Project 3 - Team 2

Commercial
Building Disclosure Visualisations

Our Team Members & GitHub Repository

Team Members: Ben, Erin, Violet and Archana

GitHub Repository: https://github.com/VeeBui/project-three-team-two

What is our visualisation about?

We are working for a building leasing and marketing company, and have been tasked with researching the Commercial Building Disclosure project, which can benefit from data visualisation.

Searchable with data, you can view a map as well as specific building metrics such as ratings, energy intensity, net lettable areas and more.

The data delivery is manual to up-date and requires processing to make easy to understand, this clearly shows the certified buildings and their ratings in an easy and visual way.

The primary objective of this project is to perform a visualisation analysis of up to 40,000 ratings within Victoria using available data from the Australian Government site www.cbd.gov.au.

Australian Government

About the CBD Program & Ethics

A National Energy Efficiency Program

The Commercial Building Disclosure (CBD) Program is a national regulatory program that requires energy efficiency information to be provided in most cases when commercial office space of 1000 square metres or more is offered for sale or lease.

Ethical Considerations

The CBD Program is an example of transparency in data collection. This is an instance where you would want the data that has been collected to be shared. The data provided to clients allows them to make an informed choice regarding the energy efficiency of the building they would like to lease which could ultimately affect their fixed costs.

Project Process

1 Obtain Data from CBD site

We saved the CSV file from the CBD website directly, this included all states and was >200mb in size.

Data Collection

2 Clean and Re-Structure Data

Using the Python tools we re-arranged the data to suit our analysis needs, limiting to VIC

Data Cleaning

3 Built Jupyter Notebook

Using Python we were able to make static charts and map.

Visualisations #1

4 Built JS + HTML

Based on the analysis we were able to make an interactive map.

Visualisations #2

Dependency Review

Our project included the following stack and dependencies;

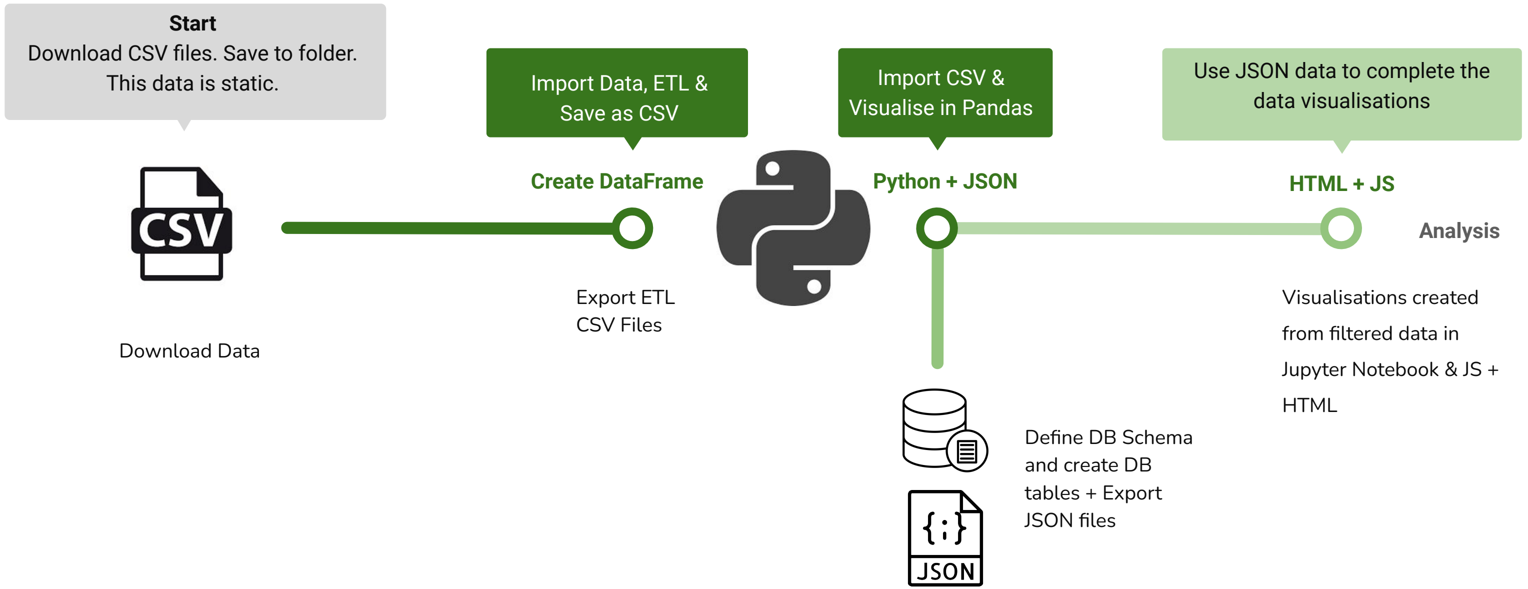
- Publicly available CSV data from Ethical Government published site
- Jupyter (Pandas) for data loading and transformation
- Jupyter for static data visualisation
- Create JSON Files from Pandas
- Create SQLIte DB to store the data

- JS to make the interactive map.

- **The new python library Plotly Express was used in Visualisations #1 Jupyter Notebook.**

Data Collection

Method For Data Collection



Method | Dependencies

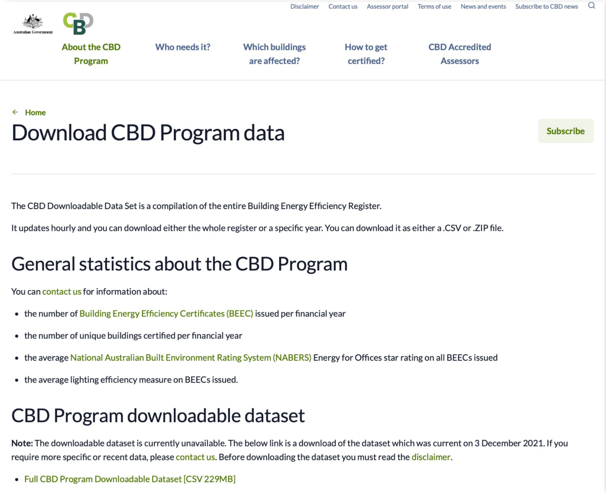
Our project included the following setup and dependencies;

```
from pathlib import Path
from sqlalchemy.ext.automap import automap_base
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy import Column, String, Float, Integer
from sqlalchemy import create_engine
from sqlalchemy.orm import Session
```

CBD | Website Link

Our project included the Commercial Building Disclosure site;

Given the file size of the download >200mb, we have not uploaded this to out GitHub Repository. Please see link to file for reference.


Disclaimer Contact us Assessor portal Terms of use News and events Subscribe to CBD news

About the CBD Program | Who needs it? | Which buildings are affected? | How to get certified? | CBD Accredited Assessors

← Home

Download CBD Program data

Subscribe

The CBD Downloadable Data Set is a compilation of the entire Building Energy Efficiency Register.

It updates hourly and you can download either the whole register or a specific year. You can download it as either a .CSV or .ZIP file.

General statistics about the CBD Program

You can contact us for information about:

- the number of Building Energy Efficiency Certificates (BEEC) issued per financial year
- the number of unique buildings certified per financial year
- the average National Australian Built Environment Rating System (NABERS) Energy for Offices star rating on all BEECs issued
- the average lighting efficiency measure on BEECs issued.

CBD Program downloadable dataset

Note: The downloadable dataset is currently unavailable. The below link is a download of the dataset which was current on 3 December 2021. If you require more specific or recent data, please contact us. Before downloading the dataset you must read the disclaimer.

- Full CBD Program Downloadable Dataset [CSV 229MB]

Full CBD Program Downloadable Dataset [CSV 229MB]

The Jupiter Lab file showing the first CSV read;

```
[1]: import pandas as pd
     import numpy as np
```

```
[2]: df = pd.read_csv(('/Users/benjaminmason/Desktop/Study/Project 03_Team 2/Resourcse/Dataset.csv'), low_memory=False)
     df.head()
```

[2]:

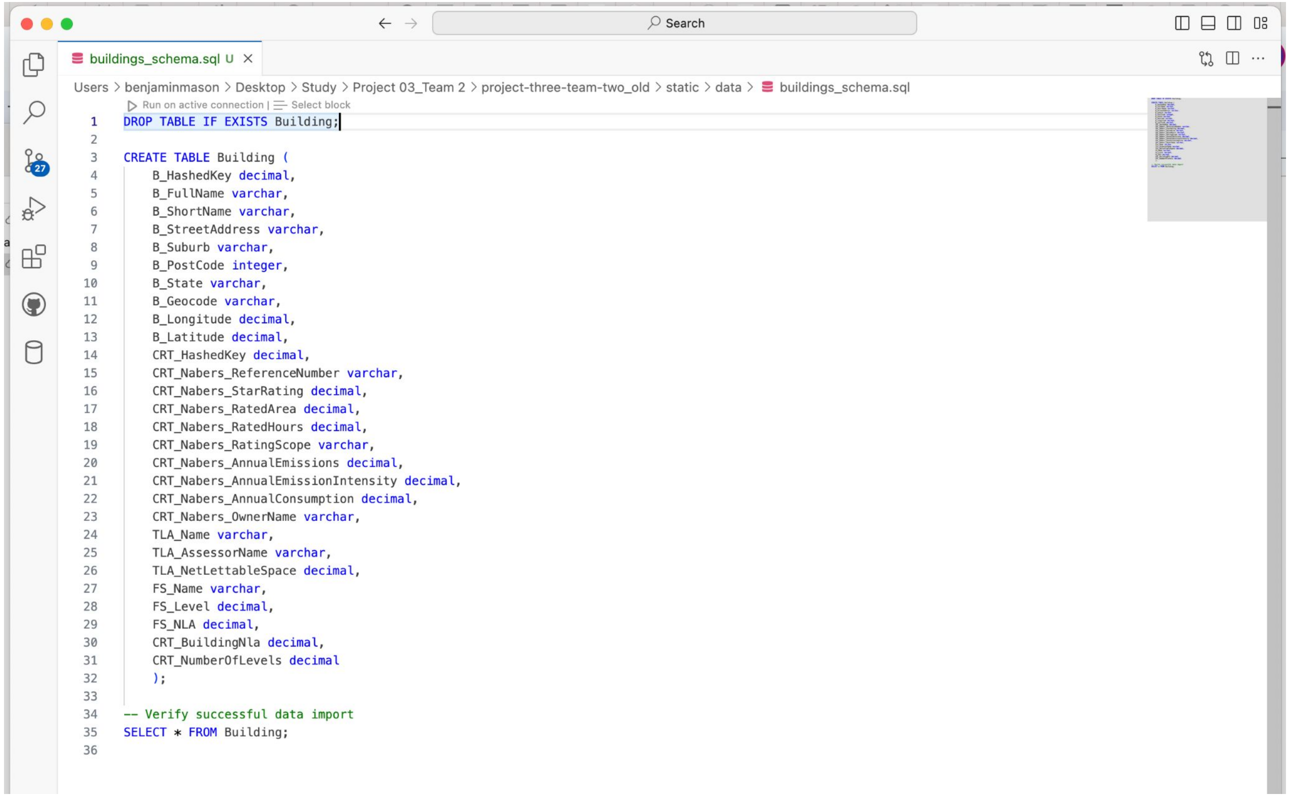
	B_HashedKey	B_FullName	B_ShortName	B_StreetAddress	B_Suburb	B_PostCode	B_State	B_Geocode	B_Longitude	B_Latitude	...	FS_Pro
0	3292694929379159574	1 Thynne Street, Bruce, ACT, 2617	NaN	1 Thynne Street	Bruce	2617	ACT	GAACT714853532	149.093585	-35.240431	...	
1	3292694929379159574	1 Thynne Street, Bruce, ACT, 2617	NaN	1 Thynne Street	Bruce	2617	ACT	GAACT714853532	149.093585	-35.240431	...	
2	3292694929379159574	1 Thynne Street, Bruce, ACT, 2617	NaN	1 Thynne Street	Bruce	2617	ACT	GAACT714853532	149.093585	-35.240431	...	
3	3292694929379159574	1 Thynne Street, Bruce, ACT, 2617	NaN	1 Thynne Street	Bruce	2617	ACT	GAACT714853532	149.093585	-35.240431	...	
4	3292694929379159574	1 Thynne Street, Bruce, ACT, 2617	NaN	1 Thynne Street	Bruce	2617	ACT	GAACT714853532	149.093585	-35.240431	...	

5 rows × 62 columns

```
[3]: df.info()
```

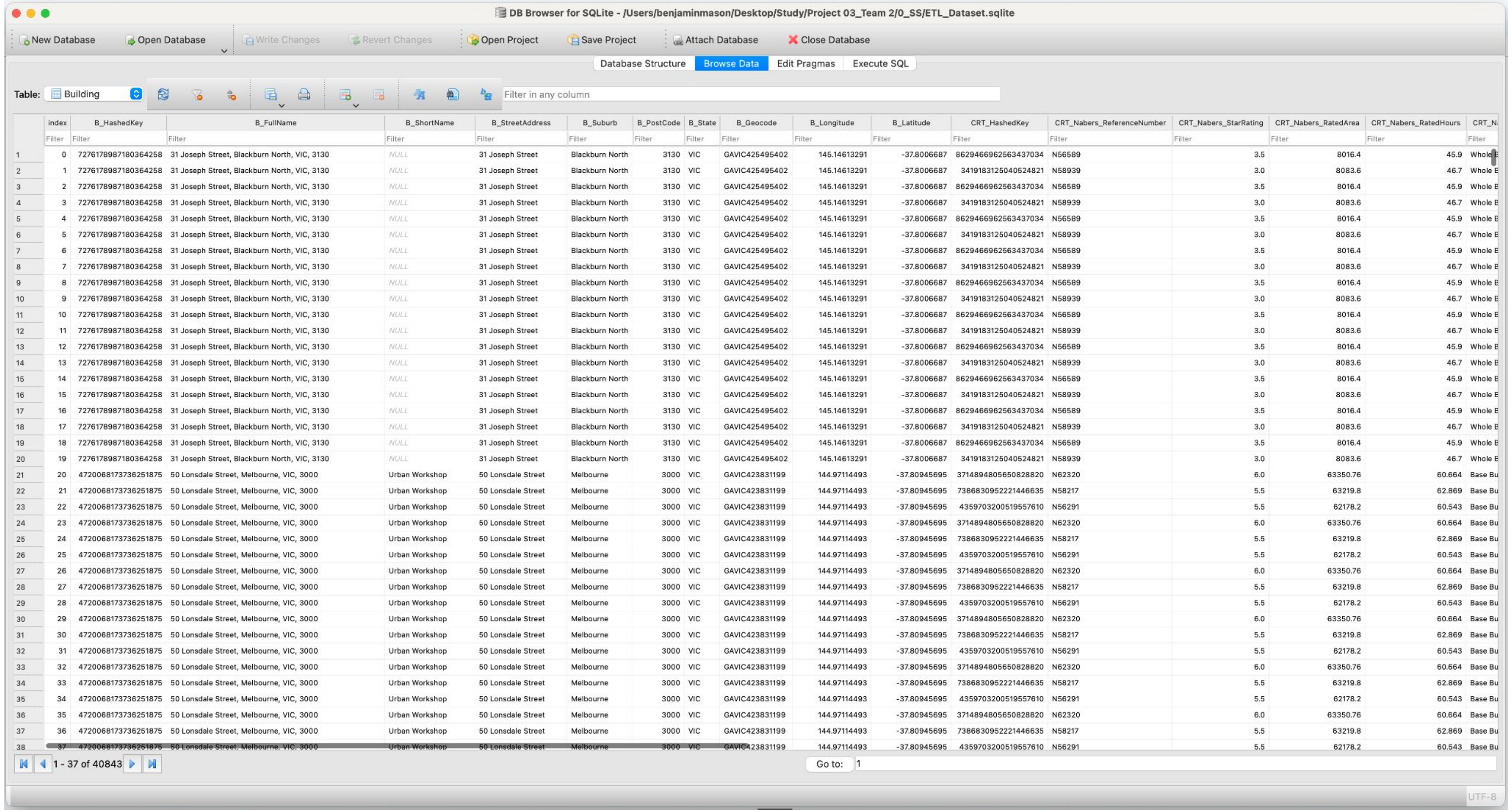
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 204749 entries, 0 to 204748
Data columns (total 62 columns):
 #   Column                           Non-Null Count   Dtype
---  ------                           --------------   -----
 0   B_HashedKey                      204749 non-null  int64
 1   B_FullName                       204749 non-null  object
 2   B_ShortName                      142420 non-null  object
 3   B_StreetAddress                  204749 non-null  object
 4   B_Suburb                         204749 non-null  object
 5   B_PostCode                       204749 non-null  int64
```

The SQLite Building Table Schema;




```sql
DROP TABLE IF EXISTS Building;

CREATE TABLE Building (
    B_HashedKey decimal,
    B_FullName varchar,
    B_ShortName varchar,
    B_StreetAddress varchar,
    B_Suburb varchar,
    B_PostCode integer,
    B_State varchar,
    B_Geocode varchar,
    B_Longitude decimal,
    B_Latitude decimal,
    CRT_HashedKey decimal,
    CRT_Nabers_ReferenceNumber varchar,
    CRT_Nabers_StarRating decimal,
    CRT_Nabers_RatedArea decimal,
    CRT_Nabers_RatedHours decimal,
    CRT_Nabers_RatingScope varchar,
    CRT_Nabers_AnnualEmissions decimal,
    CRT_Nabers_AnnualEmissionIntensity decimal,
    CRT_Nabers_AnnualConsumption decimal,
    CRT_Nabers_OwnerName varchar,
    TLA_Name varchar,
    TLA_AssessorName varchar,
    TLA_NetLettableSpace decimal,
    FS_Name varchar,
    FS_Level decimal,
    FS_NLA decimal,
    CRT_BuildingNla decimal,
    CRT_NumberOfLevels decimal
    );

-- Verify successful data import
SELECT * FROM Building;
```

The SQLite DataBase (Showing Buildings Table);



DB Browser for SQLite - /Users/benjaminmason/Desktop/Study/Project 03_Team 2/0_SS/ETL_Dataset.sqlite

New Database | Open Database | Write Changes | Revert Changes | Open Project | Save Project | Attach Database | Close Database

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: Building

Filter in any column

	index	B_HashedKey	B_FullName	B_ShortName	B_StreetAddress	B_Suburb	B_PostCode	B_State	B_Geocode	B_Longitude	B_Latitude	CRT_HashedKey	CRT_Nabers_ReferenceNumber	CRT_Nabers_StarRating	CRT_Nabers_RatedArea	CRT_Nabers_RatedHours	CRT_N
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	0	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	8629466962563437034	N56589	3.5	8016.4	45.9	Whole E
2	1	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	3419183125040524821	N58939	3.0	8083.6	46.7	Whole E
3	2	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	8629466962563437034	N56589	3.5	8016.4	45.9	Whole E
4	3	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	3419183125040524821	N58939	3.0	8083.6	46.7	Whole E
5	4	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	8629466962563437034	N56589	3.5	8016.4	45.9	Whole E
6	5	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	3419183125040524821	N58939	3.0	8083.6	46.7	Whole E
7	6	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	8629466962563437034	N56589	3.5	8016.4	45.9	Whole E
8	7	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	3419183125040524821	N58939	3.0	8083.6	46.7	Whole E
9	8	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	8629466962563437034	N56589	3.5	8016.4	45.9	Whole E
10	9	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	3419183125040524821	N58939	3.0	8083.6	46.7	Whole E
11	10	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	8629466962563437034	N56589	3.5	8016.4	45.9	Whole E
12	11	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	3419183125040524821	N58939	3.0	8083.6	46.7	Whole E
13	12	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	8629466962563437034	N56589	3.5	8016.4	45.9	Whole E
14	13	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	3419183125040524821	N58939	3.0	8083.6	46.7	Whole E
15	14	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	8629466962563437034	N56589	3.5	8016.4	45.9	Whole E
16	15	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	3419183125040524821	N58939	3.0	8083.6	46.7	Whole E
17	16	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	8629466962563437034	N56589	3.5	8016.4	45.9	Whole E
18	17	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	3419183125040524821	N58939	3.0	8083.6	46.7	Whole E
19	18	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	8629466962563437034	N56589	3.5	8016.4	45.9	Whole E
20	19	7276178987180364258	31 Joseph Street, Blackburn North, VIC, 3130	NULL	31 Joseph Street	Blackburn North	3130	VIC	GAVIC425495402	145.14613291	-37.8006687	3419183125040524821	N58939	3.0	8083.6	46.7	Whole E
21	20	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	3714894805650828820	N62320	6.0	63350.76	60.664	Base Bu
22	21	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	7386830952221446635	N58217	5.5	63219.8	62.869	Base Bu
23	22	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	4359703200519557610	N56291	5.5	62178.2	60.543	Base Bu
24	23	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	3714894805650828820	N62320	6.0	63350.76	60.664	Base Bu
25	24	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	7386830952221446635	N58217	5.5	63219.8	62.869	Base Bu
26	25	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	4359703200519557610	N56291	5.5	62178.2	60.543	Base Bu
27	26	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	3714894805650828820	N62320	6.0	63350.76	60.664	Base Bu
28	27	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	7386830952221446635	N58217	5.5	63219.8	62.869	Base Bu
29	28	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	4359703200519557610	N56291	5.5	62178.2	60.543	Base Bu
30	29	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	3714894805650828820	N62320	6.0	63350.76	60.664	Base Bu
31	30	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	7386830952221446635	N58217	5.5	63219.8	62.869	Base Bu
32	31	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	4359703200519557610	N56291	5.5	62178.2	60.543	Base Bu
33	32	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	3714894805650828820	N62320	6.0	63350.76	60.664	Base Bu
34	33	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	7386830952221446635	N58217	5.5	63219.8	62.869	Base Bu
35	34	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	4359703200519557610	N56291	5.5	62178.2	60.543	Base Bu
36	35	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	3714894805650828820	N62320	6.0	63350.76	60.664	Base Bu
37	36	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	7386830952221446635	N58217	5.5	63219.8	62.869	Base Bu
38	37	4720068173736251875	50 Lonsdale Street, Melbourne, VIC, 3000	Urban Workshop	50 Lonsdale Street	Melbourne	3000	VIC	GAVIC423831199	144.97114493	-37.80945695	4359703200519557610	N56291	5.5	62178.2	60.543	Base Bu

1 - 37 of 40843

Go to: 1

UTF-8

Data Cleaning

Method | Data Cleaning

For our project we needed to clean the data using Python;

- Drop columns that had “dirty data” meaning various data object type in the same column. This was mainly date and columns with additional separators that couldn't be split without making errors.
- Drop rows that didn't include Lat + Long for adding to the Map.
- Reduced dataset to limit the visualisations to buildings in Victoria.

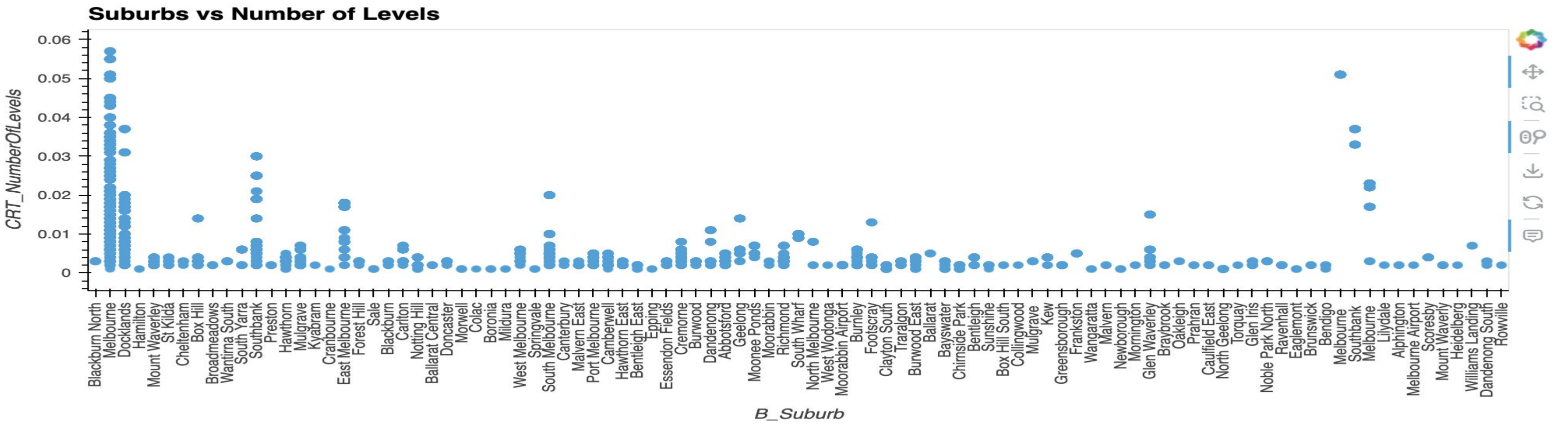
- Reduced total columns from 62 to 28.
- Reduced total rows from 204,000 to 40,800.

The Jupiter Lab file showing the leaned up dataset;

```
[46]: # Import the dependencies.
import sqlite3
from sqlalchemy import create_engine
from sqlalchemy import Column, Integer, String, Float
from sqlalchemy.ext.declarative import declarative_base
from pathlib import Path
from sqlalchemy import create_engine, text
import matplotlib.pyplot as plt

[23]: Base = declarative_base()
conn = sqlite3.connect('Buildings.sqlite')
cursor = conn.cursor()

[24]: cursor.execute('''DROP TABLE Building''')

[24]: <sqlite3.Cursor at 0x306af9240>

[25]: # Create a Buidling Class for the table.
cursor.execute('''CREATE TABLE Building(
B_HashedKey decimal, B_FullName varchar, B_ShortName varchar, B_StreetAddress varchar, B_Suburb varchar, B_PostCode integer,
B_State varchar, B_Geocode varchar, B_Longitude decimal, B_Latitude decimal,
CRT_HashedKey decimal, CRT_Nabers_ReferenceNumber varchar,
CRT_Nabers_StarRating decimal, CRT_Nabers_RatedArea decimal,
CRT_Nabers_RatedHours decimal, CRT_Nabers_RatingScope varchar,
CRT_Nabers_AnnualEmissions decimal, CRT_Nabers_AnnualEmissionIntensity decimal,
CRT_Nabers_AnnualConsumption decimal, CRT_Nabers_OwnerName varchar,
TLA_Name varchar, TLA_AssessorName varchar, TLA_NetLettableSpace decimal,
FS_Name varchar, FS_Level decimal, FS_NLA decimal, CRT_BuildingNla decimal, CRT_NumberOfLevels decimal)''')

[25]: <sqlite3.Cursor at 0x306af9240>

[26]: Building_ETL = pd.read_csv('/Users/benjaminmason/Desktop/Study/Project 03_Team 2/project-three-team-two_old/resources/ETL_Dataset.csv')

[27]: Building_ETL.to_sql('Building', conn, if_exists='append', index = False)

[27]: 40843

[28]: cursor.execute('''SELECT * FROM Building''').fetchall()

[28]: [(7276178987180364258,
  '31 Joseph Street, Blackburn North, VIC, 3130',
  None,
  '31 Joseph Street',
```

Visualisations #1
Jupyter Notebook

Number of levels of buildings per suburbs

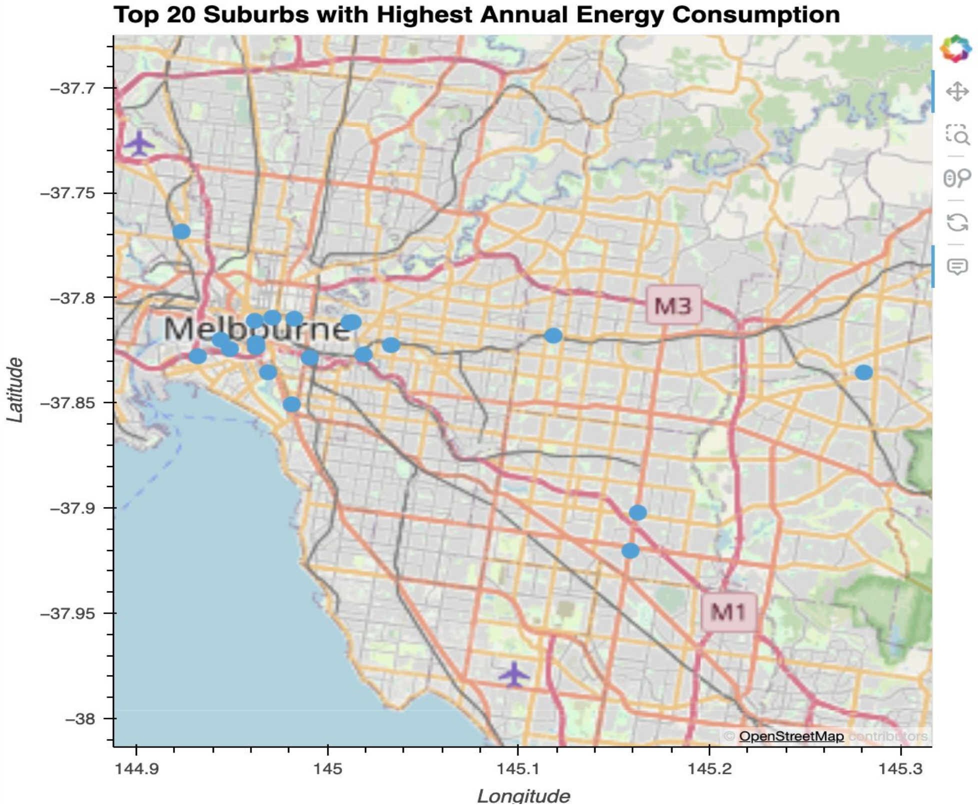


Summary:

- The scatter plot reveals the relationship between suburbs and greenhouse gas emissions.
- Most data points cluster at the lower end of GHG emissions, with a few outliers showing higher emissions.
- Without context, we can’t definitively determine the correlation or causation between suburbs and emissions.

TOP 20 Suburbs with Highest Annual Energy Consumption

Here's a map plot displaying the top 20 suburbs with the highest annual energy consumption. The plot uses latitude and longitude coordinates to visualise these suburbs on the map.

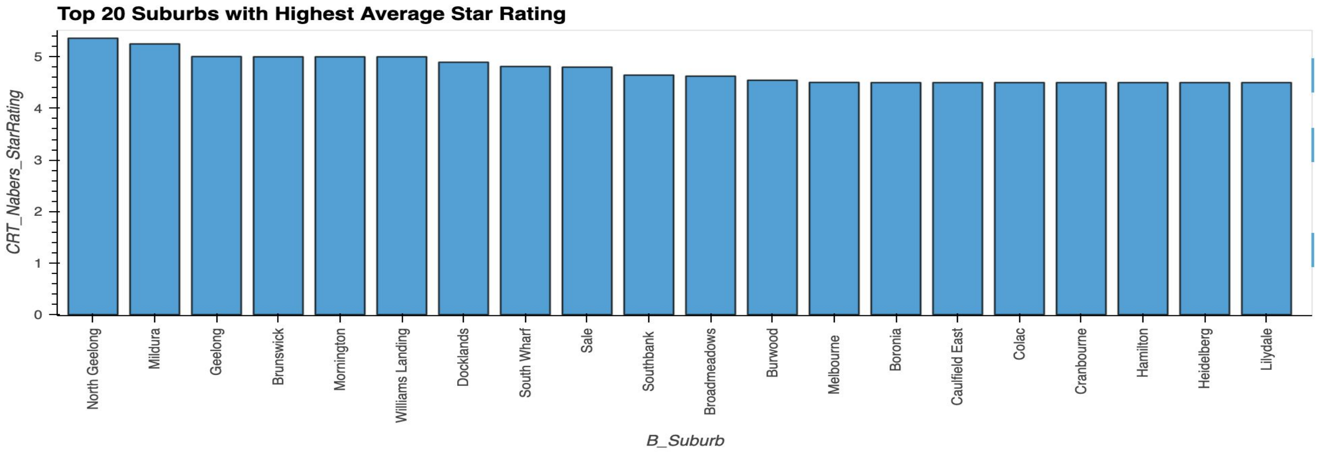


	B_Suburb
0	North Geelong
1	Mildura
2	Geelong
3	Brunswick
4	Mornington
5	Williams Landing
6	Docklands
7	South Wharf
8	Sale
9	Southbank
10	Broadmeadows
11	Burwood
12	Melbourne
13	Boronia
14	Caulfield East
15	Colac
16	Cranbourne
17	Hamilton
18	Heidelberg
19	Lilydale

"Average Star Ratings for Top 20 Suburbs"

```python
# Calculate the average 'CRT_Nabers_StarRating' for each suburb
average_ratings = building_df.groupby('B_Suburb')['CRT_Nabers_StarRating'].mean()

# Select the top 20 suburbs with the highest average rating
top_20_suburbs = average_ratings.nlargest(20)

# Convert the Series to a DataFrame
top_20_suburbs_df = top_20_suburbs.reset_index()

# Display the DataFrame
top_20_suburbs_df
```
✓ 0.0s

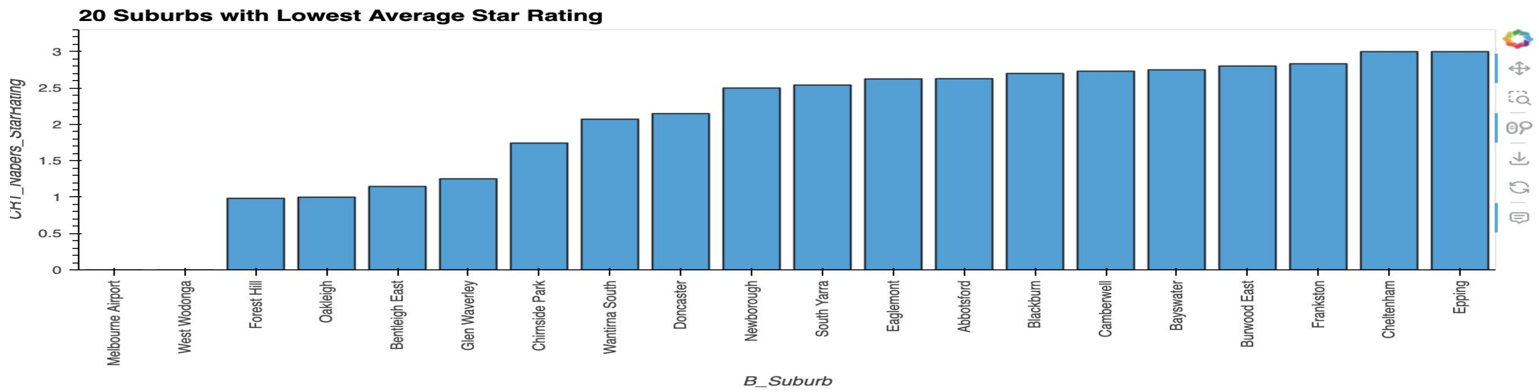
	B_Suburb	CRT_Nabers_StarRating
0	North Geelong	5.360294
1	Mildura	5.250000
2	Geelong	5.004808
3	Brunswick	5.000000
4	Mornington	5.000000
5	Williams Landing	5.000000
6	Docklands	4.894899
7	South Wharf	4.812500
8	Sale	4.800000
9	Southbank	4.645000
10	Broadmeadows	4.625000
11	Burwood	4.545455
12	Melbourne	4.503440
13	Boronia	4.500000
14	Caulfield East	4.500000
15	Colac	4.500000
16	Cranbourne	4.500000
17	Hamilton	4.500000
18	Heidelberg	4.500000
19	Lilydale	4.500000



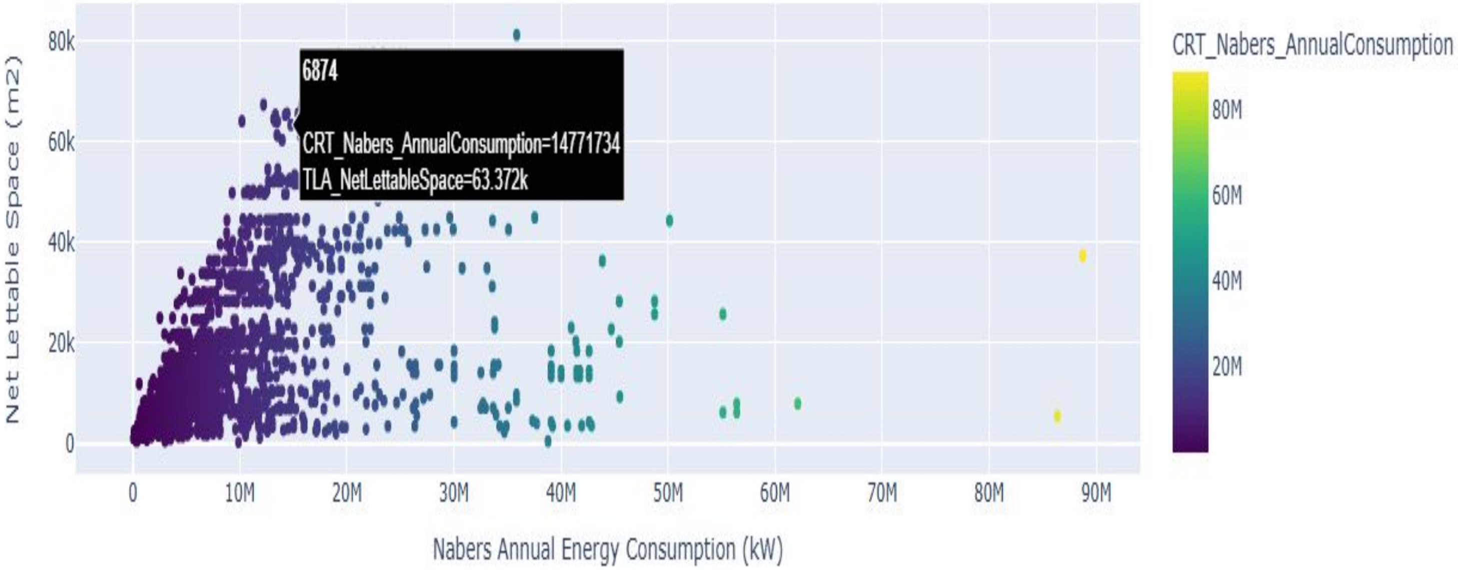
“Average Star Ratings for lowest 20 Suburbs”

```python
# 20 suburbs with the lowest average rating
bottom_20_suburbs = average_ratings.nsmallest(20)
bottom_20_suburbs_df = bottom_20_suburbs.reset_index()

# Create a bar chart of the 20 suburbs with the lowest average rating
bottom_20_suburbs_df.hvplot.bar(x='B_Suburb', y='CRT_Nabers_StarRating', title='20 Suburbs with Lowest Average Star Rating', rot=90, height=400,
```

✓ 0.2s Python



Annual Consumption vs. Net Lettable Space



Visualisations #2
Flask + HTML + JS + JSON

Method | Further Data Cleaning

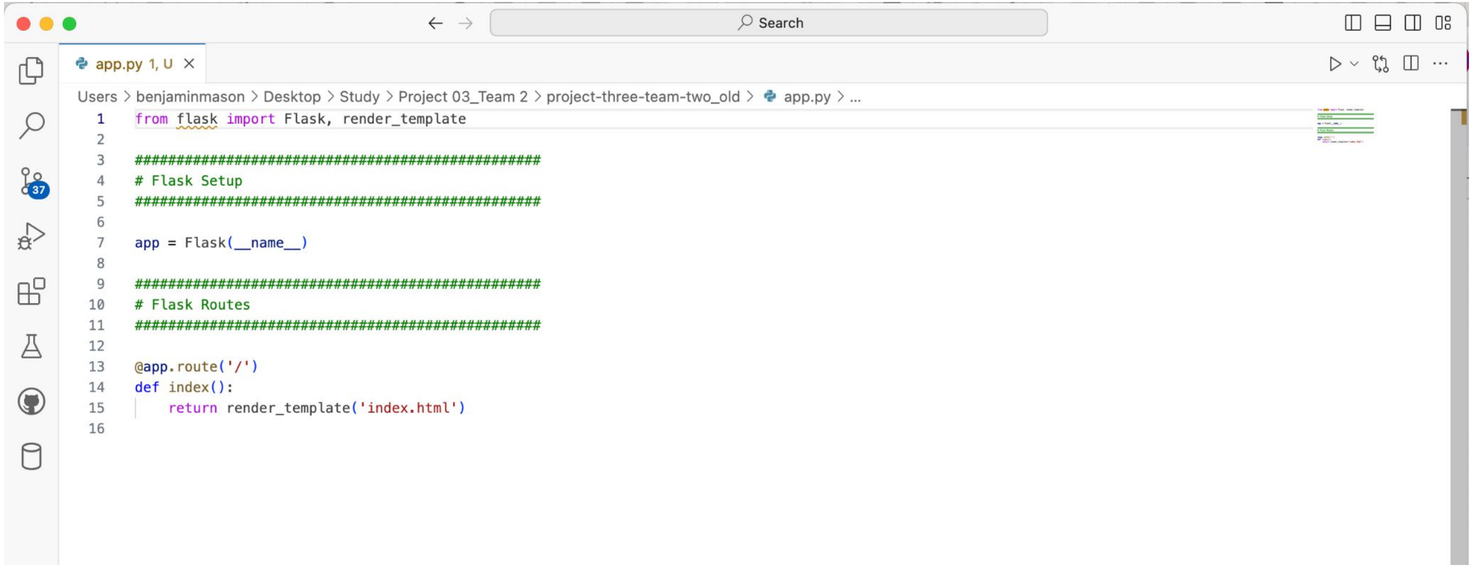
For this part of the project we needed to reduce the quantity of rows in the data set to make the map load using JavaScript & JSON;

- Drop duplicate building entries, this was done to allow the Map to load.

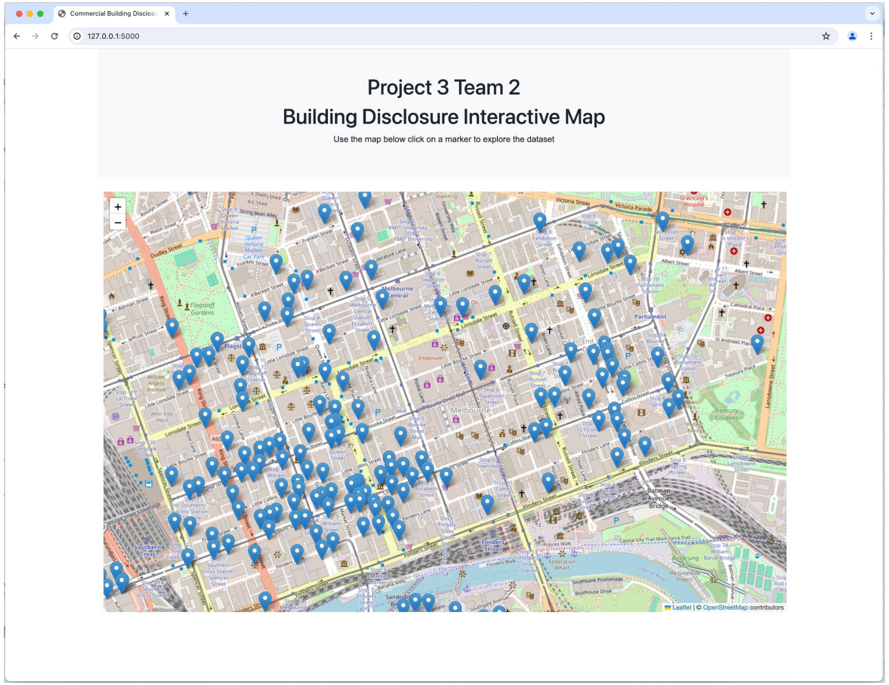
- Reduced total rows from 40,800 to 2,556

Flask | JS + JSON + CSS

This section of the project includes Flask app.py using HTML, JavaScript and JSON



Interactive Map to Explore Data



Lessons learnt

Lessons learnt

1. Using GitHub and branch for code revision management is of benefit, but has file limit <100mb.
2. Acquiring data from public sites can be “dirty” and require extensive clean up and testing for errors.
3. Street Map will not load markers when using excessive entries.
4. Generating JSON data within FLASK using the SQLite DB was difficult and as an alternative used a static JSON data file exported from Python.
5. Domain knowledge (or lack of time) can play a major role in DB backend & FLASK development, more time was needed to finish the FLASK (SQLite > JSON) back end creation.

Thank You !

Team 2