webserver and the web visualization application can be accessed from `http://localhost:8080/`.

5.4 Deploy to cloud

The web app of this project can be deployed to the cloud service - streamlit.io following the steps shown below:

– 1. Create an account on streamlit.io. This can be done by authentication through a GitHub account.

– 2. Click new app from the menu shown in figure 3

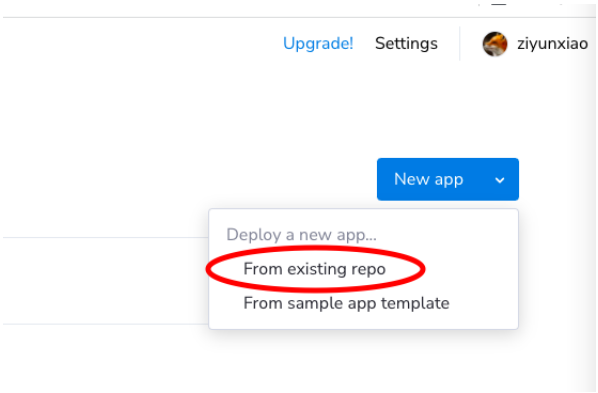


Figure 3: New app selection demonstration

– 3. Click from the existing repo. Then it shows a page to select an application repo as shown in figure 4.

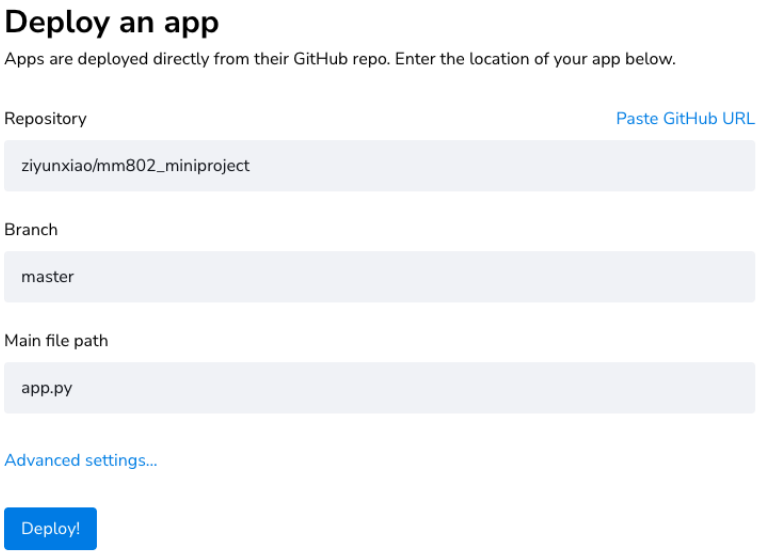


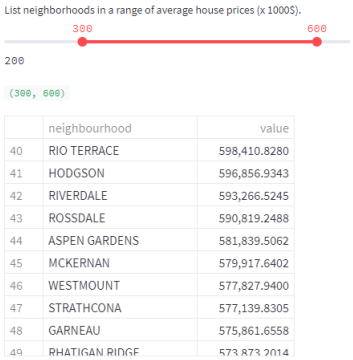
Figure 4: App deployment demonstration

– Then the app is deployed and will be updated when any changes on the repo are made.

5.5 How to use the application

There are two ways for users to use this visualization app. One way is to refine their search criteria step by step to get the search results like the following steps show:

– 1. Use the slider to define the range of the average house price to search neighborhoods. As figure 5 shows, a range of house prices from 300,000 to 600,000 Canadian Dollars was set. Then a list of neighborhoods was generated. More results can be seen by putting the cursor on the table and scrolling down.



	neighbourhood	value
40	RIO TERRACE	598,410.8280
41	HODGSON	596,856.9343
42	RIVERDALE	593,266.5245
43	ROSSDALE	590,819.2488
44	ASPEN GARDENS	581,839.5062
45	MCKERNAN	579,917.6402
46	WESTMOUNT	577,827.9400
47	STRATHCONA	577,139.8305
48	GARNEAU	575,861.6558
49	RHATIGAN RIDGE	573,873.2014

Figure 5: Search slider demonstration

– 2. Use the slider to define the range of the number of crimes to search neighborhoods. Then the search results can be obtained as step 1.

– 3. Use the slider to define the range of Price-Crime Index to search neighborhoods as step 1.

– 4. The locations of the search results will be shown on the map as shown in figure 6.

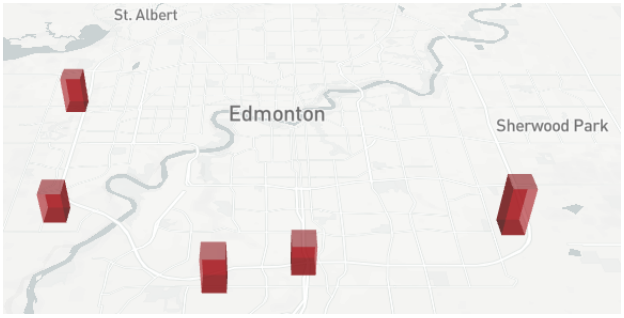


Figure 6: Map view of search results

The other way to use this app is to get search results using each search criterion individually. However, the users need to set the other search sliders to their minimum and maximum values.

6 Concluding Remarks

In this project, a web visualization application was built to help Edmontonians to search Edmonton neighborhoods based on the average housing price, the total number of crimes that occurred from 2010 to 2019, and the price-crime index of a neighborhood. A map can illustrate the locations of the search results of neighborhoods. Pandas, Numpy, and Streamlit were used to create this web visualization app. To build this web application, we are not only learned and explored HTML, JavaScript, and JQuery, but also take the initiative to learn Streamlit, which is more suitable for this project.

Due to time limit, the functions that were built for this searching tool is limited as well. This web visualization tool can be extended from three aspects:

– Functions: more search criteria can be added, such as schools, business venues;

– User-base: the user-base can be extended to other Canadian cities, such as Toronto, Vancouver, Montreal, and Calgary;

– User-interface: the user interface shall be optimized as well, when more functions are added and more users are expected.

References

[1] Edmonton, March 2022.