

# Clustering the cities of the Netherlands

March 3, 2021

Clustering the cities of the Netherlands based on venues

## 1 Introduction

In this project, we cluster the largest cities in the Netherlands based on venues. We take into consideration the venue categories (e.g. drugstores, cafés, bus stations, pubs, restaurants, shoe stores, bakeries, etc.) and the relative amount of these venue categories for each city.

By clustering the cities, we obtain an insight in similarity between cities. This information can be used for many purposes, such as helping tourists choose their new destination based on cities they previously enjoyed visiting. Similarly, this also helps people make decisions if they are thinking about moving within the Netherlands. Furthermore, our findings will help stakeholders make informed business decisions and address concerns they have related to competitors.

## 2 Data Description

### 2.1 Cities

We require geolocation data for the biggest cities in the Netherlands. To derive our solution, we scrape our data from [https://wikikids.nl/Lijst\\_van\\_grote\\_Nederlandse\\_steden](https://wikikids.nl/Lijst_van_grote_Nederlandse_steden)

1. *Naam* : Name of the city
2. *Inwoners* : The population of that city

This wikipedia page has information about the biggest cities in the Netherlands, including the population for each city. This wikipedia page lacks information about the geographical locations. To solve this problem we use ArcGIS API

#### 2.1.1 ArcGIS API

ArcGIS Online enables you to connect people, locations, and data using interactive maps. More specifically, we use ArcGIS to get the geo locations of the cities in the Netherlands. The following columns are added to our initial dataset which prepares our data.

3. *Latitude* : Latitude for city
4. *Longitude* : Longitude for city

### 2.2 Foursquare API Data

We will need data about different venues in different cities. In order to gain that information we will use the “Foursquare” locational information. Foursquare is a location data provider with

information about all manner of venues and events within an area of interest. Such information includes venue names, locations, menus and even photos. As such, the foursquare location platform will be used as the sole data source since all the stated required information can be obtained through the API.

After finding the list of cities, we then connect to the Foursquare API to gather information about venues inside each city. For all cities, we have chosen the radius to be 3 kilometers.

The data retrieved from Foursquare contained information of venues within a specified distance of the longitude and latitude of the postcodes. The information obtained per venue as follows:

1. *Stad* : Name of the city
2. *Stad Latitude* : Latitude of the city
3. *Stad Longitude* : Longitude of the city
4. *Venue* : Name of the venue
5. *Venue Latitude* : Latitude of venue
6. *Venue Longitude* : Longitude of venue
7. *Venue Category* : Category of venue

Based on the information collected for the cities, we have sufficient data to build our model. We cluster the cities together based on similar venue categories. We then present our observations and findings. Using this data, our stakeholders can take the necessary decisions.

```
[54]: pip install nbconvert[webpdf]
```

```
Requirement already satisfied: nbconvert[webpdf] in
c:\users\neokd\anaconda3\lib\site-packages (6.0.7)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in
c:\users\neokd\anaconda3\lib\site-packages (from nbconvert[webpdf]) (0.5.1)
Requirement already satisfied: bleach in c:\users\neokd\anaconda3\lib\site-
packages (from nbconvert[webpdf]) (3.2.1)
Requirement already satisfied: traitlets>=4.2 in
c:\users\neokd\anaconda3\lib\site-packages (from nbconvert[webpdf]) (5.0.5)
Requirement already satisfied: jinja2>=2.4 in c:\users\neokd\anaconda3\lib\site-
packages (from nbconvert[webpdf]) (2.11.2)
Requirement already satisfied: jupyterlab-pygments in
c:\users\neokd\anaconda3\lib\site-packages (from nbconvert[webpdf]) (0.1.2)
Requirement already satisfied: defusedxml in c:\users\neokd\anaconda3\lib\site-
packages (from nbconvert[webpdf]) (0.6.0)
Requirement already satisfied: pygments>=2.4.1 in
c:\users\neokd\anaconda3\lib\site-packages (from nbconvert[webpdf]) (2.7.2)
Requirement already satisfied: nbformat>=4.4 in
c:\users\neokd\anaconda3\lib\site-packages (from nbconvert[webpdf]) (5.0.8)
Requirement already satisfied: testpath in c:\users\neokd\anaconda3\lib\site-
packages (from nbconvert[webpdf]) (0.4.4)
Requirement already satisfied: jupyter-core in
c:\users\neokd\anaconda3\lib\site-packages (from nbconvert[webpdf]) (4.6.3)
Requirement already satisfied: mistune<2,>=0.8.1 in
c:\users\neokd\anaconda3\lib\site-packages (from nbconvert[webpdf]) (0.8.4)
Requirement already satisfied: entrypoints>=0.2.2 in
```

```

c:\users\neokd\anaconda3\lib\site-packages (from nbconvert[webpdf]) (0.3)
Requirement already satisfied: pandocfilters>=1.4.1 in
c:\users\neokd\anaconda3\lib\site-packages (from nbconvert[webpdf]) (1.4.3)
Collecting pypeteer==0.2.2; extra == "webpdf"
  Downloading pypeteer-0.2.2-py3-none-any.whl (145 kB)
Requirement already satisfied: jupyter-client>=6.1.5 in
c:\users\neokd\anaconda3\lib\site-packages (from
nbclient<0.6.0,>=0.5.0->nbconvert[webpdf]) (6.1.7)
Requirement already satisfied: async-generator in
c:\users\neokd\anaconda3\lib\site-packages (from
nbclient<0.6.0,>=0.5.0->nbconvert[webpdf]) (1.10)
Requirement already satisfied: nest-asyncio in
c:\users\neokd\anaconda3\lib\site-packages (from
nbclient<0.6.0,>=0.5.0->nbconvert[webpdf]) (1.4.2)
Requirement already satisfied: six>=1.9.0 in c:\users\neokd\anaconda3\lib\site-
packages (from bleach->nbconvert[webpdf]) (1.15.0)
Requirement already satisfied: webencodings in
c:\users\neokd\anaconda3\lib\site-packages (from bleach->nbconvert[webpdf])
(0.5.1)
Requirement already satisfied: packaging in c:\users\neokd\anaconda3\lib\site-
packages (from bleach->nbconvert[webpdf]) (20.4)
Requirement already satisfied: ipython-genutils in
c:\users\neokd\anaconda3\lib\site-packages (from
traitlets>=4.2->nbconvert[webpdf]) (0.2.0)
Requirement already satisfied: MarkupSafe>=0.23 in
c:\users\neokd\anaconda3\lib\site-packages (from jinja2>=2.4->nbconvert[webpdf])
(1.1.1)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in
c:\users\neokd\anaconda3\lib\site-packages (from
nbformat>=4.4->nbconvert[webpdf]) (3.2.0)
Requirement already satisfied: pywin32>=1.0; sys_platform == "win32" in
c:\users\neokd\anaconda3\lib\site-packages (from jupyter-
core->nbconvert[webpdf]) (227)
Collecting appdirs<2.0.0,>=1.4.3
  Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Requirement already satisfied: tqdm<5.0.0,>=4.42.1 in
c:\users\neokd\anaconda3\lib\site-packages (from pypeteer==0.2.2; extra ==
"webpdf"->nbconvert[webpdf]) (4.50.2)
Requirement already satisfied: urllib3<2.0.0,>=1.25.8 in
c:\users\neokd\anaconda3\lib\site-packages (from pypeteer==0.2.2; extra ==
"webpdf"->nbconvert[webpdf]) (1.25.11)
Collecting websockets<9.0,>=8.1
  Downloading websockets-8.1-cp38-cp38-win_amd64.whl (66 kB)
Collecting pyee<8.0.0,>=7.0.1
  Downloading pyee-7.0.4-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: pyzmq>=13 in c:\users\neokd\anaconda3\lib\site-
packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert[webpdf])
(19.0.2)

```

```

Requirement already satisfied: python-dateutil>=2.1 in
c:\users\neokd\anaconda3\lib\site-packages (from jupyter-
client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert[webpdf]) (2.8.1)
Requirement already satisfied: tornado>=4.1 in
c:\users\neokd\anaconda3\lib\site-packages (from jupyter-
client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert[webpdf]) (6.0.4)
Requirement already satisfied: pyparsing>=2.0.2 in
c:\users\neokd\anaconda3\lib\site-packages (from
packaging->bleach->nbconvert[webpdf]) (2.4.7)
Requirement already satisfied: pyrsistent>=0.14.0 in
c:\users\neokd\anaconda3\lib\site-packages (from
jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert[webpdf]) (0.17.3)
Requirement already satisfied: setuptools in c:\users\neokd\anaconda3\lib\site-
packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert[webpdf])
(50.3.1.post20201107)
Requirement already satisfied: attrs>=17.4.0 in
c:\users\neokd\anaconda3\lib\site-packages (from
jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert[webpdf]) (20.3.0)
Installing collected packages: appdirs, websockets, pyee, pyppeteer
Successfully installed appdirs-1.4.4 pyee-7.0.4 pyppeteer-0.2.2 websockets-8.1
Note: you may need to restart the kernel to use updated packages.

```

### 3 Methodology

```

[5]: import pandas as pd
import requests
import numpy as np
import matplotlib.cm as cm
import matplotlib.colors as colors
import folium
from sklearn.cluster import KMeans

```

#### 3.1 Exploring the largest cities in the Netherland

We scrape the webpage and take the first table. We need only the cities (Stad) and population (Inwoners) for further steps. We can drop the nr, province, and image of the city.

```

[6]: wiki_url = requests.get("https://wikikids.nl/
    ↳Lijst_van_grote_Nederlandse_steden")
wiki_data = pd.read_html(wiki_url.text)
wiki_data = wiki_data[1]
data = wiki_data.drop(labels=['Nr', 'Provincie', 'Stadsbeeld'], axis=1)
data = data.rename(columns={'Naam': 'Stad'})
data

```

```

[6]:      Stad  Inwoners
0  Amsterdam  862.965

```

```

1    Rotterdam    581.750
2      Den Haag    537.833
3      Utrecht    352.866
4      Eindhoven    231.642
..      ...      ...
58   Den Helder     56.707
59   Doetinchem     56.418
60   Hoogeveen      54.699
61   Terneuzen      54.687
62   Middelburg      47.754

```

[63 rows x 2 columns]

```
[7]: data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63 entries, 0 to 62
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Stad        63 non-null    object
1   Inwoners    63 non-null    float64
dtypes: float64(1), object(1)
memory usage: 1.1+ KB

```

## 3.2 Geolocations of the cities

### 3.2.1 ArcGis API

We need to get the geographical co-ordinates for the cities to plot out map. We will use the arcgis package to do so. Arcgis doesn't have a limitation on the number of API calls made.

```
[8]: !pip install arcgis
```

```

Requirement already satisfied: arcgis in c:\users\neokd\anaconda3\lib\site-
packages (1.8.4)
Requirement already satisfied: pywin32>=223; platform_system == "Windows" in
c:\users\neokd\anaconda3\lib\site-packages (from arcgis) (227)
Requirement already satisfied: requests-toolbelt in
c:\users\neokd\anaconda3\lib\site-packages (from arcgis) (0.9.1)
Requirement already satisfied: ipywidgets>=7 in
c:\users\neokd\anaconda3\lib\site-packages (from arcgis) (7.5.1)
Requirement already satisfied: keyring>=19 in c:\users\neokd\anaconda3\lib\site-
packages (from arcgis) (21.4.0)
Requirement already satisfied: requests-oauthlib in
c:\users\neokd\anaconda3\lib\site-packages (from arcgis) (1.3.0)
Requirement already satisfied: ujson>=3 in c:\users\neokd\anaconda3\lib\site-
packages (from arcgis) (4.0.1)
Requirement already satisfied: requests-ntlm in

```

c:\users\neokd\anaconda3\lib\site-packages (from arcgis) (1.1.0)  
Requirement already satisfied: requests-negotiate-sspi; platform\_system ==  
"Windows" in c:\users\neokd\anaconda3\lib\site-packages (from arcgis) (0.5.2)  
Requirement already satisfied: widgetsnbextension>=3 in  
c:\users\neokd\anaconda3\lib\site-packages (from arcgis) (3.5.1)  
Requirement already satisfied: lerc in c:\users\neokd\anaconda3\lib\site-  
packages (from arcgis) (0.1.0)  
Requirement already satisfied: requests-kerberos; platform\_system == "Windows"  
in c:\users\neokd\anaconda3\lib\site-packages (from arcgis) (0.12.0)  
Requirement already satisfied: six in c:\users\neokd\anaconda3\lib\site-packages  
(from arcgis) (1.15.0)  
Requirement already satisfied: requests in c:\users\neokd\anaconda3\lib\site-  
packages (from arcgis) (2.24.0)  
Requirement already satisfied: jupyterlab in c:\users\neokd\anaconda3\lib\site-  
packages (from arcgis) (2.2.6)  
Requirement already satisfied: winkerberos; platform\_system == "Windows" in  
c:\users\neokd\anaconda3\lib\site-packages (from arcgis) (0.7.0)  
Requirement already satisfied: pandas>=1 in c:\users\neokd\anaconda3\lib\site-  
packages (from arcgis) (1.1.3)  
Requirement already satisfied: pysnp>=2 in c:\users\neokd\anaconda3\lib\site-  
packages (from arcgis) (2.1.3)  
Requirement already satisfied: numpy>=1.16.2 in  
c:\users\neokd\anaconda3\lib\site-packages (from arcgis) (1.19.2)  
Requirement already satisfied: python-certifi-win32 in  
c:\users\neokd\anaconda3\lib\site-packages (from arcgis) (1.6)  
Requirement already satisfied: matplotlib in c:\users\neokd\anaconda3\lib\site-  
packages (from arcgis) (3.3.2)  
Requirement already satisfied: traitlets>=4.3.1 in  
c:\users\neokd\anaconda3\lib\site-packages (from ipywidgets>=7->arcgis) (5.0.5)  
Requirement already satisfied: ipykernel>=4.5.1 in  
c:\users\neokd\anaconda3\lib\site-packages (from ipywidgets>=7->arcgis) (5.3.4)  
Requirement already satisfied: ipython>=4.0.0; python\_version >= "3.3" in  
c:\users\neokd\anaconda3\lib\site-packages (from ipywidgets>=7->arcgis) (7.19.0)  
Requirement already satisfied: nbformat>=4.2.0 in  
c:\users\neokd\anaconda3\lib\site-packages (from ipywidgets>=7->arcgis) (5.0.8)  
Requirement already satisfied: pywin32-ctypes!=0.1.0,!0.1.1; sys\_platform ==  
"win32" in c:\users\neokd\anaconda3\lib\site-packages (from keyring>=19->arcgis)  
(0.2.0)  
Requirement already satisfied: oauthlib>=3.0.0 in  
c:\users\neokd\anaconda3\lib\site-packages (from requests-oauthlib->arcgis)  
(3.1.0)  
Requirement already satisfied: ntlm-auth>=1.0.2 in  
c:\users\neokd\anaconda3\lib\site-packages (from requests-ntlm->arcgis) (1.5.0)  
Requirement already satisfied: cryptography>=1.3 in  
c:\users\neokd\anaconda3\lib\site-packages (from requests-ntlm->arcgis) (3.1.1)  
Requirement already satisfied: pypiwin32>=223 in  
c:\users\neokd\anaconda3\lib\site-packages (from requests-negotiate-sspi;  
platform\_system == "Windows"->arcgis) (223)

Requirement already satisfied: notebook>=4.4.1 in  
c:\users\neokd\anaconda3\lib\site-packages (from widgetsnbextension>=3->arcgis)  
(6.1.4)

Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in  
c:\users\neokd\anaconda3\lib\site-packages (from requests->arcgis) (1.25.11)

Requirement already satisfied: chardet<4,>=3.0.2 in  
c:\users\neokd\anaconda3\lib\site-packages (from requests->arcgis) (3.0.4)

Requirement already satisfied: certifi>=2017.4.17 in  
c:\users\neokd\anaconda3\lib\site-packages (from requests->arcgis) (2020.6.20)

Requirement already satisfied: idna<3,>=2.5 in  
c:\users\neokd\anaconda3\lib\site-packages (from requests->arcgis) (2.10)

Requirement already satisfied: tornado!=6.0.0,!6.0.1,!6.0.2 in  
c:\users\neokd\anaconda3\lib\site-packages (from jupyterlab->arcgis) (6.0.4)

Requirement already satisfied: jupyterlab\_server<2.0,>=1.1.5 in  
c:\users\neokd\anaconda3\lib\site-packages (from jupyterlab->arcgis) (1.2.0)

Requirement already satisfied: Jinja2>=2.10 in  
c:\users\neokd\anaconda3\lib\site-packages (from jupyterlab->arcgis) (2.11.2)

Requirement already satisfied: pytz>=2017.2 in  
c:\users\neokd\anaconda3\lib\site-packages (from pandas>=1->arcgis) (2020.1)

Requirement already satisfied: python-dateutil>=2.7.3 in  
c:\users\neokd\anaconda3\lib\site-packages (from pandas>=1->arcgis) (2.8.1)

Requirement already satisfied: setuptools-scm in  
c:\users\neokd\anaconda3\lib\site-packages (from python-certifi-win32->arcgis)  
(5.0.1)

Requirement already satisfied: wrapt>=1.10.4 in  
c:\users\neokd\anaconda3\lib\site-packages (from python-certifi-win32->arcgis)  
(1.11.2)

Requirement already satisfied: cycler>=0.10 in  
c:\users\neokd\anaconda3\lib\site-packages (from matplotlib->arcgis) (0.10.0)

Requirement already satisfied: pillow>=6.2.0 in  
c:\users\neokd\anaconda3\lib\site-packages (from matplotlib->arcgis) (8.0.1)

Requirement already satisfied: pyparsing!=2.0.4,!2.1.2,!2.1.6,>=2.0.3 in  
c:\users\neokd\anaconda3\lib\site-packages (from matplotlib->arcgis) (2.4.7)

Requirement already satisfied: kiwisolver>=1.0.1 in  
c:\users\neokd\anaconda3\lib\site-packages (from matplotlib->arcgis) (1.3.0)

Requirement already satisfied: ipython-genutils in  
c:\users\neokd\anaconda3\lib\site-packages (from  
traitlets>=4.3.1->ipywidgets>=7->arcgis) (0.2.0)

Requirement already satisfied: jupyter-client in  
c:\users\neokd\anaconda3\lib\site-packages (from  
ipykernel>=4.5.1->ipywidgets>=7->arcgis) (6.1.7)

Requirement already satisfied: jedi>=0.10 in c:\users\neokd\anaconda3\lib\site-  
packages (from ipython>=4.0.0; python\_version >= "3.3"->ipywidgets>=7->arcgis)  
(0.17.1)

Requirement already satisfied: setuptools>=18.5 in  
c:\users\neokd\anaconda3\lib\site-packages (from ipython>=4.0.0; python\_version  
>= "3.3"->ipywidgets>=7->arcgis) (50.3.1.post20201107)

Requirement already satisfied: colorama; sys\_platform == "win32" in

c:\users\neokd\anaconda3\lib\site-packages (from ipython>=4.0.0; python\_version >= "3.3"->ipywidgets>=7->arcgis) (0.4.4)

Requirement already satisfied: pygments in c:\users\neokd\anaconda3\lib\site-packages (from ipython>=4.0.0; python\_version >= "3.3"->ipywidgets>=7->arcgis) (2.7.2)

Requirement already satisfied: prompt-toolkit!=3.0.0,!3.0.1,<3.1.0,>=2.0.0 in c:\users\neokd\anaconda3\lib\site-packages (from ipython>=4.0.0; python\_version >= "3.3"->ipywidgets>=7->arcgis) (3.0.8)

Requirement already satisfied: backcall in c:\users\neokd\anaconda3\lib\site-packages (from ipython>=4.0.0; python\_version >= "3.3"->ipywidgets>=7->arcgis) (0.2.0)

Requirement already satisfied: pickleshare in c:\users\neokd\anaconda3\lib\site-packages (from ipython>=4.0.0; python\_version >= "3.3"->ipywidgets>=7->arcgis) (0.7.5)

Requirement already satisfied: decorator in c:\users\neokd\anaconda3\lib\site-packages (from ipython>=4.0.0; python\_version >= "3.3"->ipywidgets>=7->arcgis) (4.4.2)

Requirement already satisfied: jupyter-core in c:\users\neokd\anaconda3\lib\site-packages (from nbformat>=4.2.0->ipywidgets>=7->arcgis) (4.6.3)

Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in c:\users\neokd\anaconda3\lib\site-packages (from nbformat>=4.2.0->ipywidgets>=7->arcgis) (3.2.0)

Requirement already satisfied: cffi!=1.11.3,>=1.8 in c:\users\neokd\anaconda3\lib\site-packages (from cryptography>=1.3->requests-ntlm->arcgis) (1.14.3)

Requirement already satisfied: Send2Trash in c:\users\neokd\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension>=3->arcgis) (1.5.0)

Requirement already satisfied: terminado>=0.8.3 in c:\users\neokd\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.9.1)

Requirement already satisfied: nbconvert in c:\users\neokd\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension>=3->arcgis) (6.0.7)

Requirement already satisfied: pyzmq>=17 in c:\users\neokd\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension>=3->arcgis) (19.0.2)

Requirement already satisfied: prometheus-client in c:\users\neokd\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.8.0)

Requirement already satisfied: argon2-cffi in c:\users\neokd\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension>=3->arcgis) (20.1.0)

Requirement already satisfied: json5 in c:\users\neokd\anaconda3\lib\site-packages (from jupyterlab\_server<2.0,>=1.1.5->jupyterlab->arcgis) (0.9.5)

Requirement already satisfied: MarkupSafe>=0.23 in c:\users\neokd\anaconda3\lib\site-packages (from jinja2>=2.10->jupyterlab->arcgis) (1.1.1)

Requirement already satisfied: parso<0.8.0,>=0.7.0 in c:\users\neokd\anaconda3\lib\site-packages (from jedi>=0.10->ipython>=4.0.0; python\_version >= "3.3"->ipywidgets>=7->arcgis) (0.7.0)



Requirement already satisfied: wcwidth in c:\users\neokd\anaconda3\lib\site-packages (from prompt-toolkit!=3.0.0,!>=3.0.1,<3.1.0,>=2.0.0->ipython>=4.0.0; python\_version >= "3.3"->ipywidgets>=7->arcgis) (0.2.5)

Requirement already satisfied: attrs>=17.4.0 in c:\users\neokd\anaconda3\lib\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.2.0->ipywidgets>=7->arcgis) (20.3.0)

Requirement already satisfied: pyparsing>=0.14.0 in c:\users\neokd\anaconda3\lib\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.2.0->ipywidgets>=7->arcgis) (0.17.3)

Requirement already satisfied: pycparser in c:\users\neokd\anaconda3\lib\site-packages (from cffi!=1.11.3,>=1.8->cryptography>=1.3->requests-ntlm->arcgis) (2.20)

Requirement already satisfied: pywinpty>=0.5 in c:\users\neokd\anaconda3\lib\site-packages (from terminado>=0.8.3->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.5.7)

Requirement already satisfied: jupyterlab-pygments in c:\users\neokd\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.1.2)

Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in c:\users\neokd\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.5.1)

Requirement already satisfied: mistune<2,>=0.8.1 in c:\users\neokd\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.8.4)

Requirement already satisfied: testpath in c:\users\neokd\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.4.4)

Requirement already satisfied: defusedxml in c:\users\neokd\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.6.0)

Requirement already satisfied: pandocfilters>=1.4.1 in c:\users\neokd\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (1.4.3)

Requirement already satisfied: bleach in c:\users\neokd\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (3.2.1)

Requirement already satisfied: entrypoints>=0.2.2 in c:\users\neokd\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.3)

Requirement already satisfied: nest-asyncio in c:\users\neokd\anaconda3\lib\site-packages (from nbclient<0.6.0,>=0.5.0->nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (1.4.2)

Requirement already satisfied: async-generator in c:\users\neokd\anaconda3\lib\site-packages (from nbclient<0.6.0,>=0.5.0->nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (1.10)

Requirement already satisfied: webencodings in c:\users\neokd\anaconda3\lib\site-packages (from bleach->nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.5.1)

Requirement already satisfied: packaging in c:\users\neokd\anaconda3\lib\site-packages (from bleach->nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (20.4)

```
[9]: from arcgis.geocoding import geocode
     from arcgis.gis import GIS
     gis = GIS()
```

Defining arcgis geocode function to return latitude and longitude for all cities in the Netherlands

```
[10]: def get_x_y(address1):
      lat_coords = 0
      lng_coords = 0
      g = geocode(address='{ }, Netherlands, NL'.format(address1))[0]
      lng_coords = g['location']['x']
      lat_coords = g['location']['y']
      return str(lat_coords) + "," + str(lng_coords)
```

We copy over the city names to pass it into the geolocator function that we defined above

```
[11]: geo_coordinates = data['Stad']
      coordinates_latlng = geo_coordinates.apply(lambda x: get_x_y(x))
      coordinates_latlng
```

```
[11]: 0      52.369930000000007,4.907880000000034
      1      51.914380000000005,4.487160000000074
      2      52.084090000000006,4.317320000000052
      3      52.089650000000006,5.114350000000059
      4      51.435880000000005,5.485460000000046
      ...
      58     52.958380000000003,4.758910000000071
      59     51.963700000000007,6.291360000000054
      60     52.725970000000007,6.475520000000074
      61     51.338140000000007,3.8275500000000306
      62     51.495410000000005,3.609640000000007
      Name: Stad, Length: 63, dtype: object
```

### 3.2.2 Latitude & Longitude

We extract the latitude and longitude from the collected coordinates and merge them with our source data.

```
[12]: lat = coordinates_latlng.apply(lambda x: x.split(',')[0])
      lng = coordinates_latlng.apply(lambda x: x.split(',')[1])
```

```
[13]: merged = pd.concat([data, lat.astype(float), lng.astype(float)], axis=1)
      merged.columns= ['Stad', 'Inwoners', 'Latitude', 'Longitude']
      merged
```

```
[13]:
```

	Stad	Inwoners	Latitude	Longitude
0	Amsterdam	862.965	52.36993	4.90788
1	Rotterdam	581.750	51.91438	4.48716
2	Den Haag	537.833	52.08409	4.31732
3	Utrecht	352.866	52.08965	5.11435
4	Eindhoven	231.642	51.43588	5.48546
..	...	...	...	...
58	Den Helder	56.707	52.95838	4.75891
59	Doetinchem	56.418	51.96370	6.29136
60	Hoogeveen	54.699	52.72597	6.47552
61	Terneuzen	54.687	51.33814	3.82755
62	Middelburg	47.754	51.49541	3.60964

[63 rows x 4 columns]

```
[14]: merged.dtypes
```

```
[14]: Stad          object
Inwoners      float64
Latitude      float64
Longitude     float64
dtype: object
```

### 3.2.3 Co-ordinates for the Netherlands

Getting the geocode for the Netherlands so we can center it on the map

```
[15]: nederland = geocode(address='Netherlands, NL')[0]
nederland_lng_coords = nederland['location']['x']
nederland_lat_coords = nederland['location']['y']
print('Coordinates:', nederland_lng_coords, nederland_lat_coords)
```

Coordinates: 5.616126398000063 52.24937529300007

## 3.3 Visualize the map of the Netherlands

To help visualize the map of the Netherlands and its cities, we make use of the folium package. The size of the marker is based on the population of the specific city.

```
[22]: # Creating the map of the Netherlands
map_nederland = folium.Map(location=[nederland_lat_coords,
↳nederland_lng_coords], zoom_start=7, tiles='cartodbpositron')
map_nederland

# adding markers to map
for latitude, longitude, stad, inwoners in zip(merged['Latitude'],
↳merged['Longitude'], merged['Stad'], merged['Inwoners']):
    label = '{}'.format(stad)
```

```

label = folium.Popup(label, parse_html=True)
radius_size = inwoners / 18
folium.CircleMarker(
    [latitude, longitude],
    radius=radius_size,
    popup=label,
    color='blue',
    fill=True
).add_to(map_nederland)

map_nederland

```

[22]: <folium.folium.Map at 0x25f97145f70>

```

[23]: CLIENT_ID = 'ZTZLCINMLPQ4DAMDLVW4UNTLEY5SWXVK5X2WEJEKM5DBKBU1'
CLIENT_SECRET = 'SAUEHHCYE1ZEVBYVDCNF1H5W1MT2QOKGN11KSLN3GLVYM3J'
VERSION = '20210225'

```

```

[24]: LIMIT=200

def getNearbyVenues(names, latitudes, longitudes, radius=3000):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?
↳&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT
        )

        # make the GET request
        results = requests.get(url).json()["response"]["groups"][0]["items"]

        # return only relevant information for each nearby venue
        venues_list.append([
            name,
            lat,
            lng,
            v['venue']['name'],

```

```

        v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item
    ↪in venue_list])
    nearby_venues.columns = ['Stad',
                             'Stad Latitude',
                             'Stad Longitude',
                             'Venue',
                             'Venue Category']

    return(nearby_venues)

```

### 3.3.1 Venues in the Netherlands

To proceed with the next part, we need to define Foursquare API credentials.

Using Foursquare API, we are able to get the venue and venue categories in each city in the Netherlands.

```

[25]: venues = getNearbyVenues(merged['Stad'], merged['Latitude'],
    ↪merged['Longitude'])

```

```

Amsterdam
Rotterdam
Den Haag
Utrecht
Eindhoven
Tilburg
Groningen
Almere
Breda
Nijmegen
Enschede
Apeldoorn
Haarlem
Amersfoort
Zaanstad
Arnhem
Haarlemmermeer
's Hertogenbosch
Zoetermeer
Zwolle
Maastricht
Leiden
Dordrecht
Ede
Emmen
Westland
Venlo

```

Delft  
 Deventer  
 Leeuwarden  
 Alkmaar  
 Sittard-Geleen  
 Helmond  
 Heerlen  
 Hilversum  
 Oss  
 Amstelveen  
 Súdwest-Fryslân  
 Hengelo  
 Purmerend  
 Roosendaal  
 Schiedam  
 Lelystad  
 Alphen aan den Rijn  
 Leidschendam-Voorburg  
 Almelo  
 Spijkenisse  
 Hoorn  
 Gouda  
 Vlaardingen  
 Assen  
 Bergen op Zoom  
 Capelle aan den IJssel  
 Veenendaal  
 Katwijk  
 Zeist  
 Nieuwegein  
 Roermond  
 Den Helder  
 Doetinchem  
 Hoogeveen  
 Terneuzen  
 Middelburg

```
[26]: venues.head()
```

```

[26]:      Stad  Stad Latitude  Stad Longitude  Venue \
0  Amsterdam      52.36993      4.90788      HPS
1  Amsterdam      52.36993      4.90788      Sotto
2  Amsterdam      52.36993      4.90788  De Hortus
3  Amsterdam      52.36993      4.90788  Black Gold
4  Amsterdam      52.36993      4.90788  Rosalia's Menagerie

      Venue Category
  
```

```

0      Cocktail Bar
1      Pizza Place
2      Botanical Garden
3      Coffee Shop
4      Cocktail Bar

```

```
[27]: venues.shape
```

```
[27]: (5083, 5)
```

In total, we have scraped 5083 venues for 62 cities.

### 3.3.2 Grouping by Venue Categories

We will check how many Venue Categories there are for further processing

```
[28]: venues.groupby('Venue Category').max()
```

```
[28]:
```

	Stad	Stad Latitude	Stad Longitude	\
Venue Category				
Accessories Store	Roermond	51.19614	5.98372	
Advertising Agency	Zwolle	52.51621	6.09247	
Afghan Restaurant	Tilburg	51.69088	5.48546	
African Restaurant	Amersfoort	52.15252	5.38626	
Airport	Hoogeveen	52.72597	6.47552	
...	...	...	...	
Wine Shop	Utrecht	53.21687	6.57393	
Women's Store	Rotterdam	51.91438	4.48716	
Yoga Studio	Den Haag	52.36993	4.90788	
Zoo	Maastricht	52.78223	6.89636	
Zoo Exhibit	Emmen	52.78223	6.89636	

  

	Venue
Venue Category	
Accessories Store	Michael Kors Outlet
Advertising Agency	Dopit Media
Afghan Restaurant	Zaher
African Restaurant	Restaurant De Olifant
Airport	Vliegveld Hoogeveen (EHHO)
...	...
Wine Shop	Wijnkoperij Platenburg
Women's Store	Dearhunter Vintage Clothing & Accessories
Yoga Studio	Delight Yoga
Zoo	Wildlands Adventure Zoo Emmen
Zoo Exhibit	Vlindertempel

[334 rows x 4 columns]

We can see 334 records, indicating a great diversity in venues / very well-defined venues in the Netherlands.

### 3.3.3 One Hot Encoding

We need to encode our venue categories for our clustering

```
[29]: venue_cat = pd.get_dummies(venues[['Venue Category']], prefix="", prefix_sep="")
venue_cat
```

```
[29]:
```

	Accessories Store	Advertising Agency	Afghan Restaurant	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
...	...	...	...	
5078	0	0	0	
5079	0	0	0	
5080	0	0	0	
5081	0	0	0	
5082	0	0	0	

  

	African Restaurant	Airport	American Restaurant	Amphitheater	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	
...	...	...	...	...	
5078	0	0	0	0	
5079	0	0	0	0	
5080	0	0	0	0	
5081	0	0	0	0	
5082	0	0	0	0	

  

	Apres Ski Bar	Arcade	Arepa Restaurant	...	Warehouse Store	\
0	0	0	0	...	0	
1	0	0	0	...	0	
2	0	0	0	...	0	
3	0	0	0	...	0	
4	0	0	0	...	0	
...	...	...	...	...	...	
5078	0	0	0	...	0	
5079	0	0	0	...	0	
5080	0	0	0	...	0	
5081	0	0	0	...	0	
5082	0	0	0	...	0	



	Waterfront	Whisky Bar	Windmill	Wine Bar	Wine Shop	Women's Store	\
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	0	0	0	0	0	
3	0	0	0	0	0	0	
4	0	0	0	0	0	0	
...	...	...	...	...	...	...	
5078	0	0	0	0	0	0	
5079	0	0	0	0	0	0	
5080	0	0	0	0	0	0	
5081	0	0	0	0	0	0	
5082	0	0	0	0	0	0	

	Yoga Studio	Zoo	Zoo Exhibit
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
...	...	...	...
5078	0	0	0
5079	0	0	0
5080	0	0	0
5081	0	0	0
5082	0	0	0

[5083 rows x 334 columns]

We add the cities as first column

```
[30]: venue_cat['Stad'] = venues['Stad']

# moving city column to the first column
fixed_columns = [venue_cat.columns[-1]] + list(venue_cat.columns[:-1])
venue_cat = venue_cat[fixed_columns]

venue_cat.head()
```

```
[30]:
```

	Stad	Accessories Store	Advertising Agency	Afghan Restaurant	\
0	Amsterdam	0	0	0	
1	Amsterdam	0	0	0	
2	Amsterdam	0	0	0	
3	Amsterdam	0	0	0	
4	Amsterdam	0	0	0	

  

	African Restaurant	Airport	American Restaurant	Amphitheater	\
0	0	0	0	0	

1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

	Apres Ski Bar	Arcade	...	Warehouse Store	Waterfront	Whisky Bar	\
0	0	0	...	0	0	0	
1	0	0	...	0	0	0	
2	0	0	...	0	0	0	
3	0	0	...	0	0	0	
4	0	0	...	0	0	0	

	Windmill	Wine Bar	Wine Shop	Women's Store	Yoga Studio	Zoo	Zoo Exhibit
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0

[5 rows x 335 columns]

### 3.3.4 Venue categories mean value

We will group by cities and calculate the mean venue categories value in each city

```
[31]: grouped = venue_cat.groupby('Stad').mean().reset_index()
grouped.head()
```

```
[31]:
```

	Stad	Accessories Store	Advertising Agency	\
0	's Hertogenbosch	0.0	0.0	
1	Alkmaar	0.0	0.0	
2	Almelo	0.0	0.0	
3	Almere	0.0	0.0	
4	Alphen aan den Rijn	0.0	0.0	

	Afghan Restaurant	African Restaurant	Airport	American Restaurant	\
0	0.01	0.0	0.0	0.00	
1	0.00	0.0	0.0	0.00	
2	0.00	0.0	0.0	0.00	
3	0.00	0.0	0.0	0.01	
4	0.00	0.0	0.0	0.00	

	Amphitheater	Apres Ski Bar	Arcade	...	Warehouse Store	Waterfront	\
0	0.0	0.0	0.0	...	0.01	0.0	
1	0.0	0.0	0.0	...	0.00	0.0	
2	0.0	0.0	0.0	...	0.00	0.0	
3	0.0	0.0	0.0	...	0.00	0.0	
4	0.0	0.0	0.0	...	0.00	0.0	

	Whisky Bar	Windmill	Wine Bar	Wine Shop	Women's Store	Yoga Studio \
0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0

	Zoo	Zoo Exhibit
0	0.000000	0.0
1	0.000000	0.0
2	0.000000	0.0
3	0.000000	0.0
4	0.012658	0.0

[5 rows x 335 columns]

The following function will be used to get the top most common venue categories

```
[34]: def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

Since there are too many venue categories (334), we only take the top 25 to cluster the cities.

The following function is used to label the columns of the venue correctly

```
[35]: num_top_venues = 25

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Stad']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{} {} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))
```

### 3.3.5 Top venue categories

Getting the top venue categories for each city in the Netherlands

```
[36]: # create a new dataframe
venues_sorted2 = pd.DataFrame(columns=columns)
venues_sorted2['Stad'] = grouped['Stad']
```

```

for ind in np.arange(grouped.shape[0]):
    venues_sorted2.iloc[ind, 1:] = return_most_common_venues(grouped.iloc[ind, :
→], num_top_venues)

venues_sorted2.head()

```

```

[36]:
      Stad 1st Most Common Venue 2nd Most Common Venue \
0      's Hertogenbosch      Supermarket      Bar
1      Alkmaar      Supermarket      Coffee Shop
2      Almelo      Supermarket      Shopping Mall
3      Almere      Supermarket      Sushi Restaurant
4  Alphen aan den Rijn      Supermarket      Restaurant

      3rd Most Common Venue 4th Most Common Venue      5th Most Common Venue \
0      Café      Restaurant      Mediterranean Restaurant
1      Café      Restaurant      Fast Food Restaurant
2      Drugstore      Café      Restaurant
3      Restaurant      Snack Place      Bar
4      Drugstore      Gym      Soccer Field

      6th Most Common Venue 7th Most Common Venue 8th Most Common Venue \
0      Drugstore      Gastropub      Coffee Shop
1      Park      Italian Restaurant      Drugstore
2      Pub      Department Store      Grocery Store
3      Hotel      Gym      Chinese Restaurant
4      Bookstore      Discount Store      Japanese Restaurant

      9th Most Common Venue ... 16th Most Common Venue 17th Most Common Venue \
0      Hotel ...      Italian Restaurant      Gym / Fitness Center
1      French Restaurant ...      Pharmacy      Soccer Field
2  Furniture / Home Store ...      Stadium      Soccer Stadium
3      Train Station ...      Gym / Fitness Center      Park
4      Italian Restaurant ...      Pizza Place      Motorsports Shop

      18th Most Common Venue      19th Most Common Venue 20th Most Common Venue \
0      Art Museum      Beer Garden      Spa
1      Pizza Place      Pool      Sandwich Place
2      Soccer Field      Flea Market      Food & Drink Shop
3      Asian Restaurant      Hockey Field      Spa
4      Bowling Alley      Middle Eastern Restaurant      Farmers Market

      21th Most Common Venue 22th Most Common Venue 23th Most Common Venue \
0      Bistro      Bookstore      Big Box Store
1      Church      Scottish Restaurant      Skating Rink
2      Garden Center      Multiplex      Sandwich Place
3      Pool      Discount Store      Soccer Stadium
4      Spanish Restaurant      Sporting Goods Shop      Fast Food Restaurant

```

	24th Most Common Venue	25th Most Common Venue
0	Garden Center	Frozen Yogurt Shop
1	Cheese Shop	Comfort Food Restaurant
2	Cafeteria	Discount Store
3	Soccer Field	Cultural Center
4	Sports Bar	French Restaurant

[5 rows x 26 columns]

### 3.3.6 K Means clustering

Let's cluster the cities to using K Means clustering.

```
[37]: # set number of clusters
k_num_clusters = 5

grouped_clustering = grouped.drop('Stad', 1)

# run k-means clustering
kmeans_nl5 = KMeans(n_clusters=k_num_clusters, random_state=0).
    ↪ fit(grouped_clustering)
kmeans_nl5
```

```
[37]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
            n_clusters=5, n_init=10, n_jobs=None, precompute_distances='auto',
            random_state=0, tol=0.0001, verbose=0)
```

### 3.3.7 Labelling Clustered Data

```
[38]: kmeans_nl5.labels_
```

```
[38]: array([2, 2, 1, 2, 1, 0, 4, 4, 0, 0, 2, 2, 4, 1, 4, 4, 2, 4, 2, 0, 2, 4,
           2, 2, 0, 4, 4, 0, 0, 0, 2, 4, 1, 0, 3, 2, 0, 1, 1, 2, 0, 1, 4, 2,
           2, 2, 0, 4, 2, 2, 1, 0, 0, 4, 4, 1, 2, 1, 3, 2, 2, 2, 2])
```

```
[41]: venues_sorted2.insert(0, 'Cluster Labels', kmeans_nl5.labels_ + 1)
```

Join merged with our venues\_sorted to add latitude & longitude for each of the city to prepare it for plotting

```
[42]: nl_data = merged

nl_data = nl_data.join(venues_sorted2.set_index('Stad'), on='Stad')

nl_data.head()
```

```

[42]:      Stad  Inwoners  Latitude  Longitude  Cluster Labels  \
0  Amsterdam  862.965  52.36993   4.90788           5
1  Rotterdam  581.750  51.91438   4.48716           5
2   Den Haag  537.833  52.08409   4.31732           5
3   Utrecht  352.866  52.08965   5.11435           5
4  Eindhoven  231.642  51.43588   5.48546           5

      1st Most Common Venue 2nd Most Common Venue 3rd Most Common Venue  \
0                Hotel      Coffee Shop                Bar
1                Bar                Hotel      Coffee Shop
2      Coffee Shop      Restaurant                Park
3      Coffee Shop      Restaurant                Park
4      Coffee Shop                Bar      Restaurant

      4th Most Common Venue 5th Most Common Venue ... 16th Most Common Venue  \
0      Restaurant                Café ...      Yoga Studio
1                Café      French Restaurant ...      Japanese Restaurant
2                Bar                Hotel ...                Gym
3                Bar      Italian Restaurant ...                Café
4      Hotel                Plaza ...      French Restaurant

      17th Most Common Venue 18th Most Common Venue 19th Most Common Venue  \
0                Pub      Food Truck      French Restaurant
1      Bagel Shop                Market      Vietnamese Restaurant
2      Thai Restaurant      Indonesian Restaurant      Plaza
3      Sandwich Place                Hotel      Farm
4      Bakery      Breakfast Spot      Skate Park

      20th Most Common Venue 21th Most Common Venue 22th Most Common Venue  \
0      Beer Bar      Spiritual Center      Steakhouse
1      Hostel      Italian Restaurant      Shopping Plaza
2      Asian Restaurant      Butcher      Bike Shop
3      Monument / Landmark      Beer Store      Steakhouse
4      Science Museum      Skating Rink      Lounge

      23th Most Common Venue 24th Most Common Venue 25th Most Common Venue
0      Beer Store      Soup Place      Indonesian Restaurant
1      Bookstore      Food Truck      Sporting Goods Shop
2      Wine Bar      Soccer Field      Snack Place
3      Cocktail Bar      Squash Court      Snack Place
4      Bookstore      Soccer Stadium      Food Court

```

[5 rows x 30 columns]

Drop all the NaN values to prevent data skew

```

[43]: nl_data_nonan = nl_data.dropna(subset=['Cluster Labels'])

```

## 4 Results

### 4.1 Examining our Clusters

```
[51]: pd.set_option('display.max_rows', 64)
nl_data.sort_values('Cluster Labels')
```

```
[51]:
```

	Stad	Inwoners	Latitude	Longitude	Cluster Labels	\
62	Middelburg	47.754	51.49541	3.60964	1	
61	Terneuzen	54.687	51.33814	3.82755	1	
37	Súdwest-Fryslân	82.572	53.03369	5.66133	1	
40	Roosendaal	77.097	51.53141	4.45749	1	
22	Dordrecht	118.702	51.81195	4.65647	1	
21	Leiden	120.105	52.15363	4.49381	1	
47	Hoorn	71.567	52.64243	5.05206	1	
32	Helmond	89.139	51.48223	5.65825	1	
15	Arnhem	150.354	51.98038	5.90333	1	
48	Gouda	70.981	52.01000	4.71071	1	
13	Amersfoort	150.492	52.15252	5.38626	1	
16	Haarlemmermeer	144.153	52.30539	4.69296	1	
33	Heerlen	88.461	50.88578	5.98061	1	
11	Apeldoorn	157.553	52.21652	5.96566	1	
60	Hoogeveen	54.699	52.72597	6.47552	2	
42	Lelystad	76.081	52.50605	5.47401	2	
43	Alphen aan den Rijn	72.954	52.12743	4.65882	2	
56	Nieuwegein	60.997	52.02757	5.08227	2	
44	Leidschendam-Voorburg	72.962	52.08651	4.39118	2	
49	Vlaardingenv	70.863	51.91115	4.33818	2	
46	Spijkenisse	72.499	51.84976	4.32518	2	
53	Veenendaal	63.022	52.02435	5.55375	2	
52	Capelle aan den IJssel	66.166	51.92956	4.58897	2	
45	Almelo	72.519	52.35495	6.66454	2	
55	Zeist	61.233	52.08468	5.24262	3	
51	Bergen op Zoom	66.319	51.49598	4.28200	3	
41	Schiedam	76.538	51.91847	4.39816	3	
58	Den Helder	56.707	52.95838	4.75891	3	
39	Purmerend	79.512	52.50767	4.94779	3	
38	Hengelo	80.999	52.26636	6.78972	3	
59	Doetinchem	56.418	51.96370	6.29136	3	
35	Oss	84.944	51.76818	5.52670	3	
50	Assen	67.153	52.99358	6.55897	3	
57	Roermond	56.937	51.19614	5.98372	3	
31	Sittard-Geleen	93.843	50.99637	5.86779	3	
30	Alkmaar	94.958	52.63275	4.75175	3	
7	Almere	207.904	52.37670	5.22280	3	
10	Enschede	158.343	52.22360	6.89550	3	
17	's Hertogenbosch	143.373	51.69088	5.29823	3	
18	Zoetermeer	123.505	52.06310	4.48887	3	

19	Zwolle	122.737	52.51621	6.09247	3
20	Maastricht	121.317	50.84981	5.68829	3
23	Ede	110.253	52.04543	5.66730	3
14	Zaanstad	150.384	52.44953	4.80934	3
26	Venlo	100.335	51.37234	6.17304	3
29	Leeuwarden	96.174	53.19673	5.79230	3
24	Emmen	108.132	52.78223	6.89636	3
25	Westland	103.338	52.00013	4.15888	4
54	Katwijk	62.799	52.20518	4.39659	4
1	Rotterdam	581.750	51.91438	4.48716	5
2	Den Haag	537.833	52.08409	4.31732	5
3	Utrecht	352.866	52.08965	5.11435	5
4	Eindhoven	231.642	51.43588	5.48546	5
5	Tilburg	197.020	51.55541	5.10581	5
6	Groningen	202.285	53.21687	6.57393	5
36	Amstelveen	84.646	52.30485	4.85681	5
9	Nijmegen	176.731	51.84142	5.85801	5
12	Haarlem	154.352	52.38868	4.63909	5
34	Hilversum	86.279	52.22856	5.16950	5
28	Deventer	98.326	52.25103	6.15989	5
27	Delft	98.830	52.00878	4.36534	5
8	Breda	150.520	51.59134	4.77168	5
0	Amsterdam	862.965	52.36993	4.90788	5

	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	\
62	Bar	Restaurant	Café	
61	Restaurant	Supermarket	Hotel	
37	Supermarket	Restaurant	Harbor / Marina	
40	Restaurant	Bar	Supermarket	
22	Restaurant	Café	Ice Cream Shop	
21	Restaurant	Supermarket	Museum	
47	Harbor / Marina	Restaurant	Supermarket	
32	Restaurant	Supermarket	Fast Food Restaurant	
15	Restaurant	Supermarket	Café	
48	Supermarket	Café	Restaurant	
13	Restaurant	Coffee Shop	Café	
16	Restaurant	Supermarket	Hotel	
33	Restaurant	Italian Restaurant	Supermarket	
11	Café	Supermarket	French Restaurant	
60	Supermarket	Restaurant	Sandwich Place	
42	Supermarket	Restaurant	Pharmacy	
43	Supermarket	Restaurant	Drugstore	
56	Supermarket	Restaurant	Drugstore	
44	Supermarket	Italian Restaurant	Restaurant	
49	Supermarket	Shopping Mall	Hotel	
46	Supermarket	Restaurant	Shopping Mall	
53	Supermarket	Fast Food Restaurant	Shopping Mall	



52	Supermarket	Shopping Mall	Drugstore
45	Supermarket	Shopping Mall	Drugstore
55	Supermarket	Restaurant	Drugstore
51	Supermarket	Restaurant	Fast Food Restaurant
41	Supermarket	Fast Food Restaurant	Shopping Mall
58	Supermarket	Hotel	Boat or Ferry
39	Supermarket	Bus Stop	Shopping Mall
38	Bar	Supermarket	Drugstore
59	Supermarket	Drugstore	Restaurant
35	Supermarket	Coffee Shop	Hotel
50	Supermarket	Hotel	Restaurant
57	Clothing Store	Gastropub	Hotel
31	Supermarket	Bar	Restaurant
30	Supermarket	Coffee Shop	Café
7	Supermarket	Sushi Restaurant	Restaurant
10	Supermarket	Ice Cream Shop	Park
17	Supermarket	Bar	Café
18	Supermarket	Shopping Mall	Restaurant
19	Bar	Supermarket	Restaurant
20	Supermarket	Hotel	Bar
23	Supermarket	Restaurant	Drugstore
14	Supermarket	Gym	Furniture / Home Store
26	Supermarket	Bar	Restaurant
29	Supermarket	Restaurant	Soccer Field
24	Supermarket	Shopping Mall	Snack Place
25	Supermarket	Restaurant	Beach
54	Beach	Supermarket	Restaurant
1	Bar	Hotel	Coffee Shop
2	Coffee Shop	Restaurant	Park
3	Coffee Shop	Restaurant	Park
4	Coffee Shop	Bar	Restaurant
5	Bar	Café	Restaurant
6	Bar	Restaurant	Park
36	Park	Japanese Restaurant	Coffee Shop
9	Restaurant	Bar	Pub
12	Restaurant	Bar	Café
34	Coffee Shop	Tapas Restaurant	French Restaurant
28	Restaurant	Hotel	Gastropub
27	Bakery	Soccer Field	Café
8	Coffee Shop	Bar	Italian Restaurant
0	Hotel	Coffee Shop	Bar

	4th Most Common Venue	5th Most Common Venue	...	\
62	Hotel	Plaza	...	
61	Bakery	Clothing Store	...	
37	Gastropub	Drugstore	...	
40	Bus Stop	Shopping Mall	...	

22	Sushi Restaurant	Bar	...
21	Bar	Drugstore	...
47	Bar	History Museum	...
32	Ice Cream Shop	Drugstore	...
15	Park	Coffee Shop	...
48	Drugstore	Bar	...
13	Bar	Exhibit	...
16	Shopping Mall	Bakery	...
33	Music Venue	Department Store	...
11	Restaurant	Coffee Shop	...
60	Shopping Mall	Drugstore	...
42	Drugstore	Athletics & Sports	...
43	Gym	Soccer Field	...
56	Bus Stop	Museum	...
44	Ice Cream Shop	French Restaurant	...
49	Fast Food Restaurant	Drugstore	...
46	Drugstore	Fast Food Restaurant	...
53	Restaurant	Gym / Fitness Center	...
52	Restaurant	Gym / Fitness Center	...
45	Café	Restaurant	...
55	Hotel	Gym / Fitness Center	...
51	Ice Cream Shop	Bar	...
41	Soccer Field	Diner	...
58	Café	Harbor / Marina	...
39	Drugstore	Restaurant	...
38	Restaurant	Electronics Store	...
59	Hotel	Bar	...
35	Drugstore	Italian Restaurant	...
50	Bar	Bakery	...
57	Restaurant	Boutique	...
31	Gym / Fitness Center	Gastropub	...
30	Restaurant	Fast Food Restaurant	...
7	Snack Place	Bar	...
10	Gym	Theater	...
17	Restaurant	Mediterranean Restaurant	...
18	Park	Theater	...
19	Park	Bakery	...
20	Coffee Shop	Restaurant	...
23	Bar	Park	...
14	Hotel	Museum	...
26	Café	Fast Food Restaurant	...
29	Plaza	Sandwich Place	...
24	Sandwich Place	Restaurant	...
25	Café	Drugstore	...
54	Beach Bar	Diner	...
1	Café	French Restaurant	...
2	Bar	Hotel	...

3	Bar	Italian Restaurant	...
4	Hotel	Plaza	...
5	Coffee Shop	Park	...
6	Supermarket	Italian Restaurant	...
36	French Restaurant	Indonesian Restaurant	...
9	Park	Coffee Shop	...
12	Ice Cream Shop	Grocery Store	...
34	Supermarket	Gym	...
28	French Restaurant	Soccer Field	...
27	Supermarket	Coffee Shop	...
8	Restaurant	French Restaurant	...
0	Restaurant	Café	...

	16th Most Common Venue	17th Most Common Venue	18th Most Common Venue \
62	Tea Room	Gastropub	Drugstore
61	Mediterranean Restaurant	Rock Club	Chinese Restaurant
37	Cocktail Bar	Spa	Pub
40	Theater	Furniture / Home Store	Chinese Restaurant
22	Sandwich Place	Music Venue	Hotel
21	Diner	Fast Food Restaurant	Park
47	Gastropub	Tea Room	Beach Bar
32	Sushi Restaurant	Soccer Field	Music Venue
15	Drugstore	Record Shop	Concert Hall
48	Shopping Mall	Greek Restaurant	French Restaurant
13	Sushi Restaurant	Brewery	French Restaurant
16	Gym / Fitness Center	Breakfast Spot	Pool
33	Sandwich Place	Bistro	Swiss Restaurant
11	Grocery Store	Drugstore	Pub
60	French Restaurant	Forest	Movie Theater
42	Bus Stop	Furniture / Home Store	Music Venue
43	Pizza Place	Motorsports Shop	Bowling Alley
56	Cosmetics Shop	Department Store	Hotel
44	Gym / Fitness Center	Pool	Shopping Mall
49	Beer Garden	Sushi Restaurant	Concert Hall
46	Sushi Restaurant	Friterie	Diner
53	Furniture / Home Store	Discount Store	Sauna / Steam Room
52	Department Store	Golf Course	Café
45	Stadium	Soccer Stadium	Soccer Field
55	Discount Store	Clothing Store	Steakhouse
51	Bookstore	Bistro	Theater
41	Sports Bar	Chocolate Shop	Discount Store
58	Historic Site	Discount Store	Drugstore
39	Mexican Restaurant	Tapas Restaurant	Sushi Restaurant
38	Theater	Movie Theater	Furniture / Home Store
59	Discount Store	Seafood Restaurant	Electronics Store
35	Soccer Stadium	Spa	Sushi Restaurant
50	American Restaurant	Electronics Store	Chinese Restaurant

57	Bistro	Shopping Mall	Bakery
31	Chinese Restaurant	Castle	Resort
30	Pharmacy	Soccer Field	Pizza Place
7	Gym / Fitness Center	Park	Asian Restaurant
10	Sushi Restaurant	Movie Theater	Garden Center
17	Italian Restaurant	Gym / Fitness Center	Art Museum
18	Café	French Restaurant	Bus Stop
19	Clothing Store	Garden Center	Bistro
20	Chinese Restaurant	Friterie	Gym / Fitness Center
23	Big Box Store	Sports Club	French Restaurant
14	Seafood Restaurant	Bus Stop	Drugstore
26	Shopping Mall	Indian Restaurant	History Museum
29	Café	Fast Food Restaurant	Gym
24	Discount Store	Gastropub	Garden Center
25	Market	Resort	Department Store
54	Bus Stop	Toy / Game Store	Science Museum
1	Japanese Restaurant	Bagel Shop	Market
2	Gym	Thai Restaurant	Indonesian Restaurant
3	Café	Sandwich Place	Hotel
4	French Restaurant	Bakery	Breakfast Spot
5	Pub	Bookstore	Pool Hall
6	Bistro	Gastropub	Indian Restaurant
36	Bar	Department Store	Sushi Restaurant
9	Café	French Restaurant	Supermarket
12	Supermarket	Plaza	Asian Restaurant
34	Concert Hall	Shopping Mall	Bus Stop
28	Movie Theater	Diner	Farmers Market
27	Forest	Park	Hotel
8	Movie Theater	Soccer Stadium	Grocery Store
0	Yoga Studio	Pub	Food Truck

	19th Most Common Venue	20th Most Common Venue \
62	Seafood Restaurant	Snack Place
61	Beach Bar	Beach
37	Coffee Shop	Plaza
40	Bookstore	Snack Place
22	Coffee Shop	Chinese Restaurant
21	Coffee Shop	Church
47	Snack Place	Beer Bar
32	Multiplex	Gas Station
15	Discount Store	Chinese Restaurant
48	Fish Market	Soccer Field
13	Multiplex	Hotel
16	Playground	Chinese Restaurant
33	Hotel Bar	Beer Bar
11	Clothing Store	Bar
60	Bowling Alley	Grocery Store

42	Greek Restaurant	Snack Place
43	Middle Eastern Restaurant	Farmers Market
56	Fast Food Restaurant	Hockey Field
44	Bookstore	Breakfast Spot
49	Fried Chicken Joint	Food Truck
46	Mediterranean Restaurant	Forest
53	Café	Music Venue
52	Boat or Ferry	Bowling Alley
45	Flea Market	Food & Drink Shop
55	French Restaurant	Spanish Restaurant
51	Sandwich Place	Food Stand
41	Spa	History Museum
58	Gastropub	Beach
39	Buffet	Beer Bar
38	Diner	Discount Store
59	Mexican Restaurant	Shopping Mall
35	Furniture / Home Store	French Restaurant
50	Drugstore	Soccer Field
57	Gym / Fitness Center	Burger Joint
31	Pool	Café
30	Pool	Sandwich Place
7	Hockey Field	Spa
10	Café	Brewery
17	Beer Garden	Spa
18	Ice Cream Shop	Ski Area
19	Pool	Bookstore
20	Department Store	Historic Site
23	Bowling Alley	Pharmacy
14	Chocolate Shop	Mediterranean Restaurant
26	Rest Area	Record Shop
29	Museum	Park
24	Sushi Restaurant	Amphitheater
25	Forest	Playground
54	Café	Other Nightlife
1	Vietnamese Restaurant	Hostel
2	Plaza	Asian Restaurant
3	Farm	Monument / Landmark
4	Skate Park	Science Museum
5	Smoke Shop	Record Shop
6	French Restaurant	Friterie
36	Fish Market	Farm
9	Pool	Ice Cream Shop
12	French Restaurant	Furniture / Home Store
34	Restaurant	Recording Studio
28	Garden Center	Chinese Restaurant
27	Gym / Fitness Center	Gym
8	Café	Brewery

0	French Restaurant	Beer Bar
	21th Most Common Venue	22th Most Common Venue \
62	Sports Club	Bagel Shop
61	Shoe Store	Gastropub
37	Indonesian Restaurant	Department Store
40	Mexican Restaurant	Gastropub
22	Seafood Restaurant	Liquor Store
21	Indonesian Restaurant	Chocolate Shop
47	Skating Rink	Theater
32	Café	Sandwich Place
15	Plaza	Sporting Goods Shop
48	Bistro	Beer Store
13	Creperie	Diner
16	Cocktail Bar	Brazilian Restaurant
33	Gastropub	Gas Station
11	Snack Place	Brewery
60	Fast Food Restaurant	Road
42	French Restaurant	Sporting Goods Shop
43	Spanish Restaurant	Sporting Goods Shop
56	Sandwich Place	Canal Lock
44	Martial Arts School	Bistro
49	Sporting Goods Shop	Pool
46	Sandwich Place	Café
53	Resort	Chinese Restaurant
52	Soccer Field	Snack Place
45	Garden Center	Multiplex
55	Bistro	South American Restaurant
51	Mexican Restaurant	Food Court
41	Soccer Stadium	Bus Stop
58	Lighthouse	Tennis Court
39	Resort	Music Venue
38	Bookstore	Stadium
59	Flea Market	Soccer Stadium
35	Movie Theater	Shop & Service
50	Grocery Store	Clothing Store
57	Mexican Restaurant	Museum
31	Sandwich Place	Bus Stop
30	Church	Scottish Restaurant
7	Pool	Discount Store
10	Food Court	Beer Store
17	Bistro	Bookstore
18	Sandwich Place	Pub
19	Music Venue	Museum
20	Burger Joint	Bistro
23	Creperie	Pizza Place
14	Cheese Shop	French Restaurant

26	Japanese Restaurant	Deli / Bodega
29	Cafeteria	Middle Eastern Restaurant
24	Zoo	Theater
25	Food & Drink Shop	Snack Place
54	Chinese Restaurant	Department Store
1	Italian Restaurant	Shopping Plaza
2	Butcher	Bike Shop
3	Beer Store	Steakhouse
4	Skating Rink	Lounge
5	Chinese Restaurant	Brasserie
6	Greek Restaurant	Mediterranean Restaurant
36	Pool	College Gym
9	Gym / Fitness Center	Paper / Office Supplies Store
12	Music Venue	Museum
34	Pub	Bookstore
28	Beer Garden	Performing Arts Venue
27	Ice Cream Shop	Drugstore
8	Shopping Plaza	Breakfast Spot
0	Spiritual Center	Steakhouse

	23th Most Common Venue	24th Most Common Venue \
62	Nightclub	Bookstore
61	Snack Place	Boat or Ferry
37	Discount Store	Bed & Breakfast
40	Sandwich Place	Tapas Restaurant
22	Creperie	South American Restaurant
21	Climbing Gym	Record Shop
47	Drugstore	Bistro
32	Rock Club	Diner
15	Food Court	South American Restaurant
48	Beer Garden	Library
13	Liquor Store	Convenience Store
16	Café	Cafeteria
33	Big Box Store	Drugstore
11	Deli / Bodega	Market
60	Department Store	Park
42	Bar	Diner
43	Fast Food Restaurant	Sports Bar
56	Movie Theater	Sushi Restaurant
44	Fish Market	Fast Food Restaurant
49	Soccer Field	Monument / Landmark
46	Bowling Alley	Bus Stop
53	Library	Clothing Store
52	Skating Rink	Gym Pool
45	Sandwich Place	Cafeteria
55	Food Truck	Mediterranean Restaurant
51	Soccer Field	Miscellaneous Shop

41	Snack Place	Ramen Restaurant
58	Department Store	Event Space
39	Breakfast Spot	Bistro
38	Fried Chicken Joint	French Restaurant
59	Fast Food Restaurant	Nightclub
35	Garden Center	Museum
50	Juice Bar	Gym / Fitness Center
57	Campground	Surf Spot
31	Shopping Mall	Plaza
30	Skating Rink	Cheese Shop
7	Soccer Stadium	Soccer Field
10	Liquor Store	Chinese Restaurant
17	Big Box Store	Garden Center
18	Shoe Store	Pool
19	Movie Theater	Mexican Restaurant
20	Cosmetics Shop	Bookstore
23	Shopping Mall	Business Service
14	Market	Chinese Restaurant
26	Greek Restaurant	German Restaurant
29	Bookstore	Mexican Restaurant
24	Nightclub	Electronics Store
25	Construction & Landscaping	Exhibit
54	Bakery	Convenience Store
1	Bookstore	Food Truck
2	Wine Bar	Soccer Field
3	Cocktail Bar	Squash Court
4	Bookstore	Soccer Stadium
5	Sandwich Place	Concert Hall
6	Hardware Store	Skating Rink
36	Greek Restaurant	Gym
9	Chinese Restaurant	Sports Club
12	Gastropub	Snack Place
34	Snack Place	Indonesian Restaurant
28	Cheese Shop	Brewery
27	Sandwich Place	Market
8	Plaza	Snack Place
0	Beer Store	Soup Place

#### 25th Most Common Venue

62	Bakery
61	Theater
37	Museum
40	Music Venue
22	Snack Place
21	Scenic Lookout
47	Sandwich Place
32	Department Store



15	Liquor Store
48	Brasserie
13	Food Court
16	Bus Stop
33	Board Shop
11	Bowling Alley
60	Airport
42	Asian Restaurant
43	French Restaurant
56	Gas Station
44	Indonesian Restaurant
49	Clothing Store
46	Multiplex
53	Pub
52	Bus Station
45	Discount Store
55	Shopping Plaza
51	Café
41	Museum
58	Bar
39	Skating Rink
38	Sporting Goods Shop
59	Multiplex
35	Music Venue
50	Medical Center
57	Molecular Gastronomy Restaurant
31	Snack Place
30	Comfort Food Restaurant
7	Cultural Center
10	Dutch Restaurant
17	Frozen Yogurt Shop
18	Playground
19	Middle Eastern Restaurant
20	Concert Hall
23	Diner
14	Food Court
26	Diner
29	Burger Joint
24	Middle Eastern Restaurant
25	Factory
54	Gym
1	Sporting Goods Shop
2	Snack Place
3	Snack Place
4	Food Court
5	Comfort Food Restaurant
6	Hotel

```

36             Chocolate Shop
9             Movie Theater
12            Drugstore
34            Other Great Outdoors
28            Flower Shop
27            Mediterranean Restaurant
8             Bookstore
0             Indonesian Restaurant

```

```
[63 rows x 30 columns]
```

#### 4.1.1 Visualizing the clustered cities

Plotting the clusters

```

[52]: map_clusters_nl = folium.Map(location=[nederland_lat_coords,
↪nederland_lng_coords], zoom_start=7, tiles='cartodbpositron')

# set color scheme for the clusters
x = np.arange(k_num_clusters)
ys = [i + x + (i*x)**2 for i in range(k_num_clusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster, inwoners in zip(nl_data_nonan['Latitude'],
↪nl_data_nonan['Longitude'], nl_data_nonan['Stad'], nl_data_nonan['Cluster_
↪Labels'], nl_data_nonan['Inwoners']):
    label = folium.Popup('Cluster ' + str(int(cluster)) + '\n' + str(poi) ,
↪parse_html=True)
    radius_size = inwoners / 20
    folium.CircleMarker(
        [lat, lon],
        radius=radius_size,
        popup=label,
        color=rainbow[int(cluster-1)],
        fill=True,
        fill_color=rainbow[int(cluster-1)]
    ).add_to(map_clusters_nl)

map_clusters_nl

```

```
[52]: <folium.folium.Map at 0x25f97785370>
```

## 5 Discussion & Conclusion

## 5.1 Almere, a city of business opportunities?

Of the 10 largest cities, 9 belong to the Red Cluster (Amsterdam, Rotterdam, Den Haag, Utrecht, Eindhoven, Groningen, Tilburg, Breda, Nijmegen), indicating that, in general, large cities share a similar composition of venues. One exception to this is Almere, the eighth largest city (Green Cluster). By examining the data, this discrepancy can be attributed to the fact that Almere has significantly fewer tourist-focused venues, such as bars, coffee shops or hotels. Interestingly, it is Almere that has experienced one of the largest population increases over the last few years [1]. In addition, based on prognoses by CBS, Almere is expected to increase its population by a staggering 22.8% between 2018 and 2035, the largest of the top 10 cities [2].

Considering the lack of tourist-focused venues and the purported growth of Almere, it might provide promising business opportunities for entrepreneurs. However, more research is required regarding the expected future tourist influx and the venue related demands of the locals before any conclusions can be drawn.

## 5.2 Deventer as affordable alternative for large cities

By further observing the results, one can distinguish several smaller cities that exhibit large-city like characteristics (Red Cluster). This includes Hilversum, Deventer, Delft and Amstelveen. Deventer might be of particular interest for people who like larger cities, yet are unable to afford or unwilling to pay the high rent prices asked in these cities. While the average rent price per 100 m<sup>2</sup> for the 10 largest cities is €1560 (unweighted average), for Deventer this is only €1067.[3]

## 5.3 Small to medium-sized cities

The blue cluster contains only small cities with a population between ~50.000 and ~70.000. In this cluster, venues that are more focused on locals, such as supermarkets, drug stores and fitness centers, are prevalent.

The green and purple clusters can be considered as something in-between the blue and red cluster. Both contain small to medium-sized cities with a population between ~50.000 and ~200.000. The purple cluster seems to be slightly more similar to the large cities with most cities having restaurant as the first most common venue. For cities in the green cluster, this is mostly the supermarket category. However, it might very well be possible that the arbitrary radius of 3 kilometers has an influence on the restaurant/supermarket ratio and subsequently on the cluster arrangement.

## 5.4 Seaside towns share the same cluster

As expected, the seaside cities Westland and Katwijk share the same orange cluster since both have many beach related venues.

### 5.4.1 References

[1] P. Vissers, “Nederlandse steden worden drukker, slimmer, rijker - dus ook exclusiever” Trouw, 05-May-2019. [Online]. Available: <https://www.trouw.nl/nieuws/nederlandse-steden-worden-drukker-slimmer-rijker-dus-ook-exclusiever~b3b71cd6>. [Accessed: 03-Mar-2021]

[2] “Sterke groei in steden en randgemeenten verwacht” CBS, 10-September-2019. [Online]. Available: <https://www.cbs.nl/nl-nl/nieuws/2019/37/sterke-groei-in-steden-en-randgemeenten-verwacht>. [Accessed: 03-Mar-2021]

[3] “Dit betaal je gemiddeld aan maandhuur in 33 Nederlandse steden op de vrije markt” Business Insider Nederland, 21-Jan-2020. [Online]. Available: <https://www.businessinsider.nl/huur-prijs-maand-vrije-sector-steden-nederland-randstad/>. [Accessed: 03-Mar-2021] <https://www.businessinsider.nl/huur-prijs-maand-vrije-sector-steden-nederland-randstad/>