Use of Open data to prepare the tourist trip to Munich

Tourist expectations for Munich

Data sources

Datasources

- All Open data datasouces was from https://www.opengov-muenchen.de
- Additionally Foursquare API was used for data enrichment
- Partly Google Maps

Loading data issues

- no synchronisation even with splitters in the file. Three types of options was found ', ' or ' ; ' or '\t'.
- some datasources from open data in Munich have both *csv* and *kml* format.
- no syntchronisation among majority of field names

Data Processing and Analysis

Source data examples

- Example of data with coordinates

```
# Public Wifi
# https://www.opengov-muenchen.de/dataset/m-wl

Public_WIFI_URL = 'https://www.opengov-muenche
wifi_data = pd.read_csv(Public_WIFI_URL)
print(len(wifi_data))
wifi_data.head()
```

21

	Platz	Acess Point	latitude	longitude
0	Odeonsplatz	AP_1	48.142361	11.577892
1	Odeonsplatz	AP_2	48.142429	11.577247
2	Odeonsplatz	AP_4	48.142609	11.578021
3	Sendlinger Tor	AP_1	48.133764	11.566767
4	Sendlinger Tor	AP_2	48.133906	11.567297

- Example of statistical data with Beer prices at Oktoberfest

```
# Oktoberfest data inkluding beer and fried chiken price over the years
# https://www.opengov-muenchen.de/dataset/oktoberfest/resource/e0f664cf-6dd9-474

Oktoberfest_URL = 'https://www.opengov-muenchen.de/dataset/8d6c8251-7956-4f92-8c
oktoberfest_data = pd.read_csv(Oktoberfest_URL)
print(len(oktoberfest_data))
oktoberfest_data.head()
```

34

	jahr	dauer	besucher_gesamt	besucher_tag	bier_preis	bier_konsum	hendl_preis	hendl_konsum
0	1985	16	7.1	444	3.20	54541	4.77	629520
1	1986	16	6.7	419	3.30	53807	3.92	698137
2	1987	16	6.5	406	3.37	51842	3.98	732859
3	1988	16	5.7	356	3.45	50951	4.19	720139
4	1989	16	6.2	388	3.60	51241	4.22	775674

Museum data enrichment

```
:
```

```
# Feeding Museum addresses and trying to get coordinates

museum_addresses = pd.DataFrame({'POI_Name': ['Alte Pinakothek', 'Bayerisches Nationalmuseum','Deutsches Museum - M
                                 'POI_Address': ['Barer Str. 27, 80333 München', 'Prinzregentenstraße 3, 80538 Münch
                                 })
museum_addresses["POI_Type"] = "Museum"
museum_addresses["latitude"] = np.nan
museum_addresses["longitude"] = np.nan
#columns = ["POI_Type", "POI_Name", "POI_Address", "latitude", "longitude"]
# Collecting coordinates
for index, row in museum_addresses.iterrows():

    address = row['POI_Address']
    location = geolocator.geocode(address)
    latitude = location.latitude
    longitude = location.longitude
    print(row['POI_Name'], ": ", latitude, longitude)
#    time.sleep(7)
    museum_addresses.at[index, "latitude"] = latitude
    museum_addresses.at[index, "longitude"] = longitude



museum_addresses
poi_locations = museum_addresses.copy()
poi_locations
```

```
Alte Pinakothek :  48.1482861 11.569970904087043
Bayerisches Nationalmuseum :  48.1433855 11.5911498
Deutsches Museum - Museumsinsel :  48.13003865 11.582888488551909
Deutsches Museum - Verkehrszentrum  :  48.1325395 11.5429532
Münchner Stadtmuseum :  48.1351603 11.572607585081988
Museum Brandhorst :  48.1476318 11.57385874695941
Museum Mensch und Natur :  48.15908585 11.51166615
Neue Pinakothek :  48.1499278 11.570937445883057
Pinakothek der Moderne :  48.14716025 11.572227101401982
Schackgalerie :  48.14250955 11.593434725593049
Städtische Galerie im Lenbachhaus :  48.1466458 11.5633122
48.1358538 11.5495032
```

Data aggregation

Adding POIs near Isar River

```
']:
```

```
1  isar_data.head()
2  #poi_locations = museum_addresses.copy()
3
4  for index, row in isar_data.iterrows():
5
6      address = "Unknown, hint: " + str(row['description'])
7      latitude = row['latitude']
8      longitude = row['longitude']
9      poi_name = row['title']
10     #print(poi_name,  ": ", latitude, longitude, address)
11     poi_locations= poi_locations.append([{"POI_Type": "POIs Near south Isar", "POI_Name": poi_name, "POI_Address":
12
13 if len(poi_locations) >5:
14     print(poi_locations[len(poi_locations)-5:len(poi_locations)-1])
15
```

```
             POI_Name                                        POI_Address  \
0     Fräulein Grüneis  Unknown, hint: Sortiment: verschiedene Biersor...
0           Milchhäusl  Unknown, hint: Sortiment: zu 100 Prozent aus Ö...
0         Bussis Kiosk  Unknown, hint: Sortiment: Neben Bier kann man ...
0  Kiosk Schinderstadl  Unknown, hint: Öffnungszeiten: bei schönem Wet...

              POI_Type   latitude  longitude
0  POIs Near south Isar  48.143652  11.588752
0  POIs Near south Isar  48.149779  11.585437
0  POIs Near south Isar  48.160280  11.591665
0  POIs Near south Isar  48.107352  11.552634
```

Discovering data with simple techniques

Checking WCs with Douche and Baby Changing rooms

```
# Toilets with douche
wc_douche_data= wc_data[wc_data.duschen == True].copy()
print(len(wc_data))
print(len(wc_douche_data))
wc_douche_data.head()

# Probably No chance :-(
```

```
134
0
```

address_id	address_organisation	address_organisationsbereich	bezeichnung	address_strasse	address_hausnummer	address_postleitzahl	address_ort	latitud

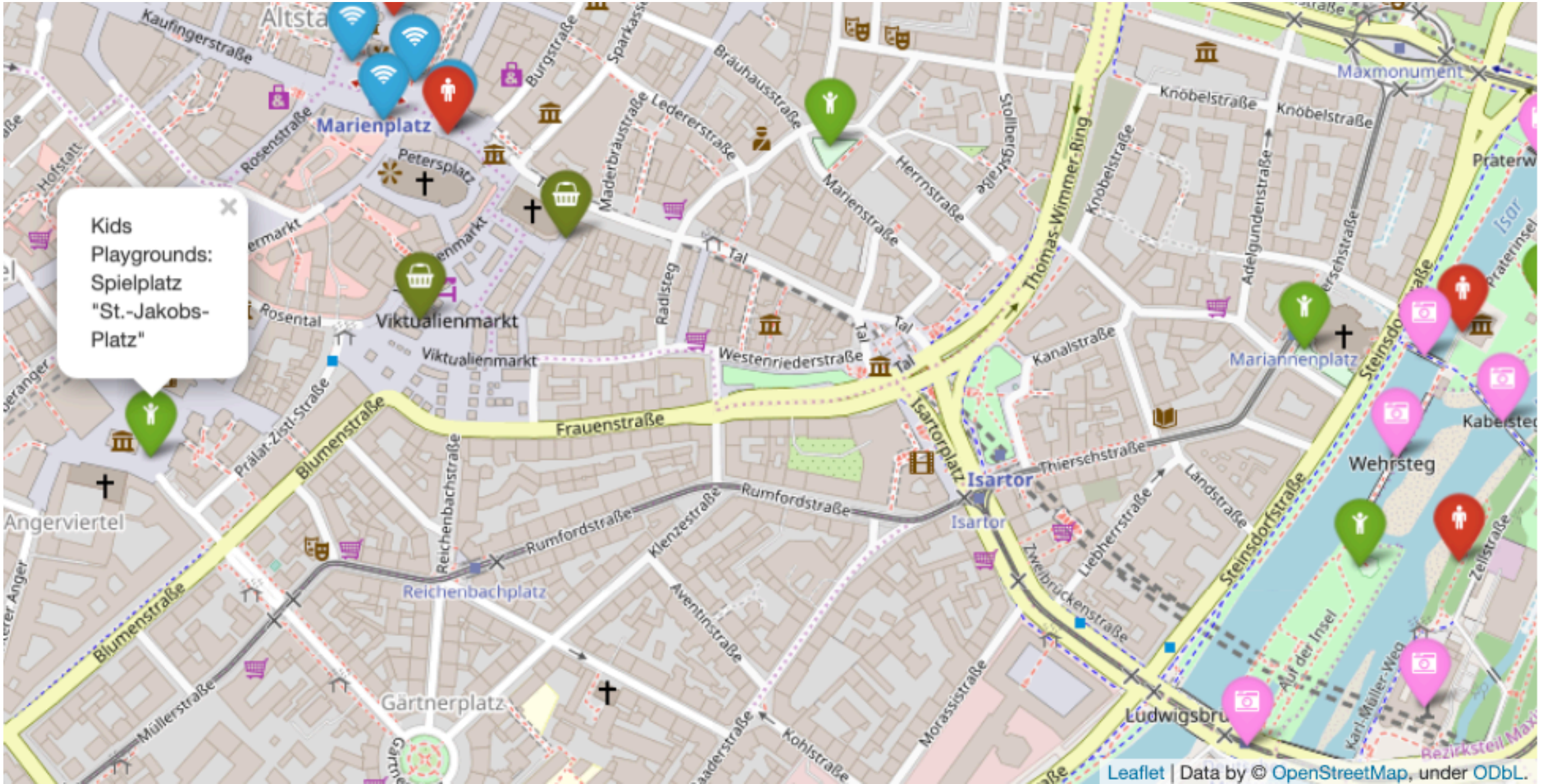
```
# baby changing rooms


wc_bcr_data= wc_data[wc_data.wickelraeume == True].copy()
print(len(wc_data))
print(len(wc_bcr_data))
wc_bcr_data.head()

```

```
134
3
```

	address_id	address_organisation	address_organisationsbereich	bezeichnung	address_strasse	address_hausnummer	address_postleitzahl	address_ort	la
82	1065675	Landeshauptstadt München	MTG	Öffentliche Toilette "Marienplatz (im S- u. U-...	NaN	NaN	NaN	NaN	48.1
83	1065677	Landeshauptstadt München	MTG	Öffentliche Toilette "Marienplatz Neues Rathau...	NaN	NaN	NaN	NaN	48.1
88	1065687	Landeshauptstadt München	MTG	Öffentliche Toilette "Münchner Freiheit (im U-...	NaN	NaN	NaN	NaN	48.1

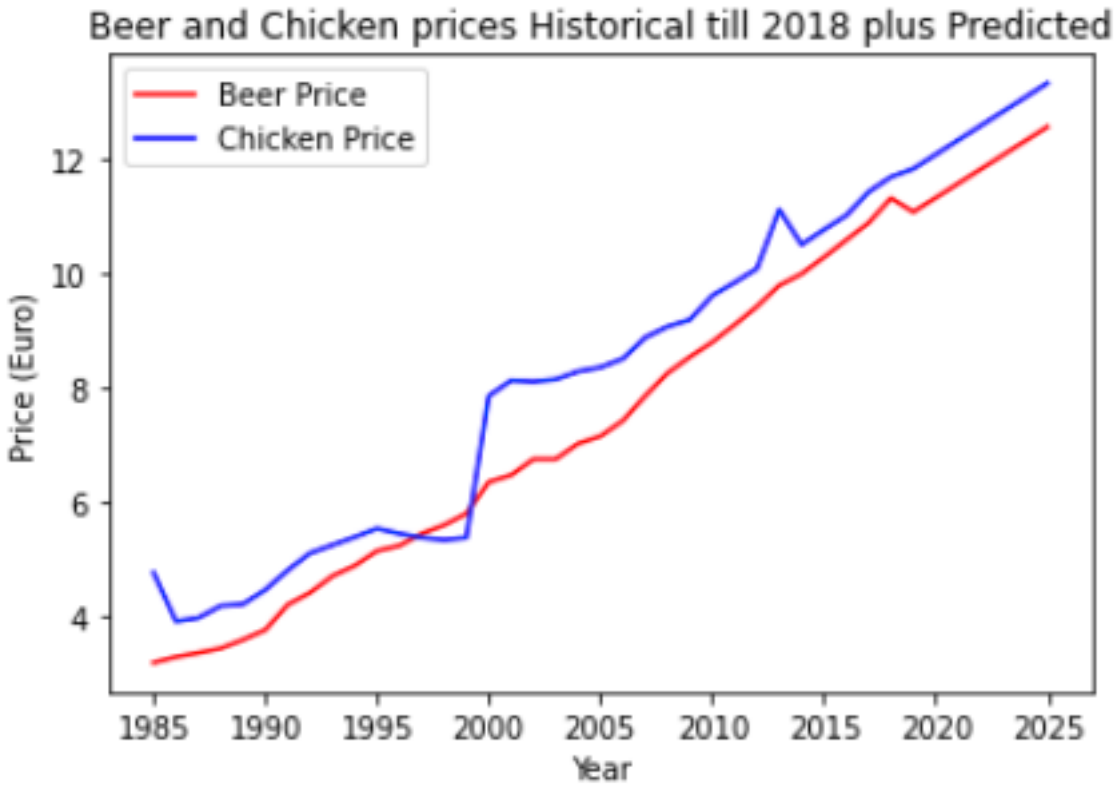
Visualizing the data with circles using Folium

Visualizing the data with icons using Folium



Predicting beer and chicken prices for the future

- Visualization



- Numerical data with predictions

```
predictions.head(10)
```

	Year	Mass Bier Price	Roasted Chicken Price
0	2019	11.058984	11.812620
1	2020	11.306455	12.061459
2	2021	11.553927	12.310298
3	2022	11.801398	12.559137
4	2023	12.048869	12.807976
5	2024	12.296341	13.056814
6	2025	12.543812	13.305653

Analysing potentially less and most loaded months for museums from open dataset based on the nubmer of visitors

- Seems like september is the best month to visit museums

	museum	month_easy	visitors_min	month_busy	visitors_max
0	Alte Pinakothek	9	319819.0	8	474533.0
1	Bayerisches Nationalmuseum	9	109722.0	12	308245.0
2	Deutsches Museum - Museumsinsel	11	1100457.0	8	2330174.0
3	Deutsches Museum - Verkehrszentrum	9	67830.0	8	115807.0
4	Museum Brandhorst	4	84969.0	10	153702.0
5	Museum Mensch und Natur	12	84070.0	3	182075.0
6	Münchner Stadtmuseum	8	139913.0	3	290504.0
7	Neue Pinakothek	9	232910.0	8	394217.0
8	Pinakothek der Moderne	6	360013.0	10	685313.0
9	Schackgalerie	9	8174.0	10	23256.0
10	Städtische Galerie im Lenbachhaus	9	230150.0	1	484913.0

What surpised

- The number of playgrounds is over seven hundreds and it's a huge number.
- Very limited data on Public WiFi Areas
- Good information about POIs around Isar river
- Some interesting data loaded once and not updates for years

Discussions and Lessons learned

Lessons learned

- Open data is a valuable source for Data Scientiest
- By the nature data is not well organized and synchronized
  - Expect extra time for preprocessing, correlation and enrichment
- Tens of data sources can be applied for different application fields
- We tested data which is potentially interesting for travellers and it’s valuable

Conclusion

Conclusion

- *Open Data* published by Munich helped us to solve a task.
- Several techniques were successfully used to discover the data
  - *simple data analysis* to determine extra futures of Public WCs
  - *visualisation* to find different types of places, like kids playgrounds, markets, museums, sightseeing points, WiFi hotspots on the interactive Munich map
  - *linear regression from sklearn library* for beer and chicken price prediction
  - *statistical analysis* to determine the best time to visit museums.
- All of these can help travellers to improve their standard and unusual travel experience in this wonderful city. Hope that ideas of this project demonstrate several data science techniques and can help different categories of tourist travellers to find places they are looking for.