COURSERA CAPSTONE PROJECT

OPENING A WELLNESS & SPA CHAIN IN KANTON OF ZUG, SWITZERLAND

SUMMARY

Evaluation of the 5 most expensive regions of kanton of Zug, Switzerland. Utilisation of various data science techniques in Python. FourSquare leveraging and Folium maps generation.

Mikołaj Wlazło

IBM APPLIED DATA SCIENCE PROFESSIONAL COURSE

JUNE, 2020

List of Contents

List of Tables

List of Figures

The complete notebook, with all codes and step by step solutions, can be found under:
https://github.com/MikolajWlazlo/Coursera_Capstone/blob/master/IBM%20Capstone%20Project.ipynb

1. Introduction

Wellness & spa is an active process of making choices toward a healthy and fulfilling life. Wellness/spa tourism is generally understood to be traveling for the purpose of enhancing health and wellbeing through the use of spas, preventative treatments and therapies, and it becomes more and more popular.

For many people, visiting wellness & spa centre is a great way to relax and enjoy themselves during weekends and holidays. Locations like this are like a one-stop destination for both locals and tourists.
Of course, as with any business decision, opening a chain of wellness & spa centres requires serious consideration, and is a lot more complicated than it seems. Particularly, the location of the centre is one of the most important decisions that will determine whether it will be a success or a failure.

2. Business Problem

A private investor is seeking a perfect location to open a chain of wellness & spa centres around Switzerland. The most expensive (in terms of gross annual rent per m^2) kantons are being targeted, and within them the locations with the same feature, and the lowest possible competition. This is motivated by the fact that this type of luxury centres mainly attract busy, mid-/high-class people.

Thus, the main objective of the project is to select the best locations where such centres can be put up, aiming at the above demographic, thereby helping the owners to achieve maximum profits.

Using data science methodology and machine learning techniques, this project aims to provide solutions to answer the business question: In Switzerland, if an investor is looking to open a wellness & spa chain, where would you recommend to open it?

The complete notebook, with all codes and step by step solutions, can be found under:
https://github.com/MikolajWlazlo/Coursera_Capstone/blob/master/IBM%20Capstone%20Project.ipynb

3. Data

To solve the problem, we will need the following data:

- List of kantons in Switzerland;
- List of gemeindens in kanton of Zug;
- Respective values for each region in the area of housing prices. This defines the scope of this project which is confined to the kanton of Zug, one of the main regions in the country of Switzerland, in Central Europe;
- Latitude and longitude coordinates of those gemeindens. This is required in order to plot the map and also to get the venue data;
- Venue data, particularly data related to services and leisure. We will use this data to perform clustering on the gemeindens and evaluate potential for the given problem.

The data for this project has been retrieved and processed through multiple sources, giving careful considerations to the accuracy of the methods used.

3.1 Regions & Gemeindens

The data for specific kanton in Switzerland and gemeinden in the kanton of Zug can be extracted from the webpage: https://realadvisor.ch/en/property-prices. The generated excel file was uploaded over to the notebook and read using pandas library for Python.

3.2 Geocoding

The file contents from *3 Zug Data.xlsx* are retrieved into a Pandas Data Frame.
The latitude and longitude of the gemeindens were gathered from: https://www.mapplus.ch.
The geometric location values are then stored into the initial Data Frame.

3.3 Venue Data

From the obtained location data, after Data Frames creation and Geocoding, the venue data is found out by passing in the required parameters to the FourSquare API, and creating another Data Frame to contain all the venue details along with the respective gemeindens.

FourSquare has one of the largest databases of 105+ million places, and is used by over 125,000 developers. FourSquare API will provide categories of the venue data for clustering and decision making.

This is a project that will make use of many data science skills, from web scraping, working with API (FourSquare), data cleaning, data wrangling, to machine learning (K-means clustering) and data visualization (Folium).

Next, the Methodology section is presented where discussion of the steps taken in this project takes place, along with the data analysis, and the used machine learning techniques.

4. Methodology

4.1 Data Acquisition & Wrangling

Firstly, the list of kantons in Switzerland, the list of gemeindens in the kanton of Zug, and the corresponding housing pricing data is needed. Fortunately, the information is available in the Internet. Extracting the data from aforementioned web pages, excel files were created, that later are loaded into the notebook using Python pandas.

Generated Data Frames are further modified to better fit the evaluation and ease navigation and understanding of the processed data.

Throughout the whole project, a basic data wrangling pandas functions are used, for instance: split, drop, sort, groupby, merge.

CODE:

```
CH.rename(columns={'Average price / m2 (Apartment)':'Apartment Price', 'Average price / m2 (House)':'House Price','Average price / m2 (Rent yearly)':'Rent Yearly'},inplace=True)
CH.drop(['Population'], axis=1, inplace=True)
CH.head()


CH[['None','Kanton']] = CH.Canton.str.split("Canton of",expand=True,)
Kanton = CH.drop(['Canton','None'], axis=1)
Kanton.head()


Zug5_merged.loc[Zug5_merged['Cluster Labels'] == 0, Zug5_merged.columns[[2] + list(range(5, Zug5_merged.shape[1]))]]
```


The complete notebook, with all codes and step by step solutions, can be found under:
https://github.com/MikolajWlazlo/Coursera_Capstone/blob/master/IBM%20Capstone%20Project.ipynb

4.2 Folium

Folium builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the choropleth library. All visualization are done with help of Folium.

The distribution of the Gross Annual Rent per m^2 in Switzerland has been reviewed and plotted on the Folium map (Mapbox Bright), with 5 the most expensive kantons marked (Zurich kanton bordering from the north with kanton Zug):

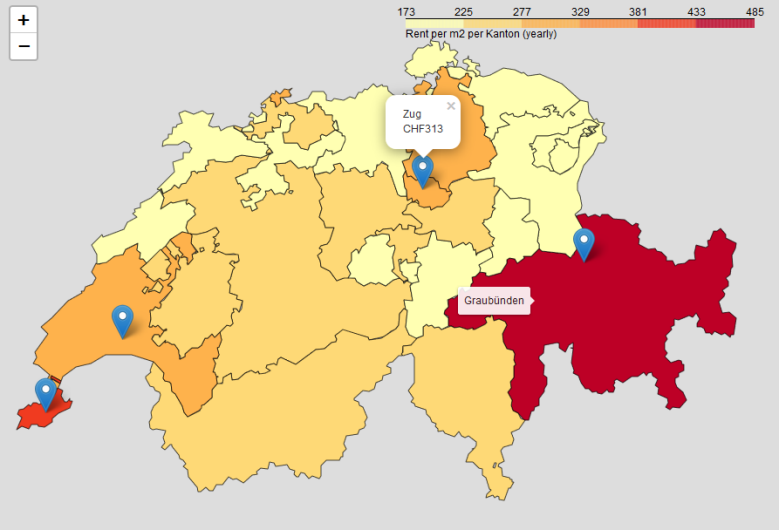

Figure 1 Kantons of Switzerland presented by Annual Rent per m^2 value (SwissMap)

	Annual rent / m^2 [CHF]	Kanton
1	485	Grisons
2	390	Geneva
3	313	Zug
4	296	Zurich
5	288	Vaud

Table 1 The most expensive kantons

CODE:

```
SwissMap = folium.Map(width=900, height=640, location=[46.8, 8.33], zoom_start=8,
tiles='Mapbox Bright')

chor = folium.Choropleth(
    geo_data='Swiss Map.geojson',
    name='choropleth',
    data=Kanton,
    columns=['Canton Number', 'Rent Yearly'],
    key_on='feature.properties.KANTONSNUM',
    fill_color='YlOrRd',
    fill_opacity=1.0,
    line_opacity=0.7,
    legend_name='Rent per m2 per Kanton (yearly)'
     ).add_to(SwissMap)

data = pd.DataFrame({
'lat':[46.8508, 46.2044, 47.1662, 47.3769, 46.5197],
'lon':[9.532,6.1432,8.5155,8.5417,6.62323],
'name':['Grisons CHF485', 'Geneva CHF390', 'Zug CHF313', 'Zurich CHF296', 'Vaud
CHF288']
})
for i in range(0,len(data)):
    folium.Marker([data.iloc[i]['lat'], data.iloc[i]['lon']],
popup=data.iloc[i]['name']).add_to(SwissMap)

chor.geojson.add_child(
    folium.features.GeoJsonTooltip(['NAME'],labels=False)
)
```


The complete notebook, with all codes and step by step solutions, can be found under:
https://github.com/MikolajWlazlo/Coursera_Capstone/blob/master/IBM%20Capstone%20Project.ipynb

Next, due to the increasing upper mid-class population and very attractive tax conditions, kanton of Zug was selected over other (sometimes even more expensive), presented regions. Within this kanton, the 5 most expensive communes (Gemeindens) were chosen.

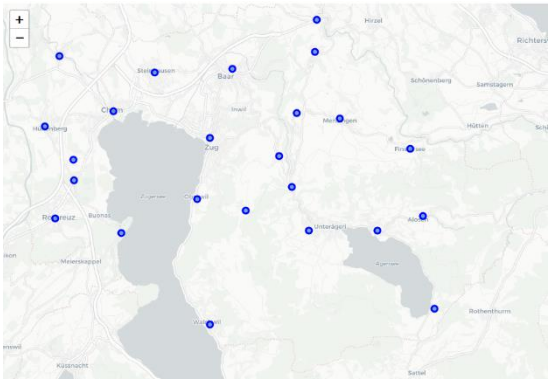

Figure 3 Gemeindens of the kanton of Zug (Map_Zug)

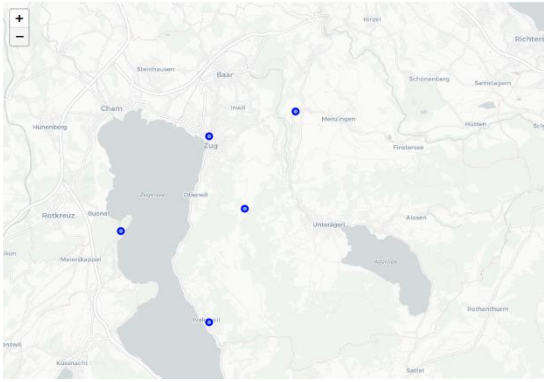

Figure 2 The most expensive gemeindens of the kanton of Zug (Map_Zug5)

CODE:

```
Map_Zug = folium.Map(width=900, height=640, location=[latitude, longitude],
zoom_start=11.5, tiles='cartodbpositron')

for lat, lng, mun, gem in zip(Zug['Latitude'], Zug['Longitude'],
Zug['Municipality'], Zug['Gemeinden']):
    label = '{}, {}'.format(gem, mun)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(Map_Zug)

Map_Zug5 = folium.Map(width=900, height=640, location=[latitude, longitude],
zoom_start=12, tiles='cartodbpositron')

for lat, lng, gem, rent in zip(Zug5['Latitude'], Zug5['Longitude'],
Zug5['Gemeinden'],Zug5['Rent Yearly']):
    label = '{}, {}'.format(gem, rent)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(Map_Zug5)
```


The complete notebook, with all codes and step by step solutions, can be found under:
https://github.com/MikolajWlazlo/Coursera_Capstone/blob/master/IBM%20Capstone%20Project.ipynb

4.3 FourSquare API

Further, the FourSquare database was leveraged to generate a Data Frame of top 100 venues found within a radius of 2000 from the selected gemeindens. A Foursquare Developer Account needed to be created in order to obtain the Foursquare ID and Foursquare secret key. This data, however, was deleted from the code due to confidentiality.

CODE:

```
LIMIT = 100
radius = 2000
url =
'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}
&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    gemeinden_latitude,
    gemeinden_longitude,
    radius,
    LIMIT)

results = requests.get(url).json()
```

Continuing, the API calls to Foursquare passing in the geographical coordinates of the gemeindens in a Python loop. Foursquare will return the venue data in JSON format and the venue name, venue category, venue latitude and longitude will be extracted. With the data, a check on how many venues were returned for each gemeinden, and examination on how many unique categories can be curated from all the returned venues can be carried out.

CODE:

```
def getNearbyVenues(names, latitudes, longitudes, radius=2000):

    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,
```


The complete notebook, with all codes and step by step solutions, can be found under:
https://github.com/MikolajWlazlo/Coursera_Capstone/blob/master/IBM%20Capstone%20Project.ipynb

```
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            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 = ['Gemeinden',
                  'Gemeinden Latitude',
                  'Gemeinden Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

4.4 One-hot Encoding

One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. For the K-means Clustering Algorithm, all unique items under Venue Category are one-hot encoded.

CODE:

```
Zug5_onehot = pd.get_dummies(Zug5_venues[['Venue Category']], prefix="",
prefix_sep="")

# add neighborhood column back to dataframe
Zug5_onehot['Gemeinden'] = Zug5_venues['Gemeinden']

# move neighborhood column to the first column
fixed_columns = [Zug5_onehot.columns[-1]] + list(Zug5_onehot.columns[:-1])
Zug5_onehot = Zug5_onehot[fixed_columns]
```

4.5 Top 5 Most Common Venues

Due to high variety in the venues, only the top 5 common venues are selected and a new Data Frame is made, which is used to train the K-means Clustering Algorithm.

CODE:

```
num_top_venues = 5

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

# create columns according to number of top venues
columns = ['Gemeinden']
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))

# create a new dataframe
Gemeinden_venues_sorted = pd.DataFrame(columns=columns)
Gemeinden_venues_sorted['Gemeinden'] = Zug5_grouped['Gemeinden']
```


The complete notebook, with all codes and step by step solutions, can be found under:
https://github.com/MikolajWlazlo/Coursera_Capstone/blob/master/IBM%20Capstone%20Project.ipynb

```
for ind in np.arange(Zug5_grouped.shape[0]):
    Gemeinden_venues_sorted.iloc[ind, 1:] =
return_most_common_venues(Zug5_grouped.iloc[ind, :], num_top_venues)
```

4.6 K-means Clustering

Lastly, we will perform clustering on the data by using k-means clustering. K-means clustering algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. It is one of the simplest and popular unsupervised machine learning algorithms, and is particularly suited to solve the problem for this project.

Gemeindens will be clustered into 3 clusters. The results will allow us to identify which gemeinden have higher concentration of potential competition, and which have lower.

CODE:

```
# set number of clusters
kclusters = 3

Zug5_grouped_clustering = Zug5_grouped.drop('Gemeinden', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters,
random_state=0).fit(Zug5_grouped_clustering)

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

5. Results

The results from the k-means clustering show that we can categorize the gemeindens into 3 clusters, based on the most popular venues within them:

- ✓ Cluster 0: Gastronomy
- ✓ Cluster 1: Retail Vending
- ✓ Cluster 2: Leisure

The results of the clustering are visualized in the map below with cluster 0 in red colour, cluster 1 in purple colour, and cluster 2 in mint green colour.

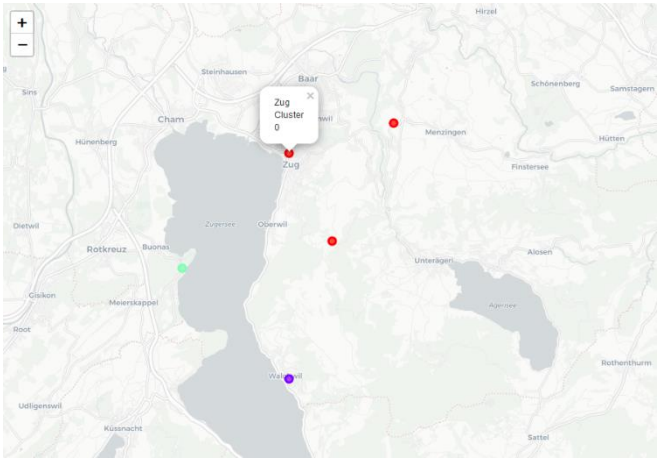

Figure 4 The most expensive gemeindens (clustered)

	Gemeinden	Annual rent / m^2 [CHF]	Cluster
1	Risch	660	2
2	Zugerberg	535	0
3	Edlibach	388	0
4	Zug	368	0
5	Walchwil	370	1

Table 2 The most expensive gemeindens (clustered)

5.1 Discussion

	Gemeinden	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Edlibach	Swiss Restaurant	Italian Restaurant	Outdoors & Recreation	Bakery	Moving Target
1	Risch	Lake	Cheese Shop	Hotel	Beach	Restaurant
2	Walchwil	Train Station	Grocery Store	Restaurant	Wine Shop	Cocktail Bar
3	Zug	Swiss Restaurant	Hotel	Restaurant	Supermarket	Italian Restaurant
4	Zugerberg	Swiss Restaurant	Hotel	Trail	Cable Car	Restaurant

Figure 5 The most common venues within the 5 most expensive gemeindens

Review showed that no wellness & spa venues are present in the 5 most expensive gemeindens of the kanton of Zug. This represents a great opportunity and high potential areas to open new centres as there is very little to no competition from existing venues.

As observations noted from the Figure 4, most of the leisure locations are concentrated in the area of Risch (also, the most expensive from the kanton – CHF660), with Lake, Hotel, and Beach indicated as the 1st, 2nd, and 4th most common venue of the locality. It looks like people are drawn to that location, which is good, but it could also be a place of high indirect competition, lowering the interest in the business (considering other 'attractions').

Cluster 0, consisting of 3 gemeindens (Zugerberg, Edlibach, Zug) can be classified as Gastronomy zone, since over 50% of venues are restaurants or food-related places. Which suggests, that great number of people tend to be agglomerated around, giving a good potential population of customers. On top of that, the gemeindens are centrally located in their respective towns.

Lastly, cluster 1 of Walchwil present the greatest variety of venues within, ranging from a train station (as the 1st most common) to a wine shop. Spa centres are likely to suffer from the location being quite isolated.

The complete notebook, with all codes and step by step solutions, can be found under:
https://github.com/MikolajWlazlo/Coursera_Capstone/blob/master/IBM%20Capstone%20Project.ipynb

From another perspective, the results also show that good public transport is present, which might promote, with good marketing, people travelling to use centre's services.

Therefore, this project recommends investors to capitalize on these findings to open new wellness & spa chain in neighbourhoods in cluster 0 with little to no competition coming only from hotel venues. Investors with unique selling propositions to stand out from the competition can also open a new centre in neighbourhoods in cluster 1 with slightly poorer location and not perfectly situated. Finally, investors are advised to avoid neighbourhoods in cluster 2 which have high concentration of leisure venues, and suffering from indirect competition might occur.

6. Conclusion

In this project, the process of identifying the business problem, specifying the data required, extracting and preparing the data, performing machine learning by clustering the data into 3 clusters based on their characteristics, and lastly providing recommendations to the relevant stakeholders, i.e. investors, regarding the best locations to open a luxury wellness & spa chain, was covered. All that was used to answer the business question raised in the introduction section of the project.

The answer proposed based on the findings is quite straightforward, but not quite so simple:
The neighbourhoods in cluster 0 are the most preferred locations to open a wellness & spa chain.

The findings of this project will help the investors to capitalize on the opportunities of high potential locations, while avoiding less potent areas in their decisions to open a new business.

6.1 Limitations & Suggestions for Future Research

In this project, only one factor was considered – highest annual rent per m^2.

Clearly, there are other factors such as population and income of residents, that could influence the location decision of a new wellness & spa centre. Future research could devise a methodology to estimate such data, and to be used in the clustering algorithm to determine the preferred locations considering more details about the localities.

Further investigation on the actual income information of inhabitants of particular region, and comparison against the average cost of living in the area, would result in an approximate overview of the potential savings/money-that-can-be-spent.

On top of that, investigating the no. of searches for the wellness & spa from the IP associated with particular region to define the potential interest of the locals in named business, could be an important insight.

The complete notebook, with all codes and step by step solutions, can be found under:
https://github.com/MikolajWlazlo/Coursera_Capstone/blob/master/IBM%20Capstone%20Project.ipynb