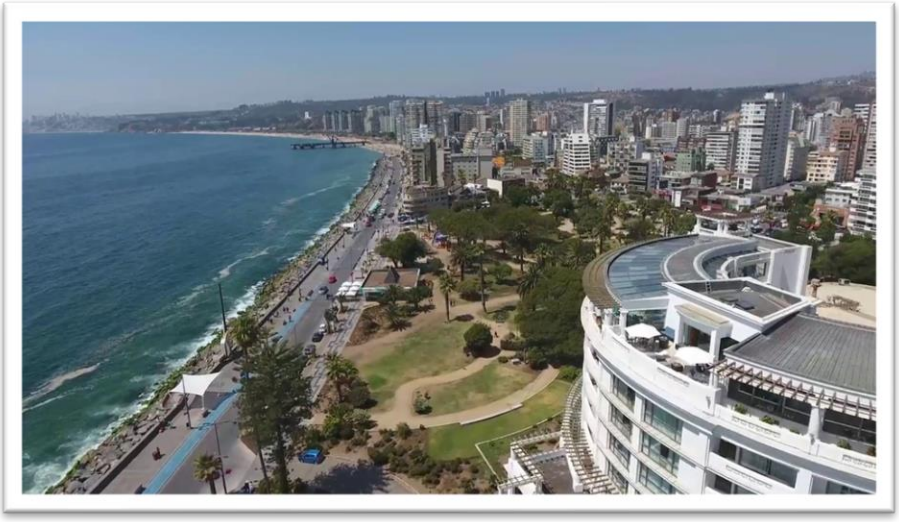
Viña del Mar vs. Valparaíso
Coursera Capstone Project



Introduction

This document explains what I did for the Coursera Capstone Project of the IBM Data Engineering Professional Certificate.

In my country, Chile, there are two cities very close to each other but very different too: Santiago and Valparaiso. Both are very touristic cities but there are very noticeable differences when you travel there, such as Viña del Mar being way wealthier than Valparaiso, having more commerce and big events, but Valparaiso has a lot of more historical places, trading, and old stores that refuse to die.

These differences are very important to take into account when trying to open a specific business in one of those two cities, therefore it would be very valuable to have insights obtained from real data in order to give eventual business owners and entrepreneurs the opportunity to make an informed decision.

My client is wondering what kind of business to open in Valparaíso. To do this, I will compare both cities to see what is missing from each other, how similar they are, and see how I could guide my client to make a good and informed decision.

Data

The data available at Foursquare gives the unique opportunity of clustering based on the venues that are in them, thus allowing to compare specific areas of a city and compare that with areas from another city.

To gain insights about what kind of business should be made within a city, I would like to compare both cities using Foursquare's API and finally be able to point out the advantages of one city over the other when trying to open a business.

Since there are many hills surrounding both cities and some hills are considered by Foursquare as their own city, due to wrong user input, I will focus on the city's central plaza.

To request the data from Foursquare's API, I decide to obtain 100 venues for each city under the geographical restriction defined earlier.

From Foursquare and after cleaning the data a bit, I obtain the following data for each venue in each city:

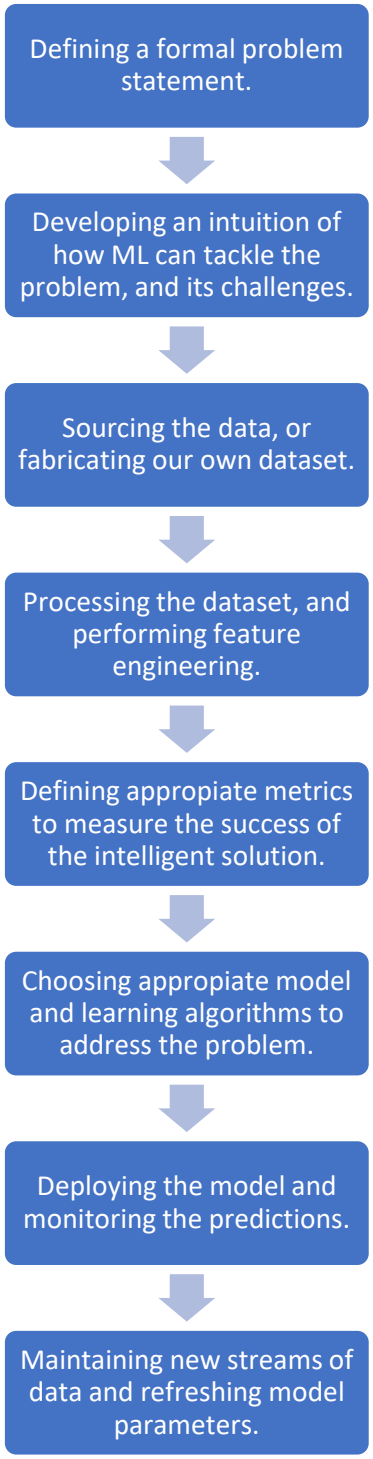
- Name.
- Category they belong to.
- Address.
- Cross street.
- Latitude.
- Longitude.
- Distance.
- Postal code.
- Country code.
- City.
- State.
- Country.
- Venue ID.

Output example

```
Out[37]: {'reasons': {'count': 0,
           'items': [{'summary': 'This spot is popular',
             'type': 'general',
             'reasonName': 'globalInteractionReason'}]},
          'venue': {'id': '506f4611e4b0184b5aa12d2d',
           'name': 'Frank Hostel',
           'location': {'address': 'Avenida Valparaiso',
            'crossStreet': 'Etchevers',
            'lat': -33.02446367351446,
            'lng': -71.55465197864842,
            'labeledLatLngs': [{'label': 'display',
              'lat': -33.02446367351446,
              'lng': -71.55465197864842}],
            'distance': 262,
            'postalCode': '2571511',
            'cc': 'CL',
            'city': 'Viña del Mar',
            'state': 'Valparaíso',
            'country': 'Chile',
            'formattedAddress': ['Avenida Valparaiso (Etchevers)',
             '2571511 Viña del Mar',
             'Valparaíso',
             'Chile']},
           'categories': [{'id': '4bf58dd8d48988d1ee931735',
             'name': 'Hostel',
             'pluralName': 'Hostels',
             'shortName': 'Hostel',
             'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/travel/hostel_',
              'suffix': '.png'},
             'primary': True}],
           'photos': {'count': 0, 'groups': []},
           'venuePage': {'id': '42010257'}},
          'referralId': 'e-0-506f4611e4b0184b5aa12d2d-0'}
```

Methodology

Before starting this project, I decided to follow the workflow set on the *book Sculpting Data for ML* by Jigyasa Grover and Rishabh Misra. This workflow goes as follows:



After applying this methodology to the project, I define the following steps or statements about each part of that workflow:

Defining a formal problem statement

What kind of business should I open, if I must choose between Valparaiso and Viña del Mar as my city.

Developing an intuition of how ML can tackle the problem, and its challenges

I could cluster based on the category of each venue to see if the market is already saturated for a specific venue category in a certain city, or what does a city have in common with the other. This way, I could see if my business' category fits within a specific city's needs, and which one should I go for.

Sourcing the data, or fabricating our own dataset

All the data will be sourced from Foursquare API.

Processing the dataset, and performing feature engineering

For the dataset obtained from Foursquare, I have processed the data to separate some columns into more useful ones. I have also decided to focus on each city's central plaza as the center of my Foursquare request, because the information submitted to Foursquare by some users about venues located on the hills of each city is wrong. For example, I obtained venues located in 'Cerro Concepción' as a city, but Cerro Concepción is part of Valparaíso. Due to this, I had to focus in a specific area. Example output:

```
In [49]: 1 dataframeFilteredVina.head(10)
```

Out[49]:

	name	categories	address	crossStreet	lat	lng	labeledLatLngs	distance	postalCode	cc	city
0	Frank Hostel	Hostel	Avenida Valparaiso	Etchevers	-33.024464	-71.554652	[{'label': 'display', 'lat': -33.024463673514...	262	2571511	CL	Viña del Mar
1	Purolivo	Gourmet Shop	Galería Somar	Loc. 6-9	-33.024226	-71.553209	[{'label': 'display', 'lat': -33.0242261422981...	133	NaN	CL	Viña del Mar
2	Panzoni	Italian Restaurant	Paseo Cousiño 12	e/ Viana y Av. Valparaíso	-33.025731	-71.553494	[{'label': 'display', 'lat': -33.0257310185469...	199	NaN	CL	Viña del Mar
3	Fuente de Soda Cevasco	Hot Dog Joint	Av. Valparaiso 700	NaN	-33.024952	-71.552665	[{'label': 'display', 'lat': -33.0249521757472...	86	NaN	CL	Viña del Mar
4	Déjà Vu	Latin American Restaurant	Calle Viana 144, 2do. piso	Paseo Cousiño	-33.025785	-71.553563	[{'label': 'display', 'lat': -33.0257852977384...	208	NaN	CL	Viña del Mar
5	Bogarín	Juice Bar	Av. Valparaíso 533	Quinta	-33.024535	-71.554497	[{'label': 'display', 'lat': -33.0245345160844...	247	NaN	CL	Viña del Mar
6	La Nonna	Fast Food Restaurant	Quinta 255	NaN	-33.025106	-71.554167	[{'label': 'display', 'lat': -33.0251055711397...	224	NaN	CL	NaN
7	Quinta Vergara	Forest	NaN	NaN	-33.027990	-71.552421	[{'label': 'display', 'lat': -33.0279900316624...	382	NaN	CL	NaN
8	Hotel Pacifico	Hotel	2 Poniente 154	NaN	-33.020759	-71.553613	[{'label': 'display', 'lat': -33.0207588089243...	456	NaN	CL	NaN
9	Empanadas Royal	Diner	Traslaviña 138	NaN	-33.023503	-71.558202	[{'label': 'display', 'lat': -33.0235028231584...	605	NaN	CL	Viña del Mar

Defining appropriate metrics to measure the success of the intelligent solution.

I will be using the latitude, longitude, name, category and city of each venue.

Choosing appropriate model and learning algorithms to address the problem.

Due to the data chosen and the clustering approach I want to take, I will be using a k-means clustering model.

Deploying the model and monitoring the predictions.

To deploy the model I have chosen, I have used IBM Watson Studio from the IBM Cloud Pak for Data to run a Jupyter notebook. This notebook has a Python kernel and I have imported the most common libraries for this kind of job, which are numpy, pandas, scikit-learn, requests and more. With these tools, I got a clustering output from scikit-learn's K-means clustering library, which will be explained next.

Maintaining new streams of data and refreshing model parameters.

This was not necessary, due to the project being a single snapshot of both cities' reality at the moment.

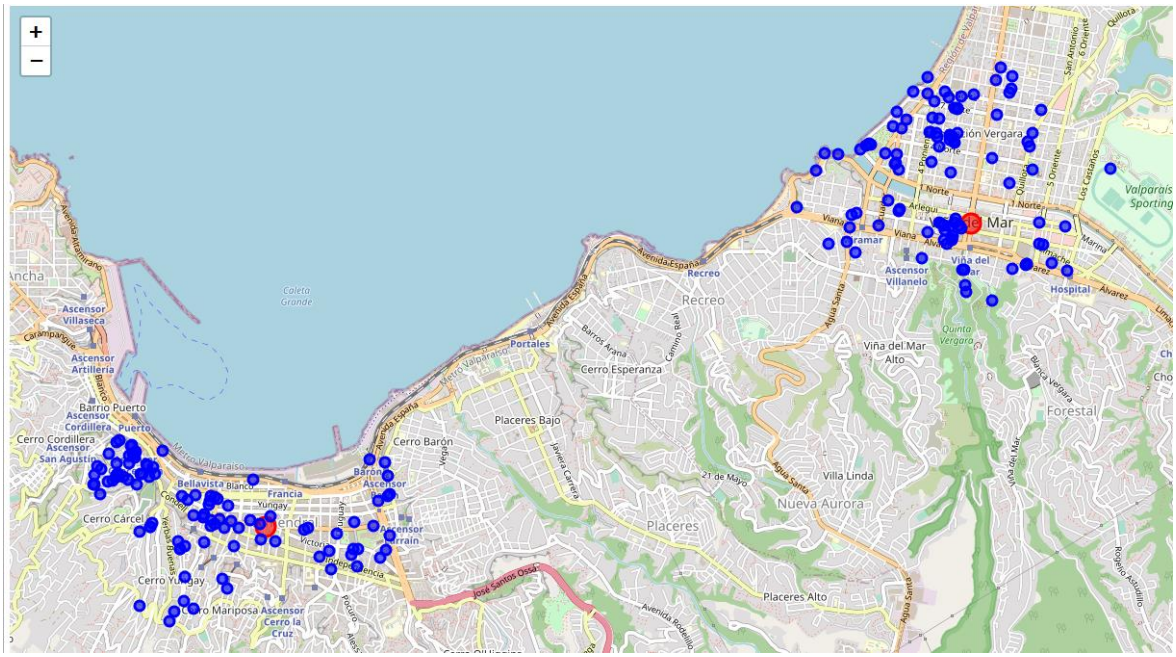
Results

First, with the data requested, cleaned, processed and every other step mentioned in the previous section being made, I made a map of both cities using Folium's library with the data I got, to corroborate if the requested locations on the API call were correct and to confirm if I will follow to the clustering step as is. The code I wrote is as follows:

Valparaiso and Viña del Mar:

In [184]:
```
latitudeValpoVina = -33.037395
longitudeValpoVina = -71.584546

venuesMapValpoVina2 = folium.Map(location=[latitudeValpoVina,longitudeValpoVina],zoom_start=14)

folium.CircleMarker(
    [latitudeVina,longitudeVina],
    radius=10,
    popup='Center of Viña del Mar',
    fill=True,
    color='red',
    fill_color='red',
    fill_opacity=0.6).add_to(venuesMapValpoVina2)

folium.CircleMarker(
    [latitudeValpo,longitudeValpo],
    radius=10,
    popup='Center of Valparaiso',
    fill=True,
    color='red',
    fill_color='red',
    fill_opacity=0.6).add_to(venuesMapValpoVina2)

for lat, lng, label in zip(dataframeValpoVina.lat, dataframeValpoVina.lng, dataframeValpoVina.categories):
    folium.CircleMarker([lat,lng],radius=5,fill=True,popup=label,color='blue',fill_color='blue',fill_opacity=0.6).add_to(venuesMapValpoVina2)


display(venuesMapValpoVina2)
```

And the output map I obtained after running the code in my Jupyter notebook is:



After that, I one-shotted each venue based on its category and calculated the frequency within each city, by grouping by city and taking the mean of occurrence within a city compared to other venues.

	city	Art Gallery	Art Museum	Austrian Restaurant	Bagel Shop	Bakery	Bar	Beach	Bed & Breakfast	Beer Bar	...	Supermarket	Surf Spot	Sushi Restaurant	Tailor Shop	T R
0	Valparaíso	0.022222	0.011111	0.000000	0.000000	0.033333	0.033333	0.000000	0.022222	0.000000	...	0.000000	0.0	0.011111	0.000000	0
1	Viña del Mar	0.000000	0.011494	0.011494	0.011494	0.000000	0.022989	0.011494	0.045977	0.011494	...	0.011494	0.0	0.080460	0.011494	0

2 rows × 90 columns

This being done, I was able to sort this data and see which places are most common to each city:

```
----Viña del Mar----
                 Venue  Freq
0    Sushi Restaurant  0.08
1         Coffee Shop  0.07
2  Italian Restaurant  0.06
3               Hotel  0.05
4     Bed & Breakfast  0.05
5      Ice Cream Shop  0.03
6         Pizza Place  0.03
7          Restaurant  0.02
8              Hostel  0.02
9               Diner  0.02


----Valparaíso----
                  Venue  Freq
0           Restaurant  0.08
1          Pizza Place  0.07
2                Hotel  0.06
3         Neighborhood  0.03
4               Bakery  0.03
5                  Bar  0.03
6  Peruvian Restaurant  0.03
7                Café  0.03
8       Ice Cream Shop  0.02
9       Sandwich Place  0.02
```

This can also be compared to the results obtained if I see the most common venues directly from the original dataframe, though I can't see the frequency of occurrence there:

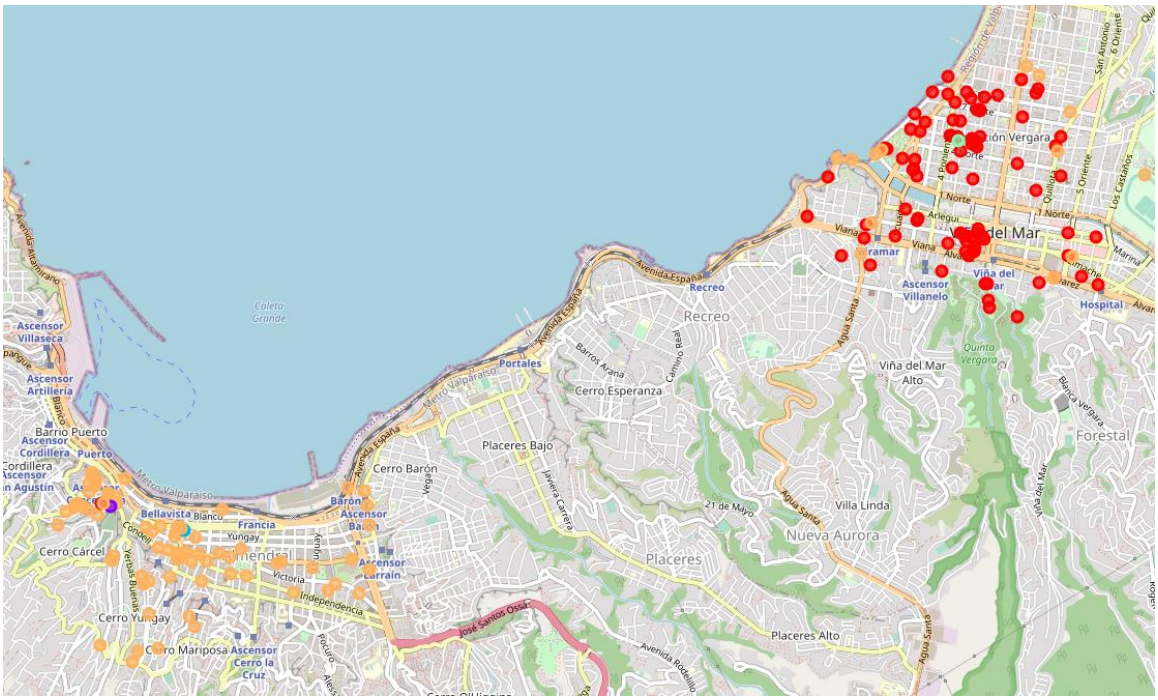
```
In [135]: df = dataframeFilteredValpo.groupby('categories').count()
          df['id'].sort_values(ascending=False).head(10)

Out[135]: categories
          Restaurant              9
          Hotel                   7
          Pizza Place             6
          Bar                     4
          Café                    4
          Bakery                  3
          Dessert Shop            3
          Peruvian Restaurant     3
          Scenic Lookout          3
          Neighborhood            3
          Name: id, dtype: int64
```

```
In [141]: df = dataframeFilteredVina.groupby('categories').count()
          df['id'].sort_values(ascending=False).head(10)

Out[141]: categories
          Sushi Restaurant      7
          Coffee Shop           6
          Italian Restaurant    5
          Bed & Breakfast       5
          Hotel                 5
          Ice Cream Shop        4
          Burger Joint          3
          Pizza Place           3
          Tea Room              3
          Dessert Shop          2
          Name: id, dtype: int64
```

This already gives me a valuable insight to get a recommendation for my client, because I know what businesses a city already has a lot of.

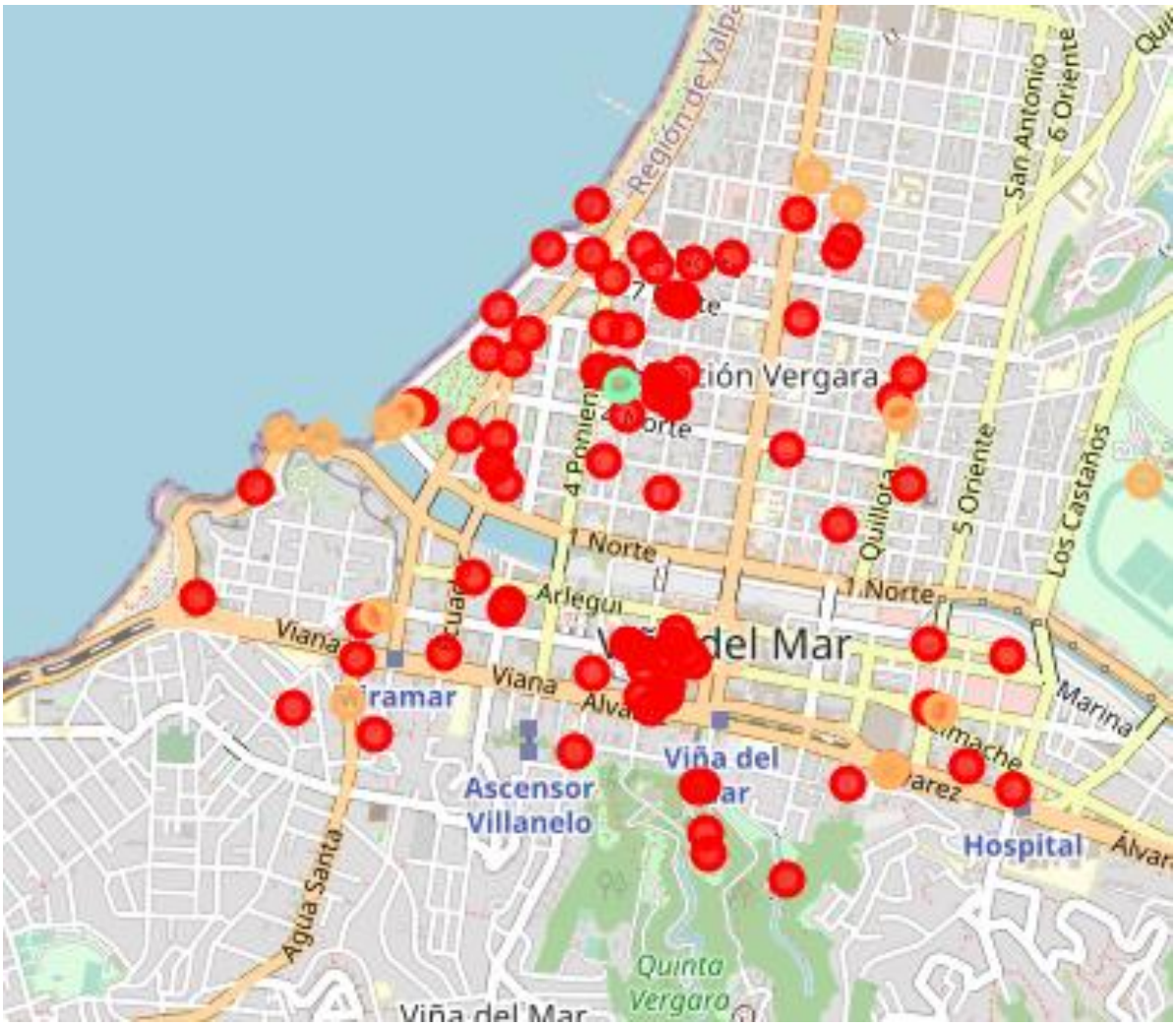
Finally, using all the data I have obtained, I do a k-means clustering and make a map using the data I obtained from that:

```
In [248]: map_clusters = folium.Map(location=[latitudeValpoVina,longitudeValpoVina],zoom_start=11)
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0,1,len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

markers_colors = []
for lat,lon,poi,cluster in zip(vinaValpoMerged3['lat'],vinaValpoMerged3['lng'],vinaValpoMerged3['name'],vinaValpoMerged3['Cluster Labels'].dropna()):
    label = folium.Popup(str(poi)+ ' Cluster ' + str(cluster), parse_html=True)
    cluster = int(cluster)
    folium.CircleMarker(
        [lat,lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```



The nodes in Valparaiso's cluster are very homogeneous, this means the city venues have a category that's very similar to each other, with specific outlier nodes that have other colors.

A big difference can be seen in Viña del Mar, where there are nodes that are the same color as the nodes from Valparaiso's cluster.


Región de Valpa
San Antonio
6 Oriente
ión Vergara
4 Poniente
1 Norte
Quillota
5 Oriente
Los Castaños
Arlegui
Viana
del Mar
Viana
Marina
Viña del
Ascensor
Villanelo
Hospital
Álvar
Agua Santa
Quinta
Vergara
Viña del Mar

Discussion

Something I noticed from the data I have used is that it seems to be missing quite a lot of venues. This can be explained due to the fact that Foursquare has never been popular within citizens of my country in comparison with other map services like Google Maps and Mapcity.cl, so if I were to make this analysis again I would try to get data from a source that locals use more.

Also, the data is kind of outdated, which can be explained by the same reason as above. Nonetheless, it has been useful enough to recognize patterns, see what each city venues are like, and take conclusions from that.

Next time, I would like to see if there is any data available from local authorities about the sales each venue makes, and create a map based on that. I believe it would be interesting because we could also point exactly which areas of a city will make us the most money, due to the amount of money being invested there. That information could be clustered and the insights could be useful for several industries.

Conclusions

From the k-means clustering done on the dataset obtained from Foursquare, I can claim that Valparaiso venues are very similar and close to each other, but the Viña del Mar cluster has some nodes marked as if they were from Valparaiso's cluster. This clearly gives us an indication that Viña del Mar has some venues that are like Valparaiso's, while Valparaiso doesn't have venues that could be characteristic of Viña del Mar. This gives us a valuable insight: We have a big market opportunity if we open a business of a category that is successful in Viña del Mar but is missing from Valparaíso as of today.

We could also see from the dataset obtained that the venues in both cities are mostly restaurants and hotels. Valparaiso has always been considered a bohemian city and we can see that in our data, where the categories 'Bar' and 'Neighborhood' are more common, while Viña del Mar has always been seen as more tourist-friendly in comparison and is missing those tags.

In conclusion, from all the data obtained, I would suggest my client to open a restaurant in Valparaíso of the same category of a successful restaurant in Viña del Mar, but that has not been opened yet in Valparaíso. I would suggest specifically a sushi restaurant, considering that it's missing from the most frequent categories at Valparaiso while being the leader category in Viña del Mar. Also, being a coastal city, it has access to a big variety of fish, so a sushi restaurant is definitely the way to go.