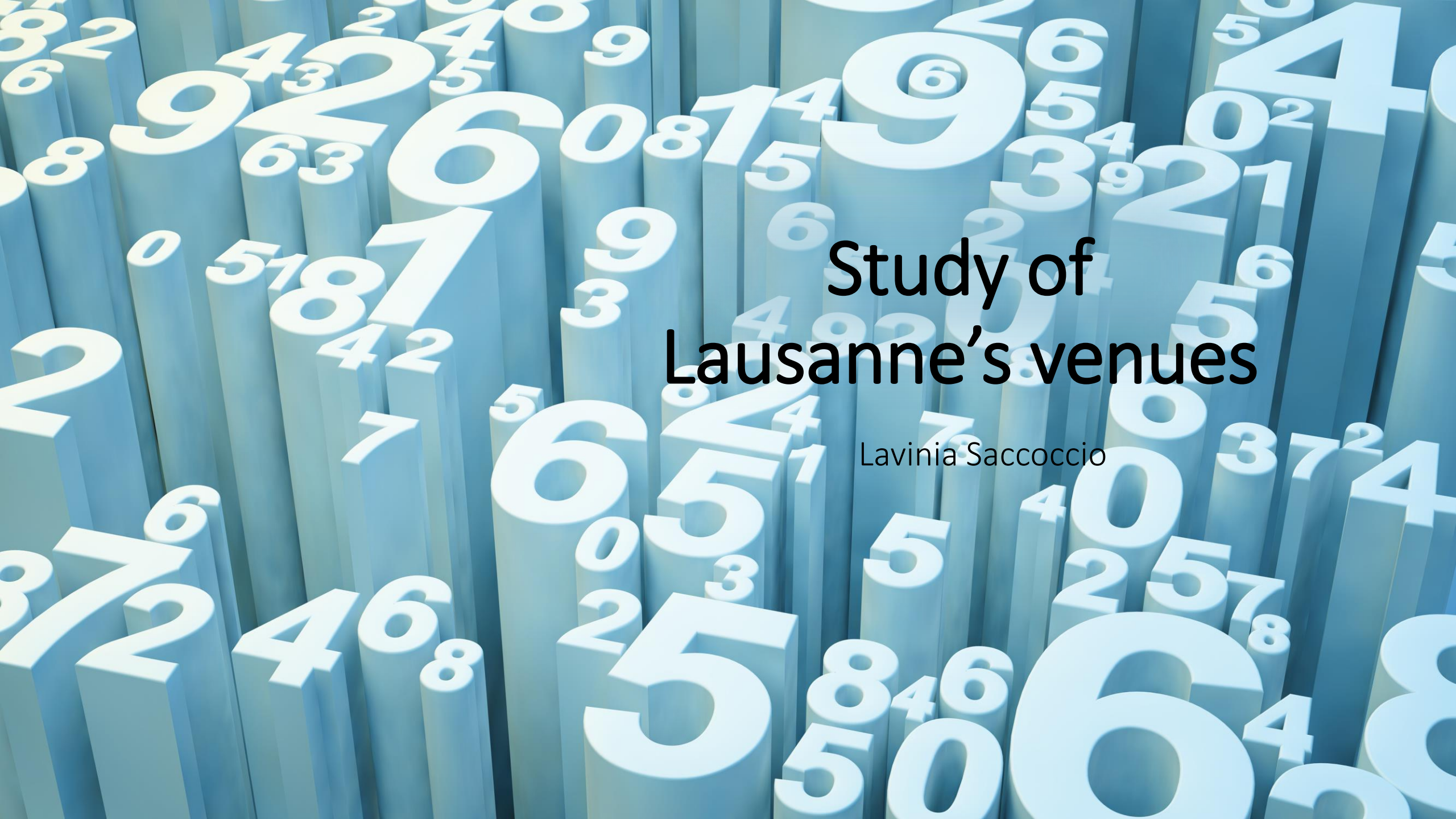
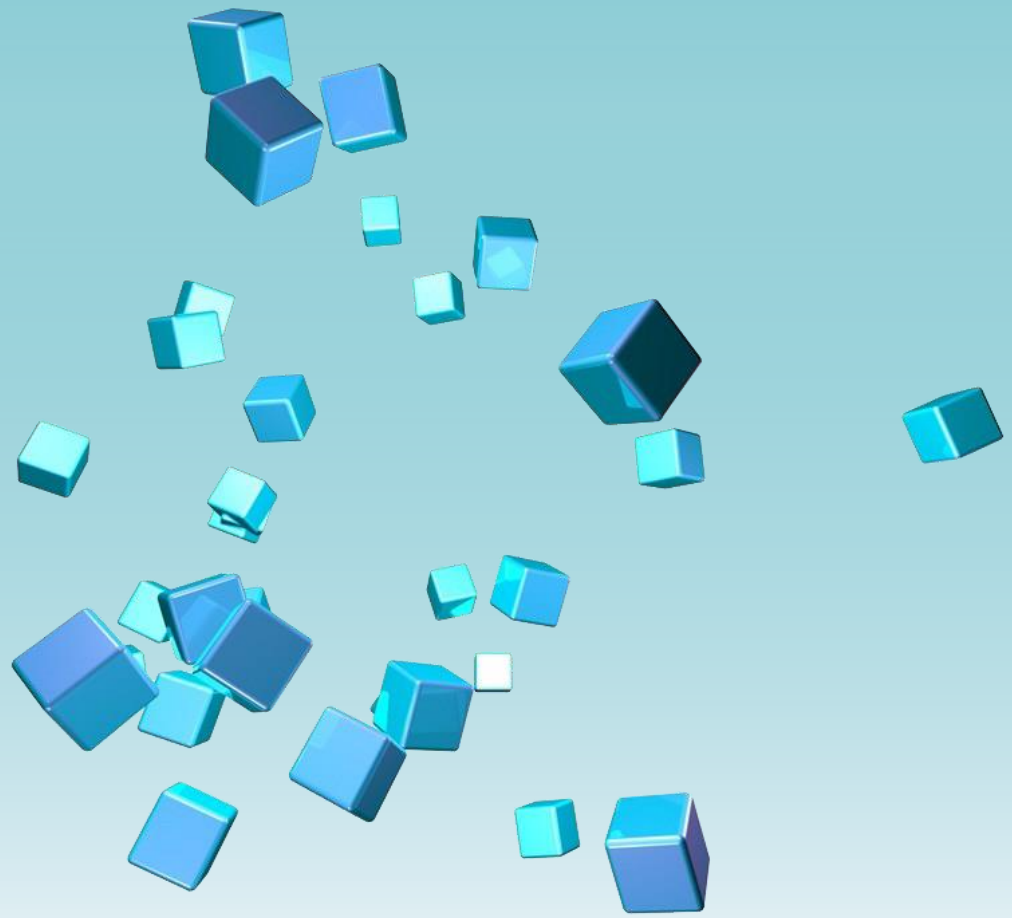


Study of Lausanne's venues

Lavinia Saccoccio

Overview

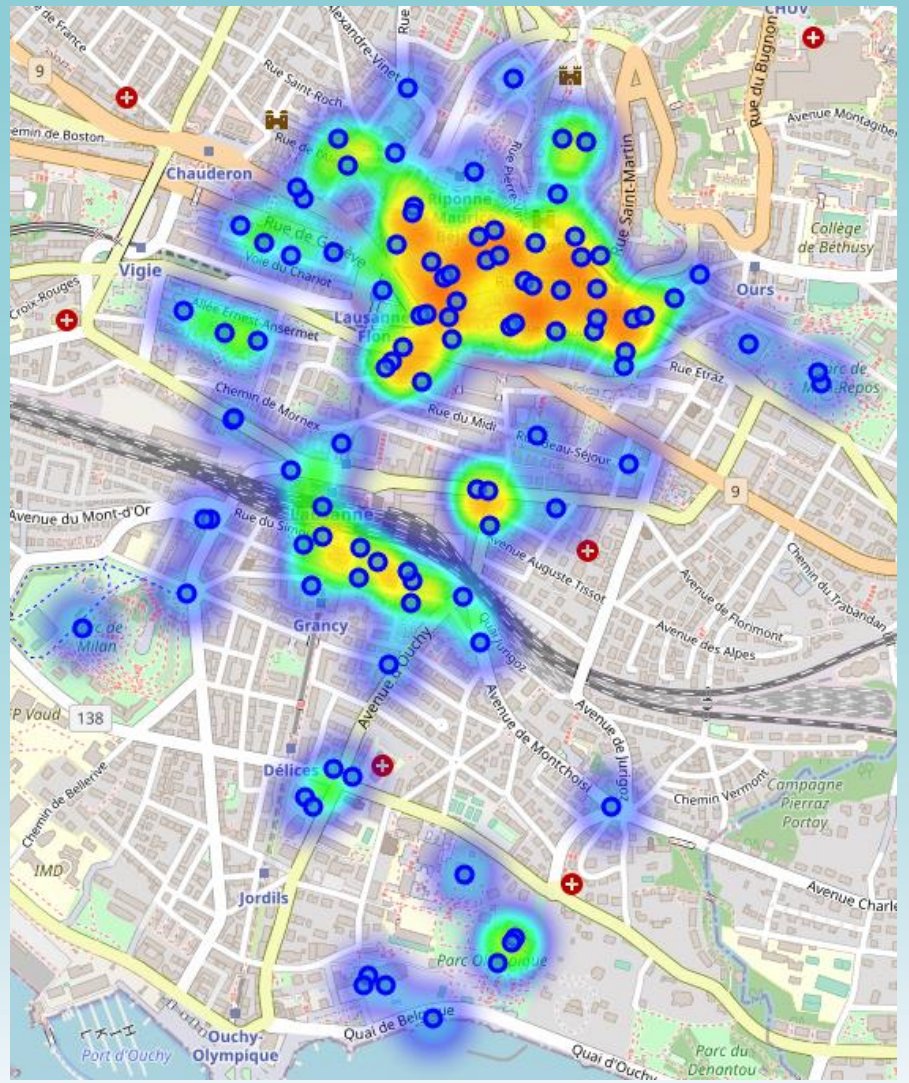
- Introduction
- Libraries used in the project
- How does k-Means work
- Algorithm :
  - Pre-processing and cleaning data
  - One hot encoding
  - Application of k-Means
- Conclusions



Introduction

The idea is to analyse the city of Lausanne and create clusters of venues that are similar to each other.

First we scrape the venues and get the coordinates, then we clean and analyse the data.



Main libraries used

- **Json**: to handle json files from scraping stage
- **Pandas**: library for data analysis
- **K-means** from **Scikit Learn**: for clustering stage
- **NumPy**: library to handle data in a vectorized manner
- **Folium**: map rendering library
- **Matplotlib**: for plots
- **Requests**: library to pull requests from the API

How does k-Means work

K-Means aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean (cluster *centroid*)

- K has to be initialised
- Each data point is assigned to its nearest centroid, based on the squared Euclidean distance (based on feature similarity)
- Used for anomaly detection, image compression, Segmentation

The Algorithm: Foursquare API

We will send GET requests to the Foursquare API in order to get the venues data for the city of Lausanne:

- Build the URL, send the GET request and write the source code into a json file
- Repeat the scraping process to extract data from the json file: venue name, category, latitude and longitude
- Put everything into a new dataframe

The Algorithm: Pre-processing and cleaning

An important task in every project is to make sure data is clean and ready to be analysed.

- Filter the dataframe and reset the indices (removes the missing values)
- Dropped the column with the venue **name**
- Check and drop any row with a **NaN** value

The Algorithm: one hot encoding

In order to run K-Means, we have to transform the dataframe into something that the machine could understand.

	name	Art Museum	Bar	Bistro	Bookstore	Breakfast Spot	Burger Joint	Café	Candy Store	Chinese Restaurant	Church	Coffee Shop	Creperie	Cupcake Shop	Department Store	Dessert Shop	Diner	Falafel Restaurant	Fast Food Restaurant	French Restaurant
0	Sleepy Bear Coffee	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	Les Trois Rois	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	Pasta e Sfizi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Café du Simplon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Les Gosses du Québec	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- One hot encoding is a process by which categorical variables are converted into binary variables (0,1)

The Algorithm: k-Means

After all the preparations we are ready to run **k-Means**:

```
[94]: # set number of clusters
      kclusters = 7

      lausanne_onehot_clustering = lausanne_onehot.drop('name', 1)

      # run k-means clustering
      kmeans = KMeans(n_clusters = kclusters, random_state = 0).fit(lausanne_onehot_clustering)

      # check cluster labels generated for each row in the dataframe
      kmeans.labels_[0:10]
```

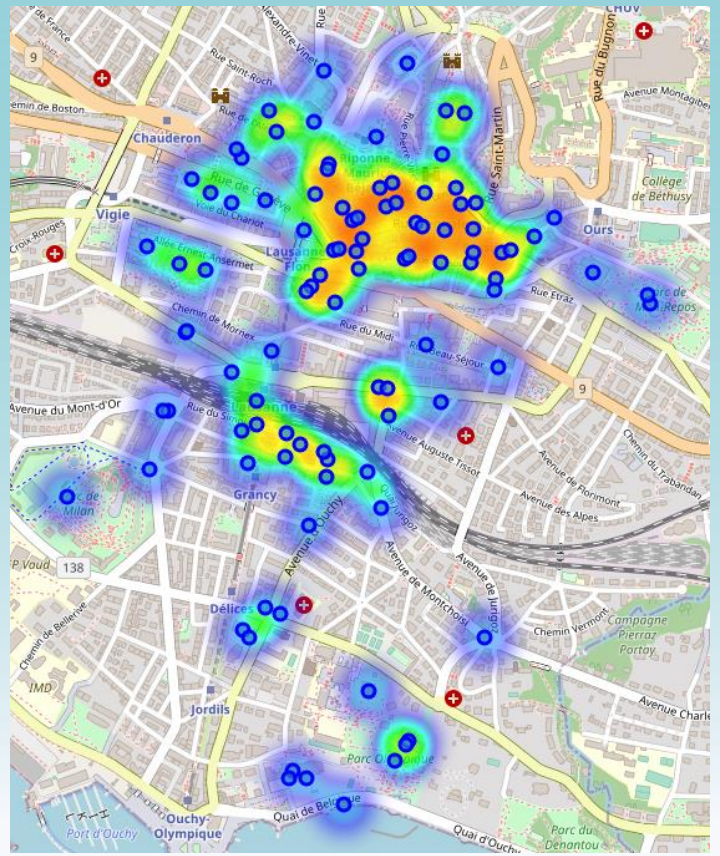
- k = 7

And we obtained a dataframe like this one:

	Cluster Labels	name	categories	lat	lng
0	0	Sleepy Bear Coffee	Coffee Shop	46.515338	6.631369
1	1	Les Trois Rois	French Restaurant	46.515515	6.631223
2	5	Pasta e Sfizi	Italian Restaurant	46.516374	6.633477
3	0	Café du Simplon	Mediterranean Restaurant	46.515961	6.629948
4	2	Les Gosses du Québec	Bar	46.517072	6.633122

Results

Map before clustering:



After clustering: