Coursera Capstone Project

Francesco M.

May 2020

A Guide to Milan for Tourists and Investors

Introduction

The city of Milan is world renowned for its high end fashion, restaurants and exclusive nightclubs. Millions of tourist trail its streets each year in search of boutiques and fashion shows. Milan is also one of the biggest cities in Italy with a total surface area of more than 1500km^2. The pulsing heart of the city is the city center, where all the famous restaurants, hotels, boutiques and clubs are located. Yet these are clustered in various areas of the center, an insight to the various fashion, nightlife and restaurant districts would provide vital information both to tourists visiting the city and to investors who desire to open business.

(a) Navigli area (b) Montenapoleone area

Figure 1: Famous neighborhoods in Milan

The aim of this project in fact, is to localize these distinct areas in Milan's city center. As mentioned, this is essential both for tourists, who through this insight can reach their favourite restaurants, shops or clubs in such a big city, but also very important to investors or whoever wants to open a business in Milan. Through the categorization of the various areas in Milan's city center it will become evident that there is a clear distinction between the restaurants, boutiques and nightlife, this is a vital information when choosing where to open a new business.

Data

To provide such an analysis, the project shall use the geospatial data provided by the official website of the municipality of Milan:

- https://dati.comune.milano.it/dataset/ds634-numeri-civici-coordinate

The following website presents a dataset with all the addresses in the city of Milan, more than 63,000 entries. The website supports two different types of formats for download, JSON and CSV, both were originally zipped, so had to be unzipped before importing. Once the data is correctly acquired and cleaned, it can be transformed in a dataset.

	District	Postal Code	Road Name	Longitude	Latitude
0	1	20121	Bastioni DI PORTA NUOVA	9.189394	45.480053
1	1	20121	Bastioni DI PORTA VENEZIA	9.202396	45.475062
2	1	20121	Bastioni DI PORTA VOLTA	9.182029	45.479434
3	1	20121	Corso DI PORTA NUOVA	9.191710	45.475896
4	1	20121	Corso GIACOMO MATTEOTTI	9.195200	45.466907

Figure 2: Data imported in Pandas DataFrame

As by Figure **??**, the dataframe presents the district (Milan has a total of 9 ditricts) the postal code, the road name and the longitude and latitude. This dataframe shall be reduced to only the city center (district 1), and further reduced by means of *KMeans* clustering to generate a series of 100 evenly spread out point's of interest that shall characterize the city center, as shown in Figure **??**.

A clustering technique has been used in this early stage of data formatting to ensure that the distribution of data was evenly spread out along Milan's city center. In fact, randomly selecting points from the dataset would have resulted in a patchy distribution of datapoints, making the analysis less effective. This methodology shall be further explored in the following section.

	Cluster Labels	Postal Code	Road Name	Longitude	Latitude
0	0	20121	Via LUIGI ALBERTINI	9.182543	45.474717
1	1	20145	Via GIOVANNI RANDACCIO	9.167483	45.476815
2	2	20154	Piazza ERCOLE LUIGI MORSELLI	9.174736	45.478654
3	3	20129	Corso VENEZIA	9.204959	45.474350
4	4	20123	Via PIETRO AZARIO	9.168378	45.460085

Figure 3: District 1 DataFrame containing 100 evenly spread points

Once the 100 datapoints are located, a call shall be made to the Foursquare API, one for each datapoint, to explore all the venues around each datapoint. The endpoint shall be:

- https : //api.foursquare.com/v2/venues/explore

This call shall return venues of all categories around each datapoint. These can be added to a dataframe, then grouped for each point of interest, as shown in Figure **??**

	Road Name	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Bastioni DI PORTA VENEZIA	Italian Restaurant	Hotel	Pizza Place	Art Gallery	African Restaurant
1	Corso DI PORTA VIGENTINA	Restaurant	Wine Bar	Pizza Place	Italian Restaurant	Bistro
2	Corso VENEZIA	Italian Restaurant	Pizza Place	Café	African Restaurant	Art Gallery
3	Corso VITTORIO EMANUELE II	Boutique	Plaza	Italian Restaurant	Sporting Goods Shop	Monument / Landmark
4	Foro BUONAPARTE	Italian Restaurant	Café	Plaza	Ice Cream Shop	Platform

Figure 4: Top categories per point of interest

Moreover, these can be clustered to observe the distribution of different categories in Milan's city center. A final analysis can be done observing the two main categories illustrated in the introduction, the nightlife (bars, pubs and clubs) and fashion (boutiques and clothing stores). This will give the desired insight in the distinction between these two different areas of Milan. These clusters shall be amply explained in the following sections.

Code and Methodology

This section shall explore the code used to complete such analysis. The geospatial data shall be imported from the website illustrated in the previous section in JSON (JavaScript Object Notation) format. As the data is zipped, the ZipFile() function must be used to unzip it, the importing procedure is shown in Figure **??**.

```
In [3]: link_json = 'http://dati.comune.milano.it/dataset/5c6519f6-6d26-41c9-b53b-6106e08d1b90/resource/cc
        9a206d-aac1-42c7-a8cc-ba11e500e488/download/ds634_civici_coordinategeografiche_20200403_json.zip'
        access_url = urllib.request.urlopen(link_json)

        # its a zipped json file so we first unzip it then read it
        zf = ZipFile(BytesIO(access_url.read()))
        zdata = zf.read('ds634_civici_coordinategeografiche_20200403.json')

        # zdata is type bytes so we cast it to string format
        s = str(zdata,'utf-8')

        # the string contains all the entries separated by \n, split these and create a series of lists, o
        ne for each datapoint
        s = str(zdata,'utf-8')
        lists = s.split('\n')
```

Figure 5: JSON unzipping and importing

The above code unzips and reads the JSON file, the data is then casted to string format, with a 'utf-8' encoding specific. Each datapoint in the string is separated by an end of

line character, therefore the split function is used. This returns a list, 63117 items long, each item is a specific adress in the city of Milan, therefore a datapoint that shall be used for the analysis. To visualize and handle this information, it shal be transformed into a pandas' dataframe. This is done reading one item at a time in the list, and extracting the relevant information, District, postal code, road name, longitude and latitude. Finally, to reduce the dataset the datapoints with same road name shall be grouped, therefore the resulting dataset shall have one point for every road in Milan, as in Figure **??**.

In [4]:
```
column_names = ['District', 'Postal Code', 'Road Name', 'Longitude', 'Latitude']
df_milan = pd.DataFrame(columns=column_names)

District = []
Name = []
Post_code = []
Lng =[]
Lat = []
for i in lists:
    # convert to dict with json
    dic = json.loads(i)
    # from dict extract the keys we want, but only chose the lines with postal code and from the c
entral districts
    if dic['CAP'] != None:
        Dist = dic['MUNICIPIO']
        PC = dic['CAP']
        Name = dic['TIPO'] + ' ' + dic['DESCRITTIVO']
        Lng = dic['LONG_WGS84']
        Lat = dic['LAT_WGS84']
        df_milan = df_milan.append({'District':Dist, 'Postal Code':PC, 'Road Name':Name, 'Longitud
e':Lng, 'Latitude':Lat}, ignore_index=True)

df_milan = df_milan.groupby(['District', 'Postal Code', 'Road Name'], as_index=False).mean()
df_milan.head()
```

Out[4]:

	District	Postal Code	Road Name	Longitude	Latitude
0	1	20121	Bastioni DI PORTA NUOVA	9.189394	45.480053
1	1	20121	Bastioni DI PORTA VENEZIA	9.202396	45.475062
2	1	20121	Bastioni DI PORTA VOLTA	9.182029	45.479434
3	1	20121	Corso DI PORTA NUOVA	9.191710	45.475896
4	1	20121	Corso GIACOMO MATTEOTTI	9.195200	45.466907

Figure 6: Milan road address dataset

The dataframe contains information on the outskirts of Milan, which shall not be used in this analysis, therefore, to focus only on the city center, only the points with District 1 shall be kept. This still leaves us with more than 700 datapoints, one for each road in the city center. To have a better view of the global trend in the city center and not focus on smaller areas we can reduce this dataframe to 100 points evenly spread out along the city center, which shall be the *points of interest*.

Randomly selecting 100 datapoints from the dataframe is not the ideal solution however, this would result in a patchy distribution which would not cover the center in a uniform and homogenous way. To obviate to this problematic we can use a clustering technique as in Figure **??**.

The above code clusters the datapoints in 100 clusters, based on their geographical location. One point is then randomly selected out of ach of the 100 clusters to be used as a point of interest. The result of such operation is visible in Figure **??**. On the left we see the initial datapoints, clustered into 100 clusters, each cluster is made up of 6/7

In [5]:
```
# We shall pick only district number 1, which is the city center
df_centro = df_milan[df_milan['District'] == '1']

# To reduce the size of the dataset (so it can be passed to foursquare) we cluster the data so it
is distributed evenly
k = 100
df_cluster = df_centro.drop('Road Name', axis=1)
kmeans = KMeans(init='k-means++', n_clusters=k, n_init=12)
kmeans.fit(df_cluster)
df_centro.insert(0, 'Cluster Labels', kmeans.labels_)
```

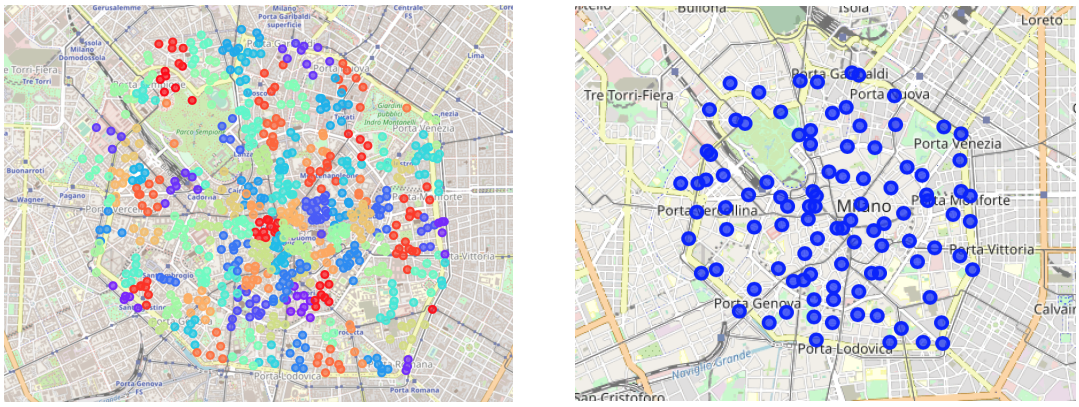
In [8]:
```
# of each of the 100 clusters we randomly select 1 so we are left with 100 evenly distributed poin
ts of interest
df_reduced = df_centro.groupby('Cluster Labels').apply(lambda x: x.sample(1)).reset_index(drop=Tru
e)
df_reduced = df_reduced.drop('District', axis=1)
df_reduced.rename
df_reduced.head()
```

Out[8]:

	Cluster Labels	Postal Code	Road Name	Longitude	Latitude
0	0	20121	Via LUIGI ALBERTINI	9.182543	45.474717
1	1	20145	Via GIOVANNI RANDACCIO	9.167483	45.476815
2	2	20154	Piazza ERCOLE LUIGI MORSELLI	9.174736	45.478654
3	3	20129	Corso VENEZIA	9.204959	45.474350
4	4	20123	Via PIETRO AZARIO	9.168378	45.460085

Figure 7: Selecting only the city center

datapoints very close together. On the right we can see the result of the reduction, the datapoints are distributed evenly around the city center, yet randomly chosen to ensure a fair analysis.



(a) Clustering of datapoints (b) Datapoint reduction

Figure 8: Points of interest around Milan

The location information of the points of interest are passed to the Foursquare API, using the *explore* endpoint on the *venues* group we can obtain information on all the venues around each of our datapoints. The relevant information from the response is stored in a dataframe, as illustrated in Figure **??**.

The dataframe contains all the venues in a 500 meter radius from the point of interest, of these venues the location, name and category is stored. A first analysis that can be done is to group all the venues for each point of interest, thus observe the most occuring in each datapoint. To do so we first use a one-hot encoding method, then group the results for each point of interest, this can be done as illustrated in Figure **??**.

Out[11]:

	Postal code	Road Name	Road Latitude	Road Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	20121	Via LUIGI ALBERTINI	45.474717	9.182543	Garibaldi Crème	45.474355	9.183466	Ice Cream Shop
1	20121	Via LUIGI ALBERTINI	45.474717	9.182543	Temakinho Brera	45.474651	9.183356	Sushi Restaurant
2	20121	Via LUIGI ALBERTINI	45.474717	9.182543	La Prosciutteria	45.474152	9.183449	Sandwich Place
3	20121	Via LUIGI ALBERTINI	45.474717	9.182543	Piccolo Teatro Studio Melato	45.472570	9.182809	Theater
4	20121	Via LUIGI ALBERTINI	45.474717	9.182543	Sugarwax	45.473760	9.181654	Spa

Figure 9: Venues dataframe

In [35]:
```
# group the venues with one hot encoding and group by point of interest:

onehot = pd.get_dummies(milan_venues[['Venue Category']], prefix="", prefix_sep="")
onehot['Road Name'] = milan_venues['Road Name']
first_col = [onehot.columns[-1]] + list(onehot.columns[:-1])
onehot = onehot[first_col]
milan_grouped = onehot.groupby('Road Name').mean().reset_index()
milan_grouped.head()
```

Figure 10: One hot encoding

Once the venues are grouped by occurence in each point of interest, we can read through each datapoint and select the five most occuring, adding these in a dataframe illustrated in Figure **??**.

In [38]:
```
# we create a dataframe with the top 5 categories for each postal code
def 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]

num_top_venues = 5
indicators = ['st', 'nd', 'rd']
columns = ['Road Name']
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))
top_venues = pd.DataFrame(columns=columns)
top_venues['Road Name'] = milan_grouped['Road Name']
for ind in np.arange(milan_grouped.shape[0]):
    top_venues.iloc[ind, 1:] = common_venues(milan_grouped.iloc[ind, :], num_top_venues)

top_venues.head()
```

Out[38]:

	Road Name	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Bastioni DI PORTA VENEZIA	Italian Restaurant	Hotel	Pizza Place	Art Gallery	African Restaurant
1	Corso DI PORTA VIGENTINA	Restaurant	Wine Bar	Pizza Place	Italian Restaurant	Bistro
2	Corso VENEZIA	Italian Restaurant	Pizza Place	Café	African Restaurant	Art Gallery
3	Corso VITTORIO EMANUELE II	Boutique	Plaza	Italian Restaurant	Sporting Goods Shop	Monument / Landmark
4	Foro BUONAPARTE	Italian Restaurant	Café	Plaza	Ice Cream Shop	Platform

Figure 11: Most occuring venues per point of interest

This insight is crucial to group the points of interest in the city center. Using this informa-

tion we can cluster the 100 points of interest in 5 different catgories, again using KMeans. on the grouped dataset we created, and shown in Figure **??**. We use this dataset beacuse it quantitavely defines the frequency of each venue category in every point of interest. We fit our KMeans model with the *milan_merged* dataframe, as in Figure **??**. We then extract the cluster labels and add them to the *top_venues* dataframe.

In [39]:
```
# divide in 5 different clusters and fit
k = 5
milan_clusters = milan_grouped.drop('Road Name', 1)
kmeans = KMeans(init='k-means++', n_clusters = k, n_init=12)
kmeans.fit(milan_clusters)

# display in a dataframe the cluster label
top_venues.insert(0, 'Cluster Labels', kmeans.labels_)
#milan_merged.drop(['index'], axis=1, inplace=True)

# merge the top 5 dataframe and the neighorood dataframe on postal code
milan_merged = pd.merge(df_reduced, top_venues, on='Road Name')

milan_merged
```

Out[39]:

	Cluster Labels_x	Postal Code	Road Name	Longitude	Latitude	Cluster Labels_y	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mo Comm Venue
0	0	20121	Via LUIGI ALBERTINI	9.182543	45.474717	3	Italian Restaurant	Ice Cream Shop	Cocktail Bar	Café	Japane Restaur
1	1	20145	Via GIOVANNI RANDACCIO	9.167483	45.476815	1	Italian Restaurant	Cocktail Bar	Pizza Place	Japanese Restaurant	Hotel
2	2	20154	Piazza ERCOLE LUIGI MORSELLI	9.174736	45.478654	1	Italian Restaurant	Cocktail Bar	Pizza Place	Chinese Restaurant	Tram Station
3	3	20129	Corso VENEZIA	9.204959	45.474350	3	Italian Restaurant	Pizza Place	Café	African Restaurant	Art Gall
4	4	20123	Via PIETRO AZARIO	9.168378	45.460085	3	Italian Restaurant	Café	Pizza Place	Supermarket	Pub

Figure 12: Clustering points of interest

So each datapoint is assigned a cluster number, *Cluster Labels_y*, from 0 to 4, essentially grouping the points of interest in 5 categories. By grouping these categories, as in Figure **??**, we can observe what they represent.

In [40]:
```
# let's see what the 5 clusters represent
top_venues.insert(0, 'Cluster Labels_y', kmeans.labels_)
clusters = top_venues.groupby('Cluster Labels_y').agg(lambda x:x.value_counts().index[0])
clusters.drop(['Road Name'], axis=1, inplace=True)
clusters
```

Out[40]:

	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
Cluster Labels_y						
0	0	Hotel	Hotel	Plaza	Italian Restaurant	Japanese Restaurant
1	1	Italian Restaurant	Cocktail Bar	Pizza Place	Ice Cream Shop	Hotel
2	2	Italian Restaurant	Plaza	Café	Hotel	Ice Cream Shop
3	3	Italian Restaurant	Ice Cream Shop	Café	Café	Pizza Place
4	4	Boutique	Plaza	Italian Restaurant	Women's Store	Monument / Landmark

Figure 13: 5 categories of points of interest in Milan city center

In the results section these shall be extensively analysed and placed on a folium map for easy visualization.

Results

In the previous section we have divided Milan city center in 100 evenly spread datapoints, then using Foursquare API location data provider we have found venues around every datapoint, finally these venues have been grouped and clustered in 5 different categories. Let's see what these 5 categories represent:

- Cluster 0: This is the famous touristic area of Milan, where all the important hotels are
- Cluster 1: The presence of a high umber of cocktail bars and restaurants indicates that this is the center of the Milan nightlife.
- Clusters 2 and 3: The high concentration of coffee shops and ice cream shops indicates that these are the daytime tourist spots in the city center.
- Cluster 4: This is undoutebly the famous fashion district, as shoen by the high concentration of boutiques and women clothing stores.

Having identified these 5 distinct areas, it is insightful to place these on a *folium* map. This is illustrated in Figure **??**, in which the single clusters have been color coded for easy reference.

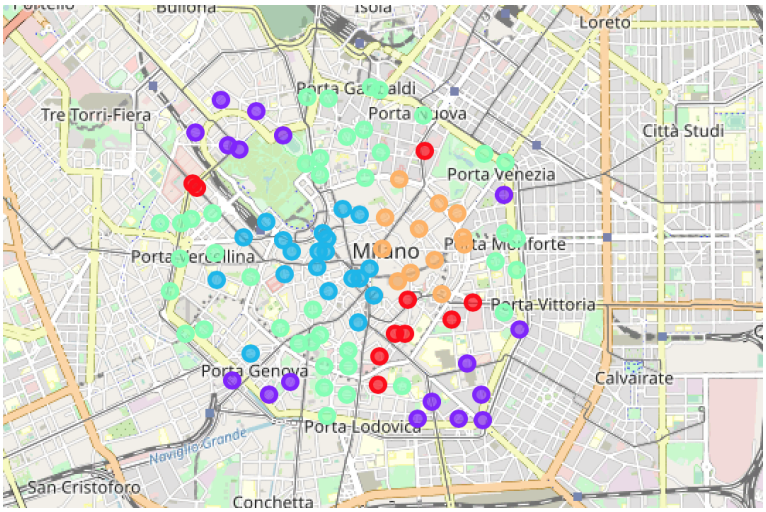


Figure 14: Distribution of clusters in Milan city center

The map is color coded in the following way:

- Cluster 0: Red
- Cluster 1: Purple
- Clusters 2: Blue

- Clusters 3: Green
- Cluster 4: Orange

The distinctions between the various areas are clear, we can see that Cluster 0, which we identified with the more touristic area of the city, matches famous areas such as the famous 'Duomo' cathedral and other highly touristic venues. As mentioned, Cluster 4 in orange represents the famous fashion district, which is located in the Eastern half of the city center as shown on the map.

Another key area identified by the map is Cluster 1 in purple, located in three distinct areas of the center: 'Navigli', 'Porta Romana' and 'Porta Sempione'. These represent the pulsating heart of the Milan nightlife, as for the high concentration of cocktail bars and restaurants.

We can further concentrate on two aspect of 'nightlife' and 'fashion', which prove central for the economy of Milan, attracting many tourists and investors all year round. To do so the venues dataframe has been reduced to include only these two categories, represented by the Foursquare API as 'boutique', 'Clothing Store', 'Pub' and others. These specific venues have been marked on a new folium map and are illustrated in Figure **??**

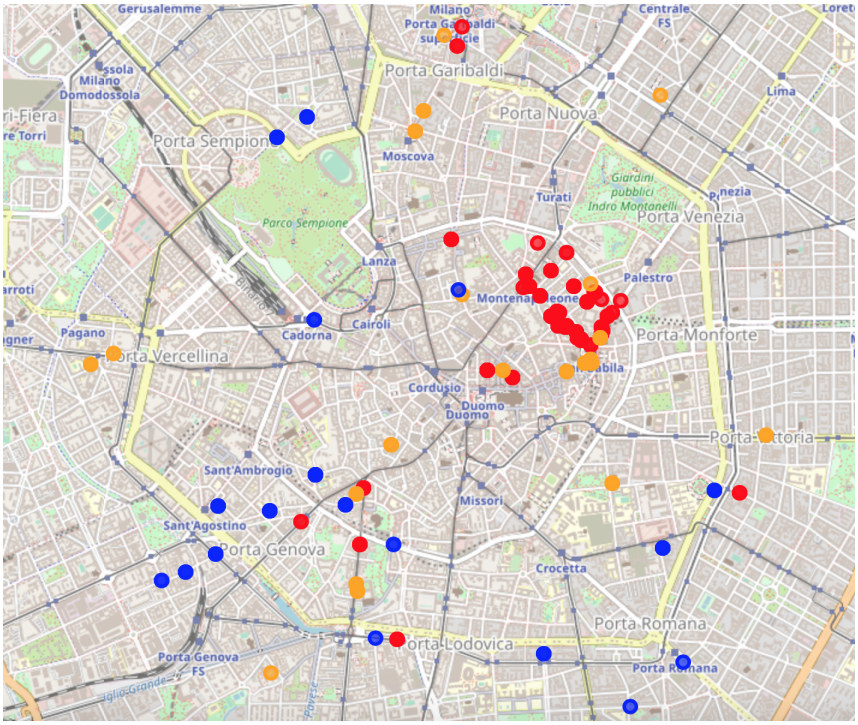


Figure 15: Distribution of fashion and nightlife venues in city center

The results seem to match the cluster identification discussed previously. As evident, the fashion district is located in the North-Eastern part of the city center whereas the cocktail bars, pubs and clubs in the South-West.

We can see a further distinction in the grouping of the fashion district, where all the high-end boutiques are located in a very tight distribution and mainly on one road, 'Via Montenapoleone' and are marked in red, whereas the lower price range clothing stores are located around this area and are marked in orange.

Observations

Figures **??** and **??** clearly illustrate the distrinction between various venue categories and their location around the city center. It is clear therefore that business startegies must be shaped to reflect this distribution.

The fashion economy clearly thrives around 'Via Montenapoleone' this is where tourists or locals are willing to spend great amounts of money for high-end fashion. It is clear therefore that if one wants to join this market with an exclusive brand, the boutique must be located close to that area, to gain visibility from its target audience, which will do its shopping there.

Similarly if an investor wants to open a cocktail bar, it must locate its business in the area where people tend to hang out at night, in order to gain visibility from its target audience. This therefore must be done in the purple clusters, around 'Porta Genova', 'Porta Sempione', 'Porta Romana' or 'Navigli'.

These two business sectors rely heavily on their location in order to thrive and make profit. It is clear that if a cocktail bar is placed far from where people hang out at night, for example in the more tourist areas of the city center, such as clusters 2 and 3, it will miss most of its target audience and likely fail. An identical observation can be made for the fashion economy, in which shop location is key to b visible to target audience.

Similarly, if the business is a hotel, it is essential that this be located centrally and close to the monuments, landmarks and all the touristic sites to attract tourists when chosing their stay. As visible in Figure **??** this is exactly where tha majority of hotels are.

Conclusion

In conclusion, this project has characterized Milan's city center by locating 100 evenly spread out points of interest through a preliminary clustering and analysing the distribution of venues around each one using Foursquare API. This has allowed to emphasize the division of various clusters in Milan, as in Figure **??**.

As stated in the previous section, this information is central to develop correct business strategies. By localizing the districts of the market an investor wants to target, he/she can precisely decide where to locate the business to have maximum visibility by its target audience.