# MSc in Data Analytics

Author: Nascimento, Gustavo H.

e-mail: [2022313@student.cct.ie](mailto:2022313@student.cct.ie)

Student ID: 2022313

# ABSTRACT

*This report aims to explore and analyse a dataset related to the Public Cycle Stands located in the administrative area of Dublin City Council, whose data were provided by the Dublin City Council Transportation department in September 2022. Using Python and data analyse, it will be demonstrated the use of geolocation for the calculation of distance, between the points of the Bikes Stands and facilities, such as public transport stop points, hospitals and colleges. Clustering techniques will also be explored through machine learning, such as K-means, Affinity Propagation, and Self Organizing Maps (S.O.M.).*

# Introduction

The government of Ireland released the first National Cycle Policy Framework (N.C.P.F.) in April 2009. To ensure that a strong cycling culture is formed in Ireland so that by 2020, 10% of all travel in Ireland would be made by bicycle, it outlined 19 high-level objectives detailed in 109 policies.

Because this study is using a dataset related to the Public Cycle Stands located in the administrative area of Dublin City Council, whose data were provided by the Dublin City Council Transportation department in September 2022, the National Cycle Policy Framework, objectives 7 and 8 were the main objective considered. This objective is related to providing secure parking for bikes and discusses “The provision of well-located, plentiful, sheltered and secure parking facilities is as important to the cyclist as the provisions for moving cyclists described above. In recreating a cycling culture, it must be easy for cyclists to park their bikes as close as possible to their destination.”(N.C.P.F., 2009).

It is important to emphasize that this is an academic study and there is no objective of reporting these results to any authority or even of confronting the data with reality, so take into account that this study may not represent reality.

# About the base

The database chosen for this study was "*Public Cycle Parking Stands DCC*" which is available on Data.gov.ie. This base was released on 12/31/2015 and the last update of it was on 09/22/2022. There are 4 versions of this base available for download, the version chosen was the CSV one, since it had features in addition to the geographical coordinates. Another important point for choosing this database was the fact that it contains data from the coordinates "X", "Y", “Easting” and “Northing”, the latter two were fundamental in data loss, as the coordinates "X" and "Y" were missing data, as can be observed in the course of this project.

The raw DataFrame is composed of 937 rows (observations) and 7 columns (features). As the DataFrame doesn’t have a data dictionary available the features have been recognized by the data as they are described below:

1. “type\_stands”: Classify the stands type. There are 7 classifications ("Sheffield Stand", "Hoops", "Stainless Steel Curved", "Railing", "Temporary Racks", "Cargo bike stands" and "Sheffield Stand +Cargo Stand");
2. “X”: Longitude;
3. “Y”: Latitude;
4. “Easting”: Longitude;
5. “Northing”: Latitude;
6. “location\_stand”: describe the location;
7. “no\_stands”: the number of spaces to store bikes in each (bikes supported).

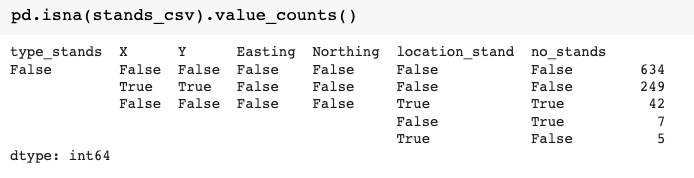
An initial analysis was possible to identify that the data has missing values in 4 columns (Figure 1).

Texto preto sobre fundo branco

Descrição gerada automaticamente

Figure 1: stand\_csv DataFrame information

The function *isna* (Figure 2) showed us that there are missing values in the columns ‘X’ and ‘Y’, these columns are coordinates necessary to create a GeoDataFrame. That same function showed us that in the coordinates ‘Easting’ and ‘Northing’, also coordinates had values in all observations.



**Figure** 2: Counting is not a value

Because the other features are not relevant for this moment of the project, observations without values defined in the features were not treated, since when was needed analyses where all observations need to be filled, as, in the case of Machine Learning, a new data frame was created with this information removed.

# Geospatial data

“Geospatial data is information that describes objects, events or other features with a location on or near the surface of the earth”( IBM, n.d.) Such information can be divided into Data layers, as described in Figure 3, the hierarchy of this division depends on the library/source being used, however, it is possible to work with one or more layers. Each layer brings information (geo data) that also might be used to store other objects individually as well as their topology, or how they relate to one another. In this study, layer 2 (buildings data) objects were used, however, filtered, as explained later.

Diagrama

Descrição gerada automaticamente

Figure 3: Geospatial data capture essential geographical attributes of the location (Wikipedia - Visual representation of themes in a GIS, n.d.)

To perform the necessary operations the raw GPS data (in that case, coordinates) needs to be converted into spatial data, which can be recognized by GeoPandas. For this to be done, the latitude and longitude points must be provided to the function *geopandas*.*points\_from\_xy*. Because the Easting and Northing features have all the values, those were the features used to create a new feature called “geometry” and as a Pandas DataFrame with the “geometry” feature is called a GeoDataFrame the bike stands DataFrame was converted to a GeoDataFrame.

Because latitude and longitude values alone are not sufficient for the point to be recognized (or plotted) on the globe. it is necessary to inform the CRS (Coordinate Reference System), There is a wide variety of CRS, and each one is designed for particular objectives in various parts of the world. As the values in the chosen columns were from the Irish government and there is no data dictionary, the official Irish CRS was used to transform the coordinates into the geometry feature (Figure 4).

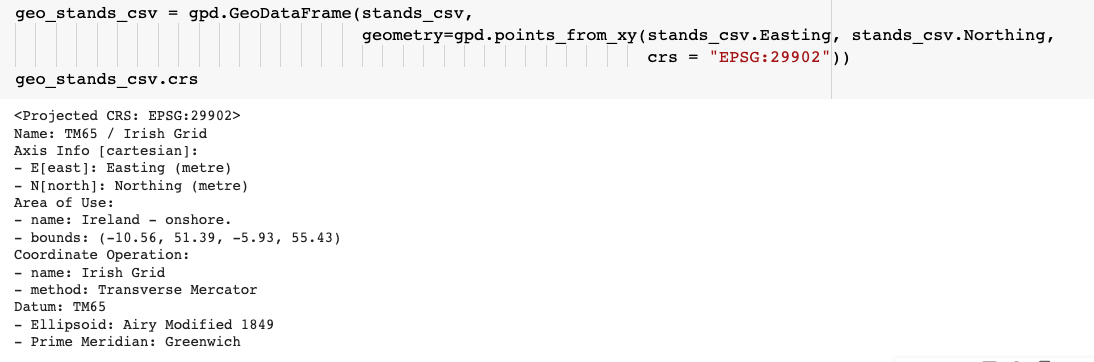


Figure 4: Print the CRS properties of stands\_csv GeoDataFrame

Once the CRS was set in the conversion of latitude and longitude, it was necessary to set the Folium and OSMnx standard CRS (EPSG:4326), so that the bike stands coordinates could be checked if they were within the expected coordinates, that is, within the administrative area of the DCC. Before that, the generated coordinates were used to see if there is more than one stand at the same point since we assumed that there were no two stands at the same spatial point. This operation resulted in 6 repeated observations, and 3 observations needed to be removed from the GeoDataFrame to be worked on, which the info is shown in Figure 5.

Interface gráfica do usuário, Texto, Aplicativo, Tabela

Descrição gerada automaticamente com confiança média

Figure 5: stand\_check GeoDataFrame information

The bike stand GeoDataFrame was plotted using Folium and this showed us the existence of points outside Dublin coordinates (Figure 6)

Mapa

Descrição gerada automaticamente

Figure 6: plotting the coordinates of the stands on the entire world map

To check which points are outside the administrative region of Dublin, a database made available by the DCC on the Smartdubin.com website was used, on this database, the geolocations of the 5 administrative areas of Dublin are found (Figure 7).

Uma imagem contendo Gráfico

Descrição gerada automaticamente

Figure 7: Plotting of Dublin administrative units

To make sure that the area coordinates were correct it was plotted using Folium and the image below was obtained as result (Figure 8). Therefore, the coordinates of the DCC area were visually confirmed.

Mapa

Descrição gerada automaticamente

Figure 8: Plotting of Dublin administrative units on Folium map

Consequently, the points of the stands were checked using the function "*gpd.intersects*". This function inserts a column into the GeoDataFrame with Boolean values (true or false) indicating whether the points are within the checked area. During this process, it was necessary to use the "*unary\_union*" property to simplify the information contained in the feature geometry of the parsed GeoDataFrame. As result was possible to identify 3 points out of the DCC area (Figure 9).

Interface gráfica do usuário, Aplicativo, Mapa

Descrição gerada automaticamente

Figure 9: Plotting the cycle stands point detected outside the DCC.

To confirm (visually) that the other 931 stands detected as within the administrative area of Dublin are in the expected location, they were plotted with the colour green to differentiate from those outside DCC (in red colour), it could be observed that there is no point green outside the DCC area. (Figure 10). For the next analyses was used only the points that were within the area of the DCC, which were 931 stands (observations). This represented a loss of 0.64% of the initial data.

Mapa com desenhos

Descrição gerada automaticamente com confiança média

Figure 10: All cycle stand points plotted on Folium map - Right zoomed in on Dublin | Left on the map worldwide.

# Bike Stands GeoDataFrame Features.

As previously mentioned the database has a feature that indicates the number of spaces available to store cycles in each stand (observation), that is, their capacity. Using this feature it was verified that there are 5226 spaces for cycles in Dublin, a mean of 5.92 per stand. It was also found that there is a stand with a much larger capacity than the other, this stand can store 166 bikes and is located in Houston station. Due to the disproportion between the Houston station's stand and the other bike stands, it was considered an outlier and for an analysis of the Distribution of the number of spaces to store bikes per stand has been removed, in this way, the Distribution can become more visible (Figure 11). However, the Houston station's stand was not removed in the others following analysis.

Gráfico, Histograma

Descrição gerada automaticamente

Figure 11: Indicators of the number of spaces for bikes in the stands - left without the outlier | right with the outlier

Another feature that could have been further explored is the feature "type\_stands', however, due to the scope of the project it was not used, so it is not represented graphically or statistically in this project, however, for future projects to analyse their differences and distribution may be interesting.

# OSMnx – Public Transport

Objective 8 of the National Cycle Policy Framework defines 10 policies to ensure Proper Integration Between Cycling And Public Transport (PT), in particular policy 8.2 which has defined that there should be cycling parking in all stations. Taking this into consideration, the OSMnx library was used to get data from all public transportation points within the DCC. The query result in a database with 745 rows and 45 columns, since all the features resulted were not needed, only the columns with the point’s name and the geometry were left. Figure 12 shows the Bike Stations (green) and public transport stops (yellow) plotted in the DCC area.

Mapa

Descrição gerada automaticamente

Figure 12: Plotting of public transport stop points and bike stands points in the background DCC area.

In sequence, two functions created by Vallejo, B and made available in their article "*Nearest Neighbor Analysis for Large Datasets in Helsinki Region*" were used to bring the distance between the bike stand and the nearest bus point. The first function was responsible for separating coordinates from points and adding them as radians to a NumPy array, so the second one was called and was responsible for creating a tree using the “*sklearn.neighbors.BallTree*” function and then getting the closest points and their distances, that information was returned to the first function which converts the distance into radians and returns that data to a new GeoDataFrame created and called “*closest\_stops*”.

The “*closest\_stops*” GeoDataFrame had the same number of rows as the bike stands GeoDataFrame and the same features as the PT stop GeoDataFrame, that happened because which row represents the closest PT stop for the stands in the same sequence, i.e. with the same index. This allowed us to merge the new GeoDataFrame with the GeoDataFrame of the bike stands. In this way, each bike stand had a PT stop point and the distance between them.

Once the distances between the nearest PT Stop and the Bike Stands were inserted into the same new GeoDataFrame called “*stands\_dcc\_stops*”, it was possible to plot the information using a colour scale to identify the size of the distance between the objects. Figure 13 shows the "Viridis" colour scale the Bike Stands and the red dots are the PT stop points.

Mapa

Descrição gerada automaticamente

Figure 13: Plotting the points of bike stands with a colour pattern according to the distance of the PT stop point, and all PT stop points in red.

The next step is to create a line (a link) symbolizing the distance between the bike stand and the nearest PT stop. For this, a new feature in the GeoDataFrame “*stands\_dcc\_stops*” was created called “*link*”, using the function “*shapely.geometry*.*LineString*”. With the links created and the other points identified, it was possible to plot the graph of Figure 14, where you can see which PT Stop each Bike stand is closest (in the "Viridis" colour scale the Bike Stands and the x are the closest PT stop points).

Mapa

Descrição gerada automaticamente

Figure 14:Plotting the points of bike stands with a colour pattern according to the distance of the PT stop point, and the nearest PT stop points connected by a line.

Using the “distance” feature generated in this process, it was possible to verify that the mean distance between a bike stand and the closest PT stop is 154.9843 meters. As shown in Figure 15, 10% of stands are up to 33.6428 meters away from a public transport stop point, and 1% are up to 11.7437 meters away.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Figure 15: shows the percentile of the distance between a Bike Stand and a PT stop.

Then, Empirical Cumulative Distribution Function (ECDF) were made by the “*sns.ecdfplot*” method. This function orders each unique observation and computes as the number of observations is less than or equal to the given observation divided by the total number of observations. It has demonstrated that only 34.5865% (axis Y – Proportion) of the bike stands are located less than 100 meters axis X – Distance) from the nearest PT stop, while 59.0763% are located less than 200 meters from the nearest PT stop. As shown in Figure 16.

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figure 16: Empirical cumulative distribution of the bike stands by distance – Right stands less than 100 meters from the PT stops | Left stands less than 200 meters from the PT stops.

As can be seen in Figure 14 there are PT stop points which have more than one bike stand linked to them, using the geographical coordinates of the closest stop points as indices it was observed that, on the mean, each stop point is related to 3.2214 bike stands. And there are PT stop points with a high number of bike stands related to them. For example, a PT stop is the closest stop point to 57 bike stands. Another information obtained by checking the coordinates of the PT stops was that only a total of 289 PT stops (38.7919% of the total) were considered the closest to a Bike Stand. From this point, the rationing of this project was reversed, and the study comes to consider that instead of looking for what is the closest public transport stop point to a Bike stand, it was verified which Bike Stand point is closest to the public transport stop point. In addition to this, was began to consider not only the public transport stop points but all the facilities points made available by the Open Street Project through the OSMnx library.

# Facilities

It is possible to explore the OSMnx library by doing searches using TAGs, which allowed to locate establishments (buildings) mapped in the Open Street Project, for classification purposes these establishments were separated into 4 groups, as listed below:

1. Sustenance: in this group are establishments classified as *'pub', 'bar', 'restaurant', 'cafe', 'Biergarten', 'fast\_food', 'food\_court'* and *'ice\_cream'*;
2. Education: in this group are establishments classified as *'college', 'university', 'school', 'library', 'driving\_school', 'kindergarten', 'language\_school', 'toy\_library', 'training'* and *'music\_school'*;
3. Healthcare: in this group are establishments classified as *'pharmacy', 'hospital', 'nursing\_home', 'baby\_hatch', 'clinic', 'dentist', 'doctors', 'social\_facility'* and *'veterinary'*;
4. Public Service and other facilities: in this group are establishments classified as *'place\_of\_worship',' post\_office', 'police'* and *'bank'*.

For each base a search was performed by the establishments of these groups, only the features that inform the category, the name and the geographical position were maintained, all the other features were dropped. After the query, the bases were unified to a single GeoDataFrame called facilities. The GeoDataFrame "*facilities*" took a total of 3908 rows (observations) and 4 columns (features): name, category, group and geometry. The feature name had 215 NaN values, however, this information was kept for information only and was not necessary for future analysis. To this GeoDataFrame the stops GeoDataFrame was added, and that resulted in a GeoDataFrame with 4653 rows, the NaN values in the Name column have increased for 277 rows.

Before going ahead was necessary to check if de coordinates were within the DCC area, to do that, was used the function “*GeoDataFrame.clip()*” and this operation resulted in a GeoDataFrame with 4639 rows, 14 lines less than the initial GeoDataFrame, the observations were plotted in the DCC area, segmented by groups, and displayed in Figure 17.

Uma imagem contendo Mapa

Descrição gerada automaticamente

Figure 17: Facilities points plotted in DCC.

The next step was to locate the closest Bike Stand for each facility, using the same process described above. This step resulted in all Bike Stands features in the facilities GeoDataFrame, now called "*facilities\_dcc\_stand*" in that case, as before, the inserted observations were duplicated as the relation is 1 to N (one stand Bike has more the one facilities close to, however, one facility has just one close Bike Stand to it, in this case, the closest Bike Stand was chosen), though, the observation distance was added, that observation was calculated for each facility point using the distance between the facility and the nearest bike stand as a straight line.

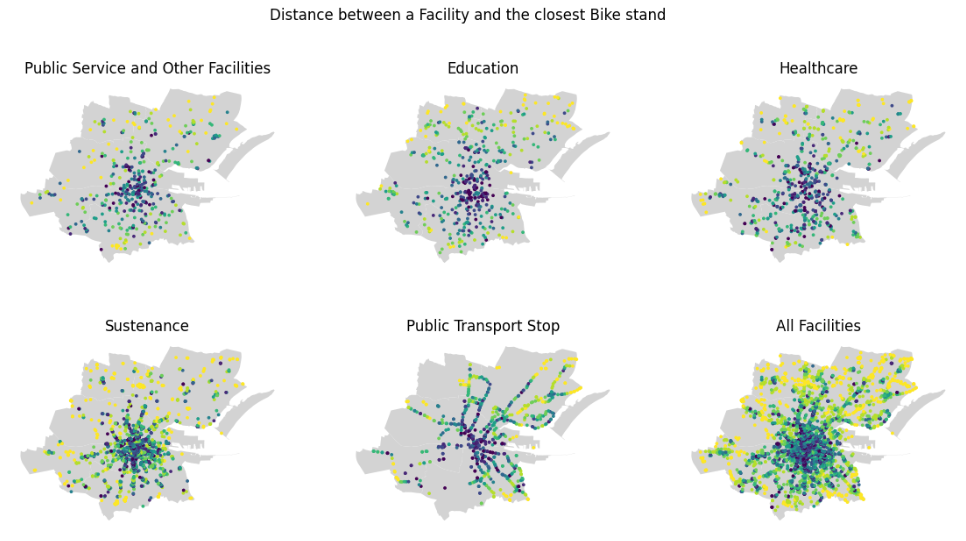


Figure 18: Facilities points (by groups) plotted in DCC are, coloured according to the distance between the point and the closest Bike Stand.

The GeoDataFrame "*facilities\_dcc\_stand*" was used for the following analyses since it has the data that interests us most.

The "*facilities\_dcc\_stand*" has a sampling of 4639 facilities, as stated earlier these facilities are divided into 5 groups and in turn, each group has specific categories, which would allow a more detailed analysis for each category, however, this more detailed analysis is recommended for future studies.

According to the distance analysis, it is verified that the average distance between a facility and the nearest bike stand, in a straight line, is 161.2926 meters, the maximum distance is 1693.1715 meters, the minimum is 0.8098 meters, and the median distance is 70.9939 meters.

As observed in the histogram (Figure 18), the parameters have very different intervals and this skewed to the right, i.e., that could have a bias, one of the extremities is elevated and has a long tail, so measures such as correlation or regression may be influenced by the peak of the distribution or outliers. Because this may have affected the distribution, the distance feature was transformed to log on base 10, and a much more normal distribution was generated, shown in figure 19.

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figure 19: Histogram of the Distance between a facility and the closet bike stand. Right: X with distance no transformed | Left: X with distance transformed in LOG10

# Normal Distribution (PDF)

In addition to understanding the distribution of the facilities concerning the bike stands, an Probability Density Function (which shows the expected probability for observing a value) was made, and again this has shown us that the data has a bias or outliers, in this way using the data transformed in LOG 10, was possible to identify a Normal Distribution. (Figure 20)

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figure 20: Normal Distribution of the Distance between a facility and the closet bike stand. Right: X with distance no transformed | Left: X with distance transformed in LOG10

# Empirical Cumulative Distribution Function (DCF)

Another Empirical Cumulative Distribution Function (ECDF) were made, and as a result, it was easier to identify where a proportion was located, for example, it is possible to see that 59.7111% (axis Y – Proportion) of the facilities are less than 100 meters (axis X – Distance) from the closest Bike Stand, as shown in Figure 21.

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figure 21: ECDF of the Distance between a facility and the closet bike stand.

Since the National Cycle Policy Framework is not exactly about the distance for each type of facility, making a mention that the bike stands should be close, for academic purposes was assumed as “close” bike stands up to 100 meters of distance for the next analyses.

# Bernoulli

A random variable that has only two possible values, often 0 and 1, is known as a Bernoulli random variable. This random variable model simulates random experiments with two possible results, referred to as "success" and "failure" in some instances. For this study, it was considered a “success” (1) the possibility of the facility being within 100 meters of a bike stand.

Gráfico, Histograma

Descrição gerada automaticamente

Figure 22: Bernoulli variable (1: success, 0: failure)

The graph in figure 22 was plotted using the “*bernoulli.pmf”* method of library “*scipy.stats*”, with a sampling of 4639 facilities which had the distance converted to 1 in case it was less than 100 and if it was not that was 0, the probability of 59.7111% of a facility having a bike stand less than 100 meters was set up.

# Binomial Distribution

The following test was performed as experimentation. In the chance of a person using 5 facilities, they will have a 32.2443% probability of finding in at least 2 facilities bikes stand up to 100 meters away. For the analysis of this hypothesis, the "*binom.cdf*" function was also used from the "*spicy.stats*" library, as a probability, the frequency that the bike stands were at least 100 meters from one of the facilities analysed was used as success probability.

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figure 23: Plotting of binom.cdf() function checking the probability of achieving a number of successes within a range of 5.

# Machine Learning - Methodology

Since 1989, the Knowledge Discovery in Database (KDD) term has stood for the entire process of gathering data and methodically enhancing it. This was the method chosen for this study since its stages are:

1. Selection and Addiction: The targeted data is identified using a database of assembled data, and the factors that will be assessed for knowledge discovery are identified.
2. Pre-processing: This stage covers the idea of data cleaning and is focused on foreseeing the data being used. In order to forecast similarly flawed, missing, and attributional mismatched data and subsequently work it out of future processes, predictive models for unreliable data are built.
3. Transformation: This stage focuses on transforming the pre-processed data into a form that can be used completely. This is accomplished by reducing the variability of the scope, and the data properties are securely set for future review. Here, the data is arranged and sorted, frequently into one type.
4. Data Mining: The data mining stage of the process, which is the most well-known, focuses on sorting through the processed data to look for interesting patterns. These patterns are graphed, trended, and charted in a way that is most beneficial to the KDD being done for the process. Regression, clustering, and grouping are all methods used in this phase; which one (or more) is used depends on the results that are anticipated and desired.
5. Evaluation and Interpretation: The data are handed over for interpretation and documentation in the final stage. To enable people more effectively assess the curated output, as the reports are generated. The data has now been cleaned, transformed, dissected based on pertinent properties, and framed into visual representations.

This project began to be done with databases already available and public, these databases did not suffer updates from their sources during the study, due to the lack of available dictionaries it was necessary for an interactive process, new choices of bases and a change of rationale during the project. In this way, the KDD data mining process fits better with the routine developed.

# Supervised or Unsupervised machine

Supervised and unsupervised machine learning models are the two main categories of machine learning, there is also a semi-supervised technique, which is an intermediary between supervised and unsupervised. In supervised learning, the model determines a map from the input data to the output data. However, you only have input data for unsupervised learning. Unsupervised learning has a variety of objectives, including lowering the dimensionality, detecting anomalies, and clustering observations.

As described previously, this project worked with a database using geolocation data, and in general, these data were the main used to do the analyses, once exposed to a small number of existing features to work is important to know that the Latitudes and Longitude features generated through the geolocation point and have no correlation since they serve to describe a specific point in a Cartesian plane.

# k-means clustering

The process of grouping population or data points into various categories is called clustering. Data points differ from those in other categories and are more similar to those in the same category. Basically, it's a grouping of things depending on how similar and unlike they are.

Because of that, in this study, the first technique tested was k-means clustering, which is an unsupervised machine-learning technique since we are working with unlabelled datasets, it is necessary to separate the data into groups based on their similarities.

The bike stand geolocation data (geometry) was utilised to execute the K-means clustering method to obtain many clusters, each of which has members whose data are most similar to one another.

Using the KElbowVisualizer method was used to indicate the number of clusters as shown in Figure 24. “The KElbowVisualizer implements the “elbow” method to help data scientists select the optimal number of clusters by fitting the model with a range of values for K.” (Yellowbrick, n.d.)

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figure 24: KElbowVisualizer method showing the distortion score.

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figure 25: KElbowVisualizer method showing the silhouette score.

As can be seen in Figure 24, it was indicated the use of 5 clusters, however, the silhouette score( Figure 25) was shown a negative relationship between the increase of clusters and the fall of the silhouette score, and because of this, was used 4 clusters also, so the model was fit using 4 and 5 clusters.

Gráfico

Descrição gerada automaticamente com confiança média

Figure 26: Right: K-means (k = 5) | Left: Silhouette K-means (k = 5)

Gráfico

Descrição gerada automaticamente

Figure 27 K-means (k = 4) | Left: Silhouette K-means (k = 4)

# Affinity Propagation

Affinity Propagation is used to identify the closest exemplar for each observation using a graph-based approach. In essence, every observation casts a "vote" for the other observations they wish to be grouped with, which causes the entire dataset to be divided into a huge number of uneven clusters.

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Figure 28: Affinity Propagation

Affinity Propagation did not fit the model very well, this resulted in 2969 clusters estimated. It was not possible to do further analysis, since it was not possible to put the cluster number (since this is part of the ML analysis) and the time of processing was long. This type of clustering was considered inappropriate for the data the way it was.

# Self Organizing Maps

A SOM (also known as Kohonen layer) is a specific kind of artificial neural network that learns through unsupervised learning to create a "map" or low-dimensional representation of the input space.

With the purpose of comparing with K-means, an array (1, 4) was placed, resulting in 4 clusters, and an array of (1, 5) resulted in 5 clusters.

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Figure 29: SOM (1, 5) plotter using coordinates

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Figure 30: SOM (1, 4) plotter using coordinates

Both the Self Organizing Map (SOM) and k-means showed good clustering results, however, it seemed to me that the SOM division is more symmetrical than taking into account the population of the region, which can be seen in figure 30, in the northern part of Dublin [6.2, 53.4]. Although it is necessary to run more tests to evaluate which Algorithm fit the best model evaluated, with the tests performed so far the K-means with 4 clusters was the best to perform the clustering of this model.

# Python libraries

The libraries needed for analysing were:

1. Geopandas: “GeoPandas is an open-source project to make working with geospatial data in python easier. GeoPandas extends the datatypes used by pandas to allow spatial operations on geometric types.” (“Training Resources for Geospatial Computing” - GeoPandas documentation);
2. Shapely: Shapely Python is a package for computational geometry, as Geopandas performs geometric operations by shapely this library is required;
3. OSMnx: “OSMnx is a Python package that lets you download geospatial data from OpenStreetMap and model, project, visualize, and analyze real-world street networks and any other geospatial geometries.” (“OSMnx 1.2.2 — OSMnx 1.2.2 documentation - Read the Docs” - OSMnx documentation).
4. Pandas: “Panda is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.” (“pandas documentation — pandas 1.5.1 documentation” - Pandas documentation);
5. Matplotlib: Matplotlib is a library used for data visualization and graphical plotting in python;
6. NumPy: “NumPy is an open-source project aiming to enable numerical computing with Python.” (“NumPy - About Us” – NumPy documentation);
7. Folium: “Folium makes it easy to visualize data that’s been manipulated in Python on an interactive leaflet map.” (“Folium – Astraea Support Center” – Folium documentation).

# References

National Transport Authority (2009). Ireland's First National Cycle Policy Framework - HSE [online] Available at: https://www.hse.ie/eng/about/who/healthwellbeing/our-priority-programmes/heal/healpublications/national-cycle-policy-framework.pdf

Vallejo, Bryan R.(2021). Nearest Neighbor Analysis for Large Datasets in Helsinki Region. [online] Towards Data Science Available at: https://towardsdatascience.com/nearest-neighbor-analysis-for-large-datasets-in-helsinki-79ebc7288c91

Holdgraf, C. and Wasser, L. (2020).Lesson 2. Customize Map Legends and Colors in Python using Matplotlib: GIS in Python. [online] Earth Data Science.org available at: https://www.earthdatascience.org/courses/scientists-guide-to-plotting-data-in-python/plot-spatial-data/customize-vector-plots/python-customize-map-legends-geopandas/

Brownlee, J. (2019) How to Use an Empirical Distribution Function in Python [online] Machinelearningmastery.com Available at: https://machinelearningmastery.com/empirical-distribution-function-in-python/

Doshi, K. (2021) Leveraging Geolocation Data for Machine Learning: Essential Techniques [online] towardsdatascience.com available at: https://towardsdatascience.com/leveraging-geolocation-data-for-machine-learning-essential-techniques-192ce3a969bc

Kurniawan, T (2021) The Clustering Algorithm with Geolocation data [online] Medium.com available at: https://medium.com/thelorry-product-tech-data/the-clustering-algorithm-with-geolocation-data-d6dd07ed36a

Pietro M. (2020) Clustering Geospatial Data [online] towardsdatascience.com available at: https://towardsdatascience.com/clustering-geospatial-data-f0584f0b04ec

IBM (n.d) Geospatial data definition [online] IBM.com available at: <https://www.ibm.com/topics/geospatial-data>

Datacamp (2019) Probability Distributions in Python Tutorial [online] Datacamp.com available at: https://www.datacamp.com/tutorial/probability-distributions-python

Gillies, Sean (2022). The Shapely User Manual [online] Shapely Documentation. Available at: https://shapely.readthedocs.io/en/stable/manual.html

Boeing, G. (2017). OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks. Computers, Environment and Urban Systems 65, 126-139.[online] OSMnx Documentation. Available at: https://osmnx.readthedocs.io/en/stable/index.html

Wiki.openstreetmap.org (n.d.) Category: Tag descriptions. [online] wiki.openstreetmap.org. Available at: https://wiki.openstreetmap.org/wiki/Category:Tag\_descriptions

pandas.pydata.org. (n.d.). pandas documentation — pandas 1.0.3 documentation. [online] Available at: https://pandas.pydata.org/docs/index.html.

seaborn (2012). seaborn: statistical data visualization — seaborn 0.9.0 documentation. [online] Pydata.org. Available at: https://seaborn.pydata.org/.

numpy.org (n.d.). Numpy documentation [online] Numpy.org. Available at: https://numpy.org/about/

folium (n.d.). Folium 0.12.1 documentation. [online] folium. Available at: https://python-visualization.github.io/folium/#

geopandas (n.d.) GeoPandas 0.12.1 documentation [online] Geopandas.org Available at: https://geopandas.org/en/stable/index.html

Matiplot (n.d.) Matplotlib 3.6.2 documentation [online] Available at: https://matplotlib.org/stable/index.html

Yellowbrick (n.d.) Elbow Method documentation [online] scikit-yb.org. Available at: https://www.scikit-yb.org/en/latest/api/cluster/elbow.html

Marsh, Charles(n.d.) Computational Geometry in Python: From Theory to Application [online] Toptal.com. available at: https://www.toptal.com/python/computational-geometry-in-python-from-theory-to-implementation