National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Shubham Ramrao Jadhav
Student ID: X22112600

School of Computing
National College of Ireland

Supervisor: Prof. Mayank Jain

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shubham Ramrao Jadhav
Student ID:	22112600
Programme:	MSc in Data Analytics
Year:	2023-2024
Module:	MSc Research Project
Supervisor:	Prof. Mayank Jain
Submission Due Date:	14 December 2023
Project Title:	Configuration Manual
Word Count:	1395
Page Count:	8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Shubham Ramrao Jadhav
Date:	27th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☑
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☑
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☑

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Shubham Ramrao Jadhav
2112600

1 Introduction

The configuration manual contains an explanation of the environment that was configured for the research project that was being conducted. This comprises information about the programming language that was utilized, the configurations of the system, and the libraries and packages that were utilized. It is anticipated that the Configuration Manual will be composed of the following six sections:

- Environment Configuration
- Technical Specifications
- Data Sources
- Steps to Implement and Execute the Project

2 Environment Configuration

This section of the configuration documentation describes the hardware and software system configuration that was used for the project.

2.1 Hardware Specification

Table 1 below lists the necessary hardware along with its specifications.

Hardware Component	**Specification**
Processor	12th Gen Intel(R) Core(TM) i5-1240P
RAM	16.0 GB
Storage	256GB SSD
Operating System	Windows 11
Architecture	64-bit
Processor Architecture	x64-based
GPU	Intel(R) Iris(R) Xe Graphics

Table 1: Hardware Specifications

Category	Software/Tool
Programming Language	Python 3.8.16
IDE	Jupyter Notebook
Web Browser	Google Chrome
Other Software	Microsoft Office and Overleaf
Storage	Github

Table 2: Software and Tools

2.2 Software Specification

As shown in Table 2, this table lists various software and tools used for different purposes.

3 Technical Specifications

3.1 Interactive Coding and Documentation

It is necessary to have the following software (Integrated Development Environments):

1. Anaconda Distribution Package: It is required to run python notebook by using "Jupyter" 1. `https://www.anaconda.com/products/distribution`

2. Python 3.9.3: Commonly used programming language for data analysis and algorithm implementation. `hhttps://www.python.org/downloads/release/python-393/`

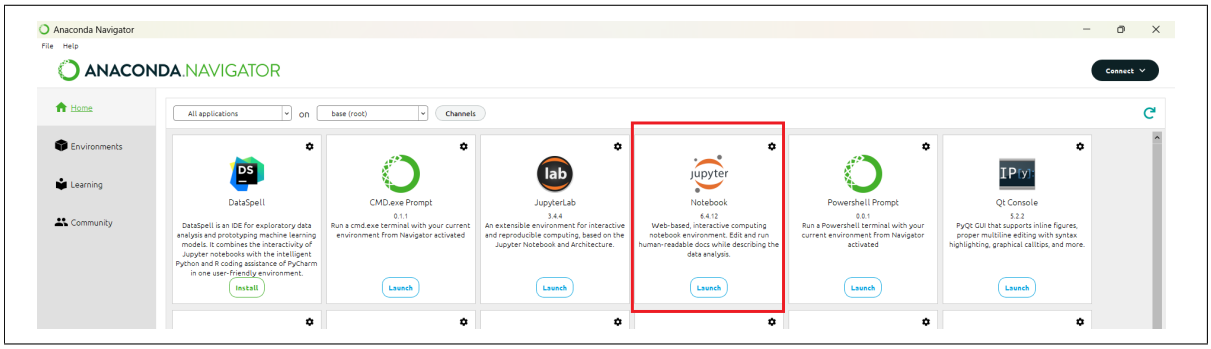

Figure 1: Anaconda Distribution

Jupyter Notebook is an excellent tool for data analysis and algorithm development, as it combines code, visualizations, and explanations in a single document.

4 Data Sources

4.1 Importing Essential Python Libraries

The Python libraries and modules listed play a pivotal role in a project aimed at determining the optimal electric vehicle charging station locations in Dublin. As shown in

Table 3, this table lists various Python libraries and modules commonly used in geospatial and data analysis tasks. Install the following essential Python libraries to support various aspects of data analysis and algorithm development:

Library/Module	Version
osmnx	1.6.0
pandas	1.4.4
folium	0.14.0
requests	2.28.1
seaborn	0.11.2
shapely.geometry	2.0.1
pyproj	3.6.0
geopandas	0.13.2
numpy	1.24.3
geopy.distance	2.3.0

Table 3: Python Libraries and Modules with Versions

The figure 2 shows the imported libraries in the project.

```
In [1]: import osmnx as ox
        import pandas as pd
        import folium
        import requests
        import matplotlib.pyplot as plt
        import seaborn as sns
        from shapely.geometry import Point, Polygon, MultiPolygon, shape
        import plotly.express as px
        import plotly.offline as pyo
        import pyproj
        import geopandas as gpd
        import warnings

        # Set up Plotly to work in offline mode
        pyo.init_notebook_mode(connected=True)
```

Figure 2: Libraries Imported

4.2 Data Acquisition

The data is collected from OpenStreetMap API [1], OpenChargeMap (OCM) API [2], Dublinked [3] and the geojson from CSO Electoral Divisions[4] for this project. This public data OpenStreetMap and OpenChargeMap accessed by using API and the data from Dublinked and CSO Electoral Divisions downloaded to the local system.

[1]https://github.com/openchargemap/ocm-system
[2]https://openchargemap.org/site/develop/api/
[3]https://www.dublincity.ie/business/economic-development-and-enterprise/smart-cities/dublinked
[4]https://www.cso.ie/en/census/census2016reports/census2016boundaryfiles/

OpenStreetMap (OSM): This data launched in 2004, OSM has developed into a massive geographic database that covers a wide range of topics including rivers, parks, and natural features in addition to roads, streets, and buildings. Volunteers, referred to as contributors, track and add data to the map, guaranteeing its accuracy and timeliness, using GPS devices, aerial images, and local expertise (*openChargeMap*; 2004). The OSM implementation code for obtaining the coordinates of Dublin's city center is displayed in Figure 3. The study retrieves Dublin County's amenities within a 25-kilometer radius.

```
In [2]: # Get the coordinates for Dublin, Ireland
        location = ox.geocode('Dublin, Ireland')

In [3]: print("The coordinates for Dublin city centre is", location)

        The coordinates for Dublin city centre is (53.3493795, -6.2605593)
```

Figure 3: OSM implementation.

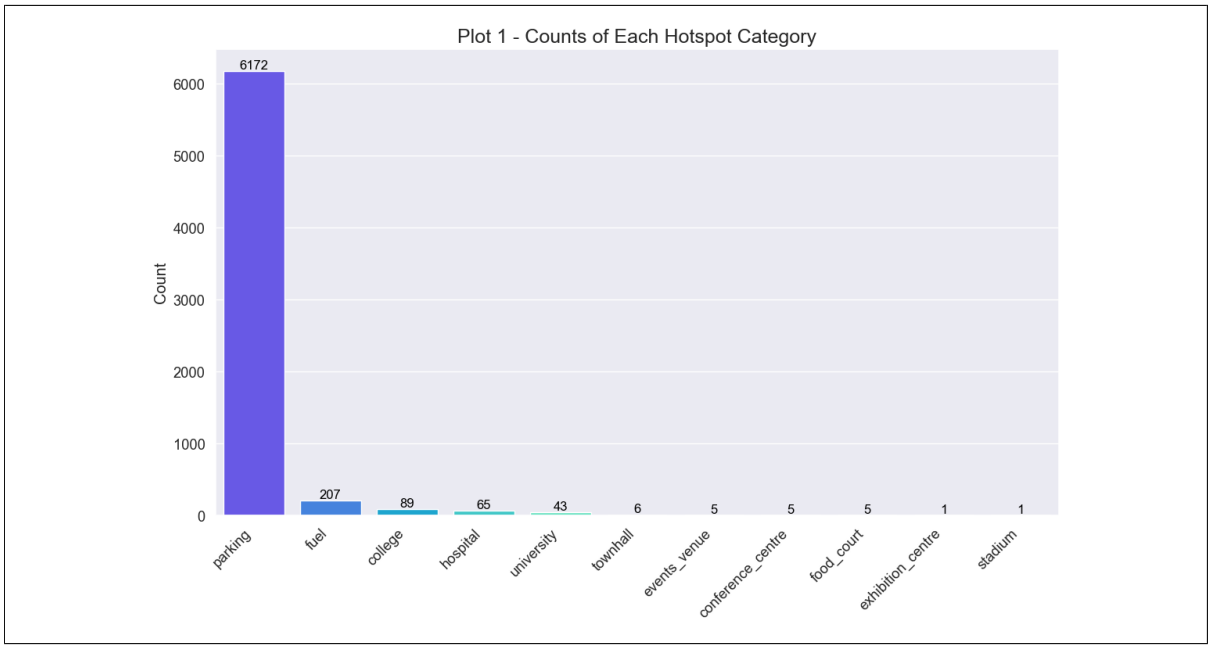

Figure 4: Feature data provided by OSM.

The number of hotspots in each hashtag category is displayed in the figure 4. With 6172 hotspots, the parking hotspot category is the most popular. Fuel and college come next, with 207 and 89 hotspots, respectively.

OpenChargeMap (OCM): Open Charge Map is the global open data registry of electric vehicle charging locations, worldwide. All of OpenChargeMap's data is usually accessible via the OpenChargeMap API, mobile apps, and the web interface (Openchargemap; n.d.). Data regarding electric vehicle charging stations can be programmatically retrieved using the application programming interface (API), refer figure 5.

This paves the way for integration with EV-related services and applications, as well as navigation systems 7. The OCM data consists of the following: UUID, AddressInfo, EVCS Connections, and connection type. The dataset has a unique key called UUID, which means that the project used an API key that OCM provided to access this data.

```
In [22]: def get_charging_stations(latitude, longitude, radius, api_key):
             url = f"https://api.openchargemap.io/v3/poi/?output=json&latitude={latitude}&longitude={longitude}&distance={radius}&dist

             try:
                 response = requests.get(url)
                 response.raise_for_status()
                 data = response.json()
                 return data

             except requests.exceptions.RequestException as e:
                 print("Error occurred while retrieving charging station data:", e)
                 return None
             except ValueError as e:
                 print("Error occurred while parsing JSON response:", e)
                 return None

         # Dublin Center coordinates
         latitude = location[0]
         longitude = location[1]

         # Radius in kilometers
         radius = 25
```

Figure 5: OCM Code

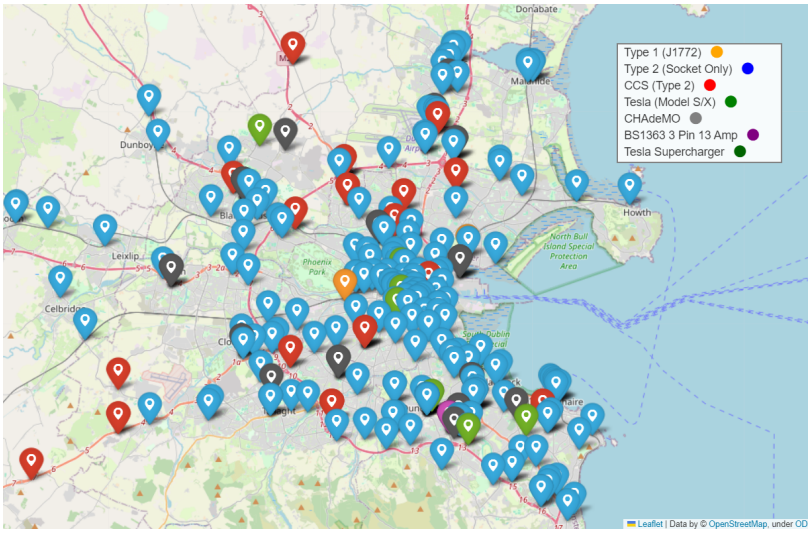

Figure 6: OCM result

Dublinked: Government agencies, local councils, and community organisations rely on Pobal's Deprivation Index, which is a part of Dublinlinked, to guide policy decisions, resource distribution, and strategic planning. Due to the absence of an API on this dataset, the project relied on a CSV file that had been downloaded on local system. This dataset focuses on Dublin's overall population as determined by the enumerative district. The city centre and its environs have the highest population density (see figure **??**), according to Dublin's population density map.

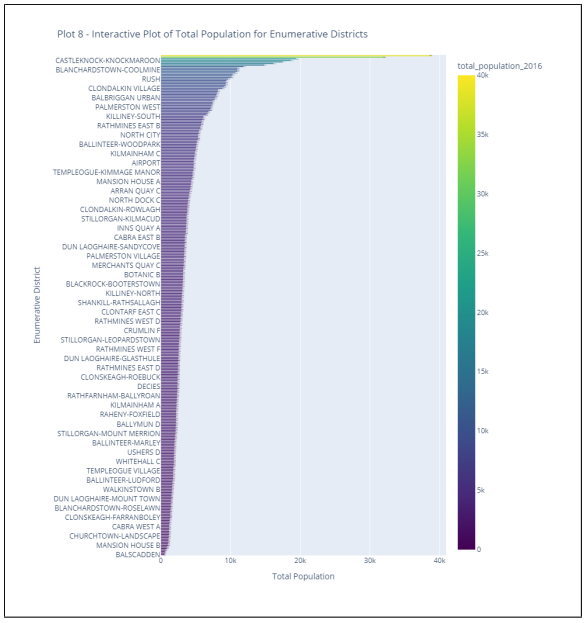

Figure 7: Total Population for Enumerative District in Dublin.

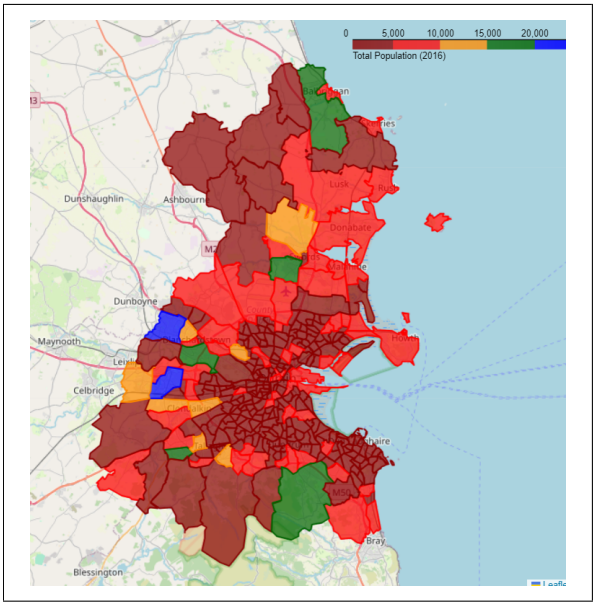

Figure 8: Dublin District Map with Population Density

CSO Electoral Divisions: GIS research, demographic studies, and administrative planning are just a few of the many useful applications that might benefit from this dataset. The collection includes 3,409 ED boundaries as an image file and a geojson file. The parameters have been refined in accordance with our licencing agreement with Tailte Éireann—National Mapping Division. By downloading these borders, any individual or organisation is doing so voluntarily and agrees to the terms and restrictions that govern their availability (*CSO Electoral Divisions, National Statistical Boundaries, generalised to 100m*; n.d.).

The district boundaries of Dublin are depicted in Figure 8, and the information regarding the population density of each district is coloured using geojson that was downloaded. Those districts that have the highest population density are represented by the colour dark red, while those districts that have the lowest population density are represented by the colour light green. From the data presented in figure 8, it can be observed that the city centre of Dublin is the most densely populated location, with a population density above 10,000 persons per square kilometre.

5 Steps to Implement and Execute the Project.

The following are the procedural stages to execute the project:

- **Step 1:** Obtain the artefact by downloading it or acquire the repository from Github [5].
- **Step 2:** Retrieve the HP-Depresvation-index[6] csv file. Additionally, download the CSO[7] Geojson file for Electoral Division.

[5]https://github.com/sr-jadhav/EVCS$_{D}ublin$
[6]https://data.gov.ie/dataset/hp-deprivation-index-scores-2016
[7]https://data.gov.ie/dataset/cso-electoral-divisions-generalised-100m3

- **Step 3:** Adjust the location based on the HP-Depresvation-index csv location in the section titled 'Pobal's HP Deprivation Index for Dublin at a District level' Changes as seen in figure 9.
- **Step 4:** Update the CSO_Electoral_Divisions_Generalised_100m geojson file present in 'Pobal's HP Deprivation Index for Dublin at a District level' section, refer figure 10
- **Step 5:** Open the 'EVCS_Dublin.ipynb' file in jupyter notebook and click on 'Kernel' -¿ 'Restart and Run all'.

Figure 9: Modify the Path as per downloaded file location for HP-Depresvation-index csv.

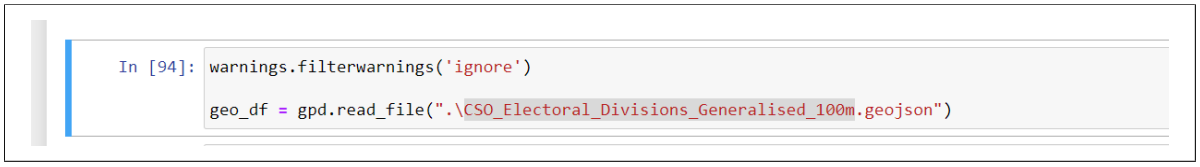

Figure 10: Replace the Path as per downloaded file location for CSO_Electoral_Divisions_Generalised_100m geojson.

References

CSO Electoral Divisions, National Statistical Boundaries, generalised to 100m (n.d.).
URL: *https://services-eu1.arcgis.com/FH5XCsx8rYXqnjF5/arcgis/rest/services/CSO$_E$lectoral$_D$ivis*

openChargeMap (2004).
URL: *https://wiki.openstreetmap.org/wiki/Researcher$_I$nformatio*

Openchargemap (n.d.). Github - openchargemap/ocm-system: Open charge map is the global public registry of electric vehicle charging locations. established 2011. help wanted.
URL: *https://github.com/openchargemap/ocm-system*