CP-Final- Week02

August 4, 2021

1 Capstone Project -Week 02 - FINAL

1.1 IBM Data Science Certificate

2 50 Munich Restaurants from Yelp Business Search Database

3 Data Science Analysis

4 Week01 - you will required to submit the following:

4.1 1. A full report consisting of all of the following components (15 marks):

4.1.1 1.1 Introduction - where you discuss the business problem and who would be interested in this project.

4.1.2 1.2 Data - where you describe the data that will be used to solve the problem and the source of the data.

See Week 01 - Capstone Project Assignment for Introduction and Data ### 1.3 Methodology section - which represents the main component of the report where you discuss and describe any exploratory data analysis that you did, any inferential statistical testing that you performed, if any, and what machine learning were used and why. Source code and data visualization of this assignment which covers the methodology section ### 1.4 Results - section where you discuss the results. ### 1.5 Discussion - section where you discuss any observations you noted and any recommendations you can make based on the results. ### 1.6 Conclusion - section where you conclude the report. 1.4 / 1.5 / 1.6 are all listed at the end of this Jupyter Notebook - following the source code and data visualization of this assignment which covers the methodology section

4.2 2. A link to your Notebook on your Github repository pushed showing your code. (15 marks)

4.3 3. Your choice of a presentation or blogpost. (10 marks)

PDF File - Jupyter Notebook Export of this Report

Here are examples of previous outstanding submissions that should give you an idea of what your report would look like, what your notebook would look like in terms of clean, clear, and well-commented code, and what your presentation would look like or your blogpost would look like:

[2]:
```
# Import all relevant Python libraries

!pip install -U numpy
!pip install -U pandas
!pip install -U scipy
!pip install -U scikit-learn
!pip install -U imbalanced-learn

import pandas as pd, numpy as np        # pd - for manipulating data, open .csv␣
 ↪and .json file/ np - for math operation
import sys, requests, lxml, re, json, urllib # library to handle requests,␣
 ↪lxml, json url
import time

! pip install yelp
! pip install yelpapi                      # install yelp api

! pip install matplotlib
import matplotlib.pyplot as plt       # to set graph, figsize
import matplotlib.cm as cm            # for handling utilities in color map
import matplotlib.colors as colors    # to generate colors

from sklearn.cluster import KMeans    # for generating cluster

!pip install beautifulsoup4
from bs4 import BeautifulSoup               # for scapping wikipedia wesite

# tranforming json file into a pandas dataframe library
from pandas.io.json import json_normalize

! pip install seaborn
import seaborn as sns

! pip install plotly
! pip install chart_studio
! pip install plotly --upgrade
import chart_studio
# chart_studio.tools.set_credentials_file(username='DemoAccount',␣
 ↪api_key='lr1c37zw81')

import plotly
import chart_studio.plotly as py
# plotly.tools.set_credentials_file(username='', api_key='')
import plotly.tools as tls
from plotly.graph_objs import *
%matplotlib inline
```

```
# libraries for displaying images
from IPython.display import Image
from IPython.core.display import HTML

!pip install geopy
from geopy.geocoders import Nominatim          # module to convert an address␣
 ↪into latitude and longitude values

! pip install folium==0.12.0
import folium                                  # generating maps

! pip install glom
from glom import glom # library that allows us to use . notation to access␣
 ↪property from a deeply nested object
```

```
Requirement already up-to-date: numpy in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (1.19.5)
Requirement already up-to-date: pandas in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (1.1.5)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.7.3 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from pandas)
(2.8.1)
Requirement already satisfied, skipping upgrade: pytz>=2017.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from pandas)
(2021.1)
Requirement already satisfied, skipping upgrade: numpy>=1.15.4 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from pandas)
(1.19.5)
Requirement already satisfied, skipping upgrade: six>=1.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from python-
dateutil>=2.7.3->pandas) (1.15.0)
Requirement already up-to-date: scipy in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (1.5.4)
Requirement already satisfied, skipping upgrade: numpy>=1.14.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from scipy)
(1.19.5)
Requirement already up-to-date: scikit-learn in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (0.24.2)
Requirement already satisfied, skipping upgrade: threadpoolctl>=2.0.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from scikit-
learn) (2.2.0)
Requirement already satisfied, skipping upgrade: scipy>=0.19.1 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from scikit-
learn) (1.5.4)
Requirement already satisfied, skipping upgrade: numpy>=1.13.3 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from scikit-
learn) (1.19.5)
```

```
Requirement already satisfied, skipping upgrade: joblib>=0.11 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from scikit-
learn) (1.0.1)
Requirement already up-to-date: imbalanced-learn in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (0.8.0)
Requirement already satisfied, skipping upgrade: scikit-learn>=0.24 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from imbalanced-
learn) (0.24.2)
Requirement already satisfied, skipping upgrade: scipy>=0.19.1 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from imbalanced-
learn) (1.5.4)
Requirement already satisfied, skipping upgrade: numpy>=1.13.3 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from imbalanced-
learn) (1.19.5)
Requirement already satisfied, skipping upgrade: joblib>=0.11 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from imbalanced-
learn) (1.0.1)
Requirement already satisfied, skipping upgrade: threadpoolctl>=2.0.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from scikit-
learn>=0.24->imbalanced-learn) (2.2.0)
Requirement already satisfied: yelp in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (1.0.2)
Requirement already satisfied: oauth2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from yelp)
(1.9.0.post1)
Requirement already satisfied: six in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from yelp)
(1.15.0)
Requirement already satisfied: httplib2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from yelp)
(0.19.1)
Requirement already satisfied: pyparsing<3,>=2.4.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
httplib2->yelp) (2.4.7)
Requirement already satisfied: yelpapi in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (2.4.0)
Requirement already satisfied: requests in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from yelpapi)
(2.25.1)
Requirement already satisfied: idna<3,>=2.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->yelpapi) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->yelpapi) (1.26.6)
Requirement already satisfied: certifi>=2017.4.17 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->yelpapi) (2021.5.30)
```

```
Requirement already satisfied: chardet<5,>=3.0.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->yelpapi) (4.0.0)
Requirement already satisfied: matplotlib in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (3.3.4)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from matplotlib)
(2.4.7)
Requirement already satisfied: pillow>=6.2.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from matplotlib)
(8.3.1)
Requirement already satisfied: numpy>=1.15 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from matplotlib)
(1.19.5)
Requirement already satisfied: python-dateutil>=2.1 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from matplotlib)
(2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from matplotlib)
(1.3.1)
Requirement already satisfied: cycler>=0.10 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/cycler-0.10.0-py3.6.egg (from matplotlib) (0.10.0)
Requirement already satisfied: six>=1.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from python-
dateutil>=2.1->matplotlib) (1.15.0)
Requirement already satisfied: beautifulsoup4 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (4.9.3)
Requirement already satisfied: soupsieve>1.2; python_version >= "3.0" in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
beautifulsoup4) (2.2.1)
Requirement already satisfied: seaborn in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (0.9.0)
Requirement already satisfied: pandas>=0.15.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from seaborn)
(1.1.5)
Requirement already satisfied: numpy>=1.9.3 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from seaborn)
(1.19.5)
Requirement already satisfied: matplotlib>=1.4.3 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from seaborn)
(3.3.4)
Requirement already satisfied: scipy>=0.14.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from seaborn)
(1.5.4)
Requirement already satisfied: python-dateutil>=2.7.3 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
pandas>=0.15.2->seaborn) (2.8.1)
```

```
Requirement already satisfied: pytz>=2017.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
pandas>=0.15.2->seaborn) (2021.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
matplotlib>=1.4.3->seaborn) (2.4.7)
Requirement already satisfied: pillow>=6.2.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
matplotlib>=1.4.3->seaborn) (8.3.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
matplotlib>=1.4.3->seaborn) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/cycler-0.10.0-py3.6.egg (from matplotlib>=1.4.3->seaborn) (0.10.0)
Requirement already satisfied: six>=1.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from python-
dateutil>=2.7.3->pandas>=0.15.2->seaborn) (1.15.0)
Requirement already satisfied: plotly in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (5.1.0)
Requirement already satisfied: six in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from plotly)
(1.15.0)
Requirement already satisfied: tenacity>=6.2.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from plotly)
(8.0.1)
Requirement already satisfied: chart_studio in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (1.1.0)
Requirement already satisfied: plotly in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
chart_studio) (5.1.0)
Requirement already satisfied: six in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
chart_studio) (1.15.0)
Requirement already satisfied: retrying>=1.3.3 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
chart_studio) (1.3.3)
Requirement already satisfied: requests in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
chart_studio) (2.25.1)
Requirement already satisfied: tenacity>=6.2.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
plotly->chart_studio) (8.0.1)
Requirement already satisfied: idna<3,>=2.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->chart_studio) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
```

```
requests->chart_studio) (1.26.6)
Requirement already satisfied: certifi>=2017.4.17 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->chart_studio) (2021.5.30)
Requirement already satisfied: chardet<5,>=3.0.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->chart_studio) (4.0.0)
Requirement already up-to-date: plotly in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (5.1.0)
Requirement already satisfied, skipping upgrade: six in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from plotly)
(1.15.0)
Requirement already satisfied, skipping upgrade: tenacity>=6.2.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from plotly)
(8.0.1)
Requirement already satisfied: geopy in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (2.2.0)
Requirement already satisfied: geographiclib<2,>=1.49 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from geopy)
(1.52)
Collecting folium==0.12.0
  Downloading https://files.pythonhosted.org/packages/e6/7a/e227526f4a82e7
52aa0352ca63d93166ec00ac0fdb63fa7066f94208cade/folium-0.12.0-py2.py3-none-
any.whl (94kB)
     |                  | 102kB 24.8MB/s ta 0:00:01
Requirement already satisfied: branca>=0.3.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
folium==0.12.0) (0.4.2)
Requirement already satisfied: requests in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
folium==0.12.0) (2.25.1)
Requirement already satisfied: numpy in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
folium==0.12.0) (1.19.5)
Requirement already satisfied: jinja2>=2.9 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
folium==0.12.0) (3.0.1)
Requirement already satisfied: idna<3,>=2.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->folium==0.12.0) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->folium==0.12.0) (1.26.6)
Requirement already satisfied: certifi>=2017.4.17 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->folium==0.12.0) (2021.5.30)
Requirement already satisfied: chardet<5,>=3.0.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
```

```
requests->folium==0.12.0) (4.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
jinja2>=2.9->folium==0.12.0) (2.0.1)
Installing collected packages: folium
  Found existing installation: folium 0.5.0
    Uninstalling folium-0.5.0:
      Successfully uninstalled folium-0.5.0
Successfully installed folium-0.12.0
Requirement already satisfied: glom in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (20.11.0)
Requirement already satisfied: face>=20.1.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from glom)
(20.1.1)
Requirement already satisfied: boltons>=19.3.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from glom)
(21.0.0)
Requirement already satisfied: attrs in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from glom)
(21.2.0)
```

[3]:
```
# define yelp api key, authorization data
api_key='FM3e75vT6xALNvy1nHL8CkmZCTNuVd6s6bGyj47L_IHcLkmqjxk4xQ3yEToUkfC8uzwC-35U6e3VsPaE0hZu_
headers = {'Authorization': 'Bearer %s' % api_key}

url='https://api.yelp.com/v3/businesses/search'

# yelp api, businesses search - params, specifies relevant parameters - term,␣
 ↪location, limit
# params term - restaurant, bars, cafes, italian, japan, china, location city,␣
 ↪limit 5, 10, 20 max 50
# limit is 50 restaurant entries, specified by yelp api
params = {'term':'restaurant','location':'Munich', 'limit':'50'}

# Making a get request to the API
req=requests.get(url, params=params, headers=headers)

# data json parsing
data = json.loads(req.text)


# proceed only if the status code is 200
print('Proceed only if the Request Object status code is 200', '\n')
print('The Request Object status code is {}'.format(req.status_code), '\n')
print('Data Type Request Object', '\n')
print(type(req), '\n') # print Data Type Request Object
print('Data Type TEXT Request Object - STRING', '\n')
```

```
print(type(req.text), '\n') # printing text from req response
print('Data Type JSON Request Object- METHOD', '\n')
print(type(req.json), '\n') # printing json from req response
print('Data Type JSON parsing Request Object- DICTIONARY', '\n')
print(type(data),  '\n')
```

```
Proceed only if the Request Object status code is 200

The Request Object status code is 200

Data Type Request Object

<class 'requests.models.Response'>

Data Type TEXT Request Object - STRING

<class 'str'>

Data Type JSON Request Object- METHOD

<class 'method'>

Data Type JSON parsing Request Object- DICTIONARY

<class 'dict'>
```

[4]:
```
# declare panda datframe, parse request object to json - businesses is data␣
 ↪input from yelp api
df2 = pd.DataFrame.from_dict(req.json()['businesses'])

# display dimension, column names and column types for df2
print(len(df2), '\n') #Print how many rows
print(df2.columns, '\n') #Print column names
print(df2.dtypes, '\n') #Print column types

# Display df2 - notice columns are multi nested and unsorted
df2.head()
```

```
50

Index(['id', 'alias', 'name', 'image_url', 'is_closed', 'url', 'review_count',
       'categories', 'rating', 'coordinates', 'transactions', 'price',
       'location', 'phone', 'display_phone', 'distance'],
      dtype='object')

id                object
```

```
alias             object
name              object
image_url         object
is_closed           bool
url               object
review_count       int64
categories        object
rating           float64
coordinates       object
transactions      object
price             object
location          object
phone             object
display_phone     object
distance         float64
dtype: object
```

[4]:

	id	alias	name	image_url	is_closed	url	review_count	categories	rating	coordinates	transactions	price
0	njLmw1IVbtOb2m-4TNr8ow	augustiner-keller-münchen-2	Augustiner-Keller	https://s3-media3.fl.yelpcdn.com/bphoto/5VtnFb…	False	https://www.yelp.com/biz/augustiner-keller-m%C…	633	[{'alias': 'bavarian', 'title': 'Bavarian'}, {…	4.0			
1	SbJxG5IbPsbzGcBl8ZL8uA	marbella-münchen	Marbella	https://s3-media2.fl.yelpcdn.com/bphoto/JwNNsG…	False	https://www.yelp.com/biz/marbella-m%C3%BCnchen…	38	[{'alias': 'spanish', 'title': 'Spanish'}, {'a…	4.5			
2	1b1SH21SQt_rY92gEZDKEg	neuhauser-augustiner-münchen	Neuhauser Augustiner	https://s3-media3.fl.yelpcdn.com/bphoto/zGP5G0…	False	https://www.yelp.com/biz/neuhauser-augustiner-…	36	[{'alias': 'german', 'title': 'German'}]	4.5			
3	oNF_jG_WQP0WZ7BBZ-ylXg	yee-chino-restaurant-münchen	Yee Chino Restaurant	https://s3-media2.fl.yelpcdn.com/bphoto/x9Y7Bk…	False	https://www.yelp.com/biz/yee-chino-restaurant-…	49	[{'alias': 'panasian', 'title': 'Pan Asian'}]	4.5			
4	wOxHVpGYaqeXgrVLOgaAPQ	an-an-münchen	An An	https://s3-media2.fl.yelpcdn.com/bphoto/_ILb2u…	False	https://www.yelp.com/biz/an-an-m%C3%BCnchen?ad…	12	[{'alias': 'vietnamese', 'title': 'Vietnamese'}]	5.0			

```
0       {'latitude': 48.1435, 'longitude': 11.55195}           []    €€
1  {'latitude': 48.1538833, 'longitude': 11.5416787}           []    €€
2        {'latitude': 48.1592, 'longitude': 11.5406}           []    €€
3  {'latitude': 48.1632208, 'longitude': 11.5436741}           []    €€
4       {'latitude': 48.14902, 'longitude': 11.5429}           []   NaN

                                          location          phone  \
0  {'address1': 'Arnulfstr. 52', 'address2': None…    +4989594393
1  {'address1': 'Horemansstr. 30', 'address2': ''…  +498912779753
2  {'address1': 'Hübnerstr. 23', 'address2': None…   +49891202130
3  {'address1': 'Helene-Weber-Allee 19', 'address…  +498915988587
4  {'address1': 'Elvirastr. 12', 'address2': None…  +498955286459

     display_phone     distance
0    +49 89 594393  1440.274028
1  +49 89 12779753   129.674074
2   +49 89 1202130   497.766512
3  +49 89 15988587   928.266080
4  +49 89 55286459   654.180864
```

[72]:
```
# normalize json to new dataframe - d1
d1 = pd.json_normalize(data["businesses"])
# d2 dataframe, with appropriate column names, json subcolumns coordinates.
 ↪latitude / coordinates.longitude / location.display_address
d2 = d1[['name','location.display_address','phone', 'categories', 'rating',␣
 ↪'review_count', 'coordinates.latitude','coordinates.longitude', 'price']]
# rename columns
d2.rename({'location.display_address': 'address', 'coordinates.latitude':␣
 ↪'latitude',
         'coordinates.longitude': 'longitude'}, axis=1, inplace=True)
# maximize column width to display appropriately nested json data in column -␣
 ↪categories
pd.set_option("display.max_colwidth", -1)

d2.head()
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/pandas/core/frame.py:4308: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:9: FutureWarning:
```

```
Passing a negative integer is deprecated in version 1.0 and will not be
supported in future version. Instead, use None to not limit the column width.
```

[72]:
```
                  name                                          address  \
0  Augustiner-Keller     [Arnulfstr. 52, 80335 Munich, Germany]
1  Marbella              [Horemansstr. 30, 80636 Munich, Germany]
2  Neuhauser Augustiner  [Hübnerstr. 23, 80637 Munich, Germany]
3  Yee Chino Restaurant  [Helene-Weber-Allee 19, 80637 Munich, Germany]
4  An An                 [Elvirastr. 12, 80636 Munich, Germany]

           phone  \
0  +4989594393
1  +498912779753
2  +49891202130
3  +498915988587
4  +498955286459

                categories  \
0  [{'alias': 'bavarian', 'title': 'Bavarian'}, {'alias': 'beergarden', 'title':
'Beer Garden'}]
1  [{'alias': 'spanish', 'title': 'Spanish'}, {'alias': 'tapasmallplates',
'title': 'Tapas/Small Plates'}]
2  [{'alias': 'german', 'title': 'German'}]
3  [{'alias': 'panasian', 'title': 'Pan Asian'}]
4  [{'alias': 'vietnamese', 'title': 'Vietnamese'}]

   rating  review_count   latitude  longitude price
0  4.0     633          48.143500  11.551950  €€
1  4.5     38           48.153883  11.541679  €€
2  4.5     36           48.159200  11.540600  €€
3  4.5     49           48.163221  11.543674  €€
4  5.0     12           48.149020  11.542900  NaN
```

[74]:
```
# categories column, contains nested json data - we need to extract that and␣
 ↪separate it to new columns
# d3 new df that contains only categories column
d3 = d2
d3['cat'] = d2['categories']

# notice nested json data in categories column - which hasn't been normalized/
 ↪flattened by previous json normailization
# print(d3.columns, '\n')
d3.head()
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:4: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[74]:

	name	address	phone	categories	rating	review_count	latitude	longitude	price	cat
0	Augustiner-Keller	[Arnulfstr. 52, 80335 Munich, Germany]	+4989594393	[{'alias': 'bavarian', 'title': 'Bavarian'}, {'alias': 'beergarden', 'title': 'Beer Garden'}]	4.0	633	48.143500	11.551950	€€	[{'alias': 'bavarian', 'title': 'Bavarian'}, {'alias': 'beergarden', 'title': 'Beer Garden'}]
1	Marbella	[Horemansstr. 30, 80636 Munich, Germany]	+498912779753	[{'alias': 'spanish', 'title': 'Spanish'}, {'alias': 'tapasmallplates', 'title': 'Tapas/Small Plates'}]	4.5	38	48.153883	11.541679	€€	[{'alias': 'spanish', 'title': 'Spanish'}, {'alias': 'tapasmallplates', 'title': 'Tapas/Small Plates'}]
2	Neuhauser Augustiner	[Hübnerstr. 23, 80637 Munich, Germany]	+49891202130	[{'alias': 'german', 'title': 'German'}]	4.5	36	48.159200	11.540600	€€	[{'alias': 'german', 'title': 'German'}]
3	Yee Chino Restaurant	[Helene-Weber-Allee 19, 80637 Munich, Germany]	+498915988587	[{'alias': 'panasian', 'title': 'Pan Asian'}]	4.5	49	48.163221	11.543674	€€	[{'alias': 'panasian', 'title': 'Pan Asian'}]
4	An An	[Elvirastr. 12, 80636 Munich, Germany]	+498955286459	[{'alias': 'vietnamese', 'title': 'Vietnamese'}]	5.0	12	48.149020	11.542900	NaN	[{'alias': 'vietnamese', 'title': 'Vietnamese'}]

[75]:
```
# flatten categories/cat column - declare new column al1, al2, al3 that contain␣
 ↪separated nested categories data
df1 = (
    pd.DataFrame(d3["cat"]
    .apply(pd.Series))
)
# name df1 columns
df1.columns = ['al1', 'al2', 'al3']

# check df1 dimension - 50 throughout this assignment - remember limit 50 -␣
 ↪yelp api
print(len(df1), '\n') #Print how many rows
print(df1.columns, '\n') #Print column names
print(df1.dtypes, '\n') #Print column types

# we have for df1, categories with same terms but also different - still nested␣
 ↪data in columns
df1.head(10)
```

```
50

Index(['al1', 'al2', 'al3'], dtype='object')

al1    object
al2    object
al3    object
dtype: object
```

[75]:

	al1	al2	al3
0	{'alias': 'bavarian', 'title': 'Bavarian'}	{'alias': 'beergarden', 'title': 'Beer Garden'}	NaN
1	{'alias': 'spanish', 'title': 'Spanish'}	{'alias': 'tapasmallplates', 'title': 'Tapas/Small Plates'}	NaN
2	{'alias': 'german', 'title': 'German'}	NaN	NaN
3	{'alias': 'panasian', 'title': 'Pan Asian'}	NaN	NaN
4	{'alias': 'vietnamese', 'title': 'Vietnamese'}	NaN	NaN
5	{'alias': 'mexican', 'title': 'Mexican'}		
6	{'alias': 'thai', 'title': 'Thai'}		
7	{'alias': 'italian', 'title': 'Italian'}		
8	{'alias': 'vietnamese', 'title': 'Vietnamese'}		
9	{'alias': 'italian', 'title': 'Italian'}		

```
5  {'alias': 'cocktailbars', 'title': 'Cocktail Bars'}          NaN
6  NaN                                                          NaN
7  NaN                                                          NaN
8  {'alias': 'wok', 'title': 'Wok'}                             NaN
9  NaN                                                          NaN
```

[8]:
```
# json flattening dataframe columns, al1, al2, al3 - new columns with nested data
json_struct = json.loads(df1.to_json(orient="records"))
df_flat = pd.json_normalize(json_struct) #use pd.io.json
# df1.head()
print(len(df_flat), '\n') # Print how many rows, check dimension 50
# display categories dataframe, notice similarites and differences, so entry0 Bavarian/Beergarden, entry1 Spanish/Tapas, entry2 Panasian...
df_flat.head(3)
```

```
50
```

[8]:

	al3	al1.alias	al1.title	al2.alias	al2.title	al2	\
0	NaN	bavarian	Bavarian	beergarden	Beer Garden	NaN	
1	NaN	spanish	Spanish	tapasmallplates	Tapas/Small Plates	NaN	
2	NaN	german	German	NaN	NaN	NaN	

	al3.alias	al3.title
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN

[9]:
```
# al1.title, al2.title -  columns with relevant data
df_flat2 = df_flat[['al1.title', 'al2.title', 'al3.title']]
# cat1, cat2, cat3 - 3 distinct categories to describe restaurants - mainly nationality and style
# rename to cat 1, cat2
df_flat2.columns = ['cat1', 'cat2', 'cat3']
df_flat2.insert(0, 'catflag', df_flat2.index + 0)


# catflag - new column with multiple conditions - to uniqly categroize restaurants -
conditions = [
    (df_flat2['cat1'] == 'Spanish'),
    (df_flat2['cat1'] == 'Bavarian'),
    (df_flat2['cat1'] == 'Pan Asian'),
    (df_flat2['cat2'] == 'Vietnamese')]
choices = ['Continental', 'Continental', 'Continental', 'Asian']
```

```
df_flat2['catflag'] = np.select(conditions, choices, default='Conditional Value␣
↪1')

df_flat2.head(2)
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:16: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[9]:

	catflag	cat1	cat2	cat3
0	Continental	Bavarian	Beer Garden	NaN
1	Continental	Spanish	Tapas/Small Plates	NaN

[94]:
```
# merge d2 with df_flat2
dff = pd.concat([d2, df_flat2], axis=1)
dff1 = dff[['cat1', 'cat2', 'cat3']]

count1 = dff1['cat1'].value_counts()
count2 = dff1['cat2'].value_counts()
count3 = dff1['cat3'].value_counts()

# check column df_flat2 size 50 matches
print('Number of Rows df', '\n')
print(len(df_flat2), '\n')

# Frequency of occurencies for cat1, cat2, cat3 - Categories Restaurants
# cat1 is nationality, cat2 is style, cat3 is type
print('Freq of occurencies cat1', '\n')
print(count1, '\n')
print('Freq of occurencies cat2', '\n')
print(count2, '\n')
print('Freq of occurencies cat3', '\n')
print(count3)

dff.head()
```

```
Number of Rows df

50
```

```
Freq of occurencies cat1

Italian               7
Bavarian              5
German                4
Greek                 4
Cafes                 4
Chinese               3
Vietnamese            3
Pan Asian             3
Barbeque              2
Thai                  2
Mexican               2
Cocktail Bars         1
Kebab                 1
Pancakes              1
Korean                1
Tapas Bars            1
Breakfast & Brunch    1
International         1
Vegan                 1
Ramen                 1
Spanish               1
Austrian              1
Name: cat1, dtype: int64

Freq of occurencies cat2

American (Traditional)    2
Steakhouses               2
Breakfast & Brunch        2
Dive Bars                 1
Cocktail Bars             1
Mediterranean             1
Tapas/Small Plates        1
Seafood                   1
Bavarian                  1
Brewpubs                  1
Wok                       1
Beer Garden               1
Smokehouse                1
Cafes                     1
Austrian                  1
Pizza                     1
Vegetarian                1
Name: cat2, dtype: int64

Freq of occurencies cat3
```

```
Gluten-Free           1
Gastropubs            1
Breakfast & Brunch    1
Burgers               1
Food Stands           1
Seafood               1
Name: cat3, dtype: int64
```

[94]:

	name	address	phone	categories	rating	review_count	latitude	longitude	price	cat
0	Augustiner-Keller	[Arnulfstr. 52, 80335 Munich, Germany]	+4989594393	[{'alias': 'bavarian', 'title': 'Bavarian'}, {'alias': 'beergarden', 'title': 'Beer Garden'}]	4.0	633	48.143500	11.551950	€€	[{'alias': 'bavarian', 'title': 'Bavarian'}, {'alias': 'beergarden', 'title': 'Beer Garden'}]
1	Marbella	[Horemansstr. 30, 80636 Munich, Germany]	+498912779753	[{'alias': 'spanish', 'title': 'Spanish'}, {'alias': 'tapasmallplates', 'title': 'Tapas/Small Plates'}]	4.5	38	48.153883	11.541679	€€	[{'alias': 'spanish', 'title': 'Spanish'}, {'alias': 'tapasmallplates', 'title': 'Tapas/Small Plates'}]
2	Neuhauser Augustiner	[Hübnerstr. 23, 80637 Munich, Germany]	+49891202130	[{'alias': 'german', 'title': 'German'}]	4.5	36	48.159200	11.540600	€€	[{'alias': 'german', 'title': 'German'}]
3	Yee Chino Restaurant	[Helene-Weber-Allee 19, 80637 Munich, Germany]	+498915988587	[{'alias': 'panasian', 'title': 'Pan Asian'}]	4.5	49	48.163221	11.543674	€€	[{'alias': 'panasian', 'title': 'Pan Asian'}]
4	An An	[Elvirastr. 12, 80636 Munich, Germany]	+498955286459	[{'alias': 'vietnamese', 'title': 'Vietnamese'}]	5.0	12	48.149020	11.542900	NaN	[{'alias': 'vietnamese', 'title': 'Vietnamese'}]

```
                catflag        cat1                cat2 cat3
0  Continental           Bavarian    Beer Garden         NaN
1  Continental           Spanish     Tapas/Small Plates  NaN
2  Conditional Value 1   German      NaN                 NaN
3  Continental           Pan Asian   NaN                 NaN
4  Conditional Value 1   Vietnamese  NaN                 NaN
```

[95]:
```
# dff is the final dataframe containing all relevant required columns to work␣
 ↪with data science tools
# drop categories column, to classify Munich Restaurants using appropriate␣
 ↪Python Data Science Tools
dff.drop('categories', inplace=True, axis=1)
dff.drop('cat', inplace=True, axis=1)
# insert incremental identifier column for data science tools - used later for␣
 ↪plotting data
dff.insert(0, 'ID', dff.index + 0)

# Replace NaN entries with empty spaces
dff = dff.fillna('')
# Sort dff Alphabetically - Column Name - Display all dff
dff.sort_values(by=['name'], ascending=True)
```

[95]:
```
     ID                           name  \
17   17  AMIGOS RESTAURANTE
36   36  AOI
20   20  Al Paladino
4    4   An An
11   11  Asia Imbiss Feinkost
10   10  Augustiner Bräustuben
0    0   Augustiner-Keller
21   21  Café Gollier
27   27  California Bean
25   25  Drunken Cow Bar & Grill
18   18  Döner King's
46   46  Frischfutter
43   43  Gaststätte Osteria da Antonio
30   30  Grill & Grace
32   32  Gyoza Bar
28   28  Hannes
44   44  Hickory
26   26  Josefa
39   39  Kam Lung
23   23  Kim's
31   31  Kymata Modern Taverna & Winebar
47   47  La Casina
22   22  Lilli P .
```

```
48  48  Madame Hu
13  13  Malzraum
1   1   Marbella
49  49  Marita Café
42  42  Max´s Beef Noodles
14  14  Mr. Pancake
37  37  Mun Mun
40  40  Naxos Taverna
2   2   Neuhauser Augustiner
12  12  Neuhauser Wohnküche
45  45  Nur Einmal Leben
8   8   Phó
15  15  Quan Com
41  41  Restaurant 181
34  34  Restaurant Poseidon
7   7   Risotto Ristorante
38  38  Ristorante Pizzeria Roma
6   6   Rüen Thong
19  19  S'Maillinger
33  33  Schiller Bräu
24  24  Servus Heidi
16  16  Trattoria Anni 60
9   9   Trattoria Bellini
29  29  VolkArt
35  35  Wirtshaus Maxvorstadt
3   3   Yee Chino Restaurant
5   5   Zapata Mexican Bar

                                                        address  \
17  [Dachauer Str. 153, 80335 Munich, Germany]
36  [Volkartstr. 22, 80634 Munich, Germany]
20  [Heimeranplatz 1, 80999 Munich, Germany]
4   [Elvirastr. 12, 80636 Munich, Germany]
11  [Leonrodstr. 67, 80636 Munich, Germany]
10  [Landsberger Str. 19, 80339 Munich, Germany]
0   [Arnulfstr. 52, 80335 Munich, Germany]
21  [Gollierstr. 36, 80339 Munich, Germany]
27  [Dachauer Str. 12, 80335 Munich, Germany]
25  [Gabelsbergerstr. 58, 80333 Munich, Germany]
18  [Heideckstr. 14, 80637 Munich, Germany]
46  [Marsstr. 13, 80335 Munich, Germany]
43  [Fasaneriestr. 4, 80636 Munich, Germany]
30  [Guldeinstr. 50, 80339 Munich, Germany]
32  [Augustenstr. 47 A, 80333 Munich, Germany]
28  [Hedwigstr. 9, 80636 Munich, Germany]
44  [Blutenburgstr. 112, 80636 Munich, Germany]
26  [Westendstr. 29, 80339 Munich, Germany]
```

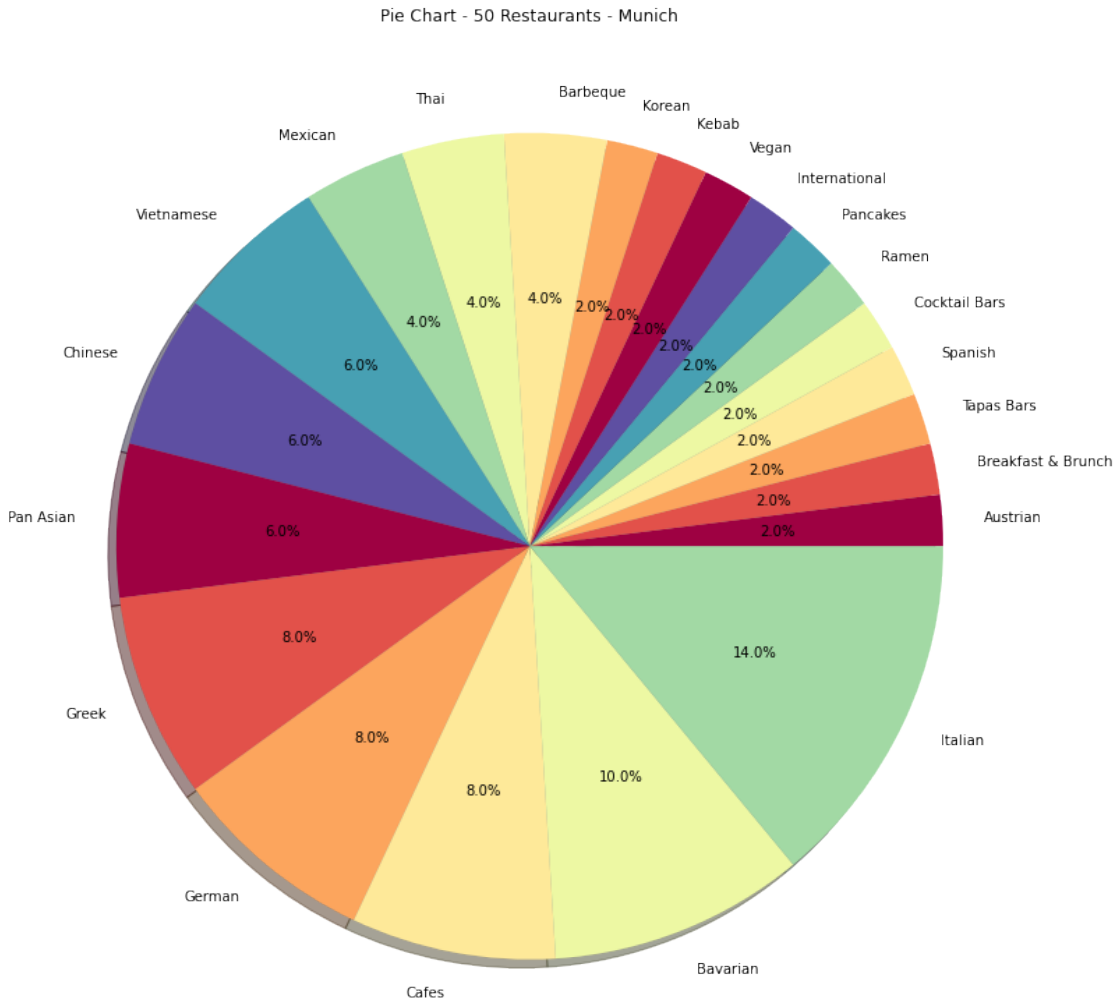
```
39  [Blutenburgstr. 53, 80636 Munich, Germany]
23  [Theresienstr. 138, 80333 Munich, Germany]
31  [Volpinistraße 19, 80638 Munich, Germany]
47  [Frohschammerstr. 14, 80807 Munich, Germany]
22  [Lilli-Palmer-Str. 2, 80636 Munich, Germany]
48  [Gollierstr. 20, 80339 Munich, Germany]
13  [Artilleriestr. 5, 80636 Munich, Germany]
1   [Horemansstr. 30, 80636 Munich, Germany]
49  [Schulstr. 34, 80634 Munich, Germany]
42  [Sendlinger-Tor-Platz 10, 80336 Munich, Germany]
14  [Gabelsbergerstr. 34, 80333 Munich, Germany]
37  [Münchner Freiheit 18, 80802 Munich, Germany]
40  [Verdistr. 33, 81247 Munich, Germany]
2   [Hübnerstr. 23, 80637 Munich, Germany]
12  [Lachnerstr. 1, 80639 Munich, Germany]
45  [Riesenfeldstr. 72, 80809 Munich, Germany]
8   [Nymphenburger Str. 70, 80335 Munich, Germany]
15  [Wendl-Dietrich-Str. 4, 80634 Munich, Germany]
41  [Spiridon-Louis-Ring 7, Olympiaturm, 80809 Munich, Germany]
34  [Leonrodstr. 85, 80636 Munich, Germany]
7   [Hirschgartenallee 38, 80639 Munich, Germany]
38  [Dom-Pedro-Str. 6, 80637 Munich, Germany]
6   [Thorwaldsenstr. 19, 80335 Munich, Germany]
19  [Maillinger Str. 4, 80636 Munich, Germany]
33  [Schillerstr. 23, 80336 Munich, Germany]
24  [Landsberger Str. 73, 80339 Munich, Germany]
16  [Blutenburgstr. 94, 80636 Munich, Germany]
9   [Nymphenburger Str. 120, 80636 Munich, Germany]
29  [Volkartstr. 15, 80634 Munich, Germany]
35  [Augustenstr. 53, 80333 Munich, Germany]
3   [Helene-Weber-Allee 19, 80637 Munich, Germany]
5   [Wilderich-Lang-Str. 4, 80634 Munich, Germany]

              phone  rating  review_count   latitude  longitude price  \
17  +498918985117   5.0     2             48.153812  11.553989
36  +498918008880   4.0     5             48.155570  11.534390
20  +49895025657    4.5     73            48.133400  11.534600  €€
4   +498955286459   5.0     12            48.149020  11.542900
11  +49891292705    5.0     4             48.157940  11.543960
10  +4989507047     4.0     417           48.139120  11.545660  €€
0   +4989594393     4.0     633           48.143500  11.551950  €€
21  +498920188886   5.0     8             48.135810  11.540820  €
27  +498955264422   4.5     140           48.142893  11.560298  €€
25  +498954356230   4.0     24            48.149750  11.560920  €€€
18                  4.5     7             48.161600  11.540400  €
46  +498969300858   4.5     7             48.142920  11.558610  €
43  +49891231265    4.0     32            48.157840  11.542620  €€
```

```
30  +498914348940   4.0     27          48.138859  11.529170  €€€
32  +498920346647   4.5     60          48.148641  11.562419  €
28  +498920062510   4.5     18          48.152500  11.539900  €€
44  +498913928530   4.5     8           48.150570  11.537388
26  +498928979183   4.5     21          48.138370  11.546180  €
39  +49891291254    4.0     47          48.149810  11.539470  €€
23  +498937966880   4.0     59          48.151650  11.562890  €€
31  +498914332533   4.5     34          48.165267  11.511693  €€
47  +49893598320    4.5     38          48.179131  11.568794  €€
22  +498990175452   4.5     4           48.144564  11.537821
48  +498945217272   4.0     24          48.135948  11.543390  €€
13  +4989187997     4.5     21          48.156370  11.540220  €€
1   +498912779753   4.5     38          48.153883  11.541679  €€
49  +498913011652   4.5     3           48.149320  11.535380  €€
42                  4.5     17          48.134046  11.566703
14  +498989059057   4.5     72          48.148922  11.563762  €
37  +4989248817588  4.5     33          48.163070  11.587570  €€
40  +498985793920   4.5     60          48.163890  11.479100  €€
2   +49891202130    4.5     36          48.159200  11.540600  €€
12  +498912021478   4.5     22          48.156140  11.530650  €€
45  +498935396563   4.5     98          48.183310  11.563310  €€
8   +498912738768   4.5     28          48.149813  11.548280  €€
15  +498912022167   4.5     43          48.152810  11.531610  €€
41  +4989350948181  4.0     127         48.174529  11.553731  €€€€
34  +498918979623   4.0     77          48.158870  11.545710  €€
7   +498917095709   4.5     15          48.153960  11.507260  €€
38  +49891574163    4.0     2           48.159830  11.545720  €€
6   +498912715461   4.5     33          48.150960  11.548670  €€
19  +498978795888   4.0     27          48.149310  11.545970  €€
33  +4989890584822  4.5     51          48.136700  11.561150  €€
24  +498955276303   4.5     17          48.139630  11.537750  €€
16  +498999018083   4.5     32          48.150060  11.540010  €€
9   +498912789888   4.5     26          48.151290  11.540930  €€
29  +498921969883   4.5     33          48.154970  11.533990  €€€
35  +498995493526   4.5     35          48.149250  11.562602  €€
3   +498915988587   4.5     49          48.163221  11.543674  €€
5   +49891665822    4.0     62          48.148284  11.535622  €€

                catflag              cat1                    cat2  \
17  Conditional Value 1  Mexican            American (Traditional)
36  Conditional Value 1  Ramen
20  Conditional Value 1  Italian
4   Conditional Value 1  Vietnamese
11  Continental          Pan Asian
10  Conditional Value 1  German             Bavarian
0   Continental          Bavarian           Beer Garden
21  Conditional Value 1  Cafes              Breakfast & Brunch
```

```
27  Conditional Value 1  Cafes                American (Traditional)
25  Conditional Value 1  Cocktail Bars       Steakhouses
18  Conditional Value 1  Kebab
46  Conditional Value 1  Vegan               Vegetarian
43  Conditional Value 1  Italian
30  Conditional Value 1  Barbeque            Steakhouses
32  Conditional Value 1  Chinese
28  Conditional Value 1  Austrian
44  Conditional Value 1  Barbeque            Smokehouse
26  Conditional Value 1  Cafes
39  Conditional Value 1  Chinese
23  Conditional Value 1  Korean
31  Conditional Value 1  Greek
47  Conditional Value 1  Italian
22  Conditional Value 1  Breakfast & Brunch  Cafes
48  Continental          Pan Asian
13  Conditional Value 1  German              Dive Bars
1   Continental          Spanish             Tapas/Small Plates
49  Conditional Value 1  Cafes               Breakfast & Brunch
42  Conditional Value 1  Chinese
14  Conditional Value 1  Pancakes
37  Conditional Value 1  Thai
40  Conditional Value 1  Greek               Seafood
2   Conditional Value 1  German
12  Conditional Value 1  International
45  Conditional Value 1  Greek               Mediterranean
8   Conditional Value 1  Vietnamese          Wok
15  Conditional Value 1  Vietnamese
41  Conditional Value 1  German
34  Conditional Value 1  Greek
7   Conditional Value 1  Italian
38  Conditional Value 1  Italian
6   Conditional Value 1  Thai
19  Continental          Bavarian            Austrian
33  Continental          Bavarian            Brewpubs
24  Continental          Bavarian
16  Conditional Value 1  Italian             Pizza
9   Conditional Value 1  Italian
29  Conditional Value 1  Tapas Bars
35  Continental          Bavarian
3   Continental          Pan Asian
5   Conditional Value 1  Mexican             Cocktail Bars

                cat3
17
36
20
```

```
4
11
10  Gastropubs
0
21
27  Breakfast & Brunch
25  Burgers
18
46  Food Stands
43
30
32
28
44
26
39
23
31
47
22
48
13
1
49
42
14
37
40  Gluten-Free
2
12
45
8
15
41
34
7
38
6
19
33
24
16  Seafood
9
29
35
3
5
```

[97]:
```
# pie chart
# import libraries
import matplotlib.ticker as ticker
import matplotlib.cm as cm
import matplotlib as mpl
from matplotlib.gridspec import GridSpec
import matplotlib.pyplot as plt

from collections import Counter

# declare categories column
my_cat1 =dff['cat1']

title_type = dff.groupby('cat1').agg('count')

# Pie Chart - matplotlib
type_labels = title_type.ID.sort_values().index
type_counts = title_type.ID.sort_values()
plt.figure(1, figsize=(50,30))
the_grid = GridSpec(2, 2)
cmap = plt.get_cmap('Spectral')
colors = [cmap(i) for i in np.linspace(0, 1, 8)]

plt.subplot(the_grid[0, 1], aspect=1, title='Pie Chart - 50 Restaurants -␣
↪Munich')
type_show_ids = plt.pie(type_counts, labels=type_labels, autopct='%1.1f%%',␣
↪shadow=True, colors=colors)
plt.show()
```

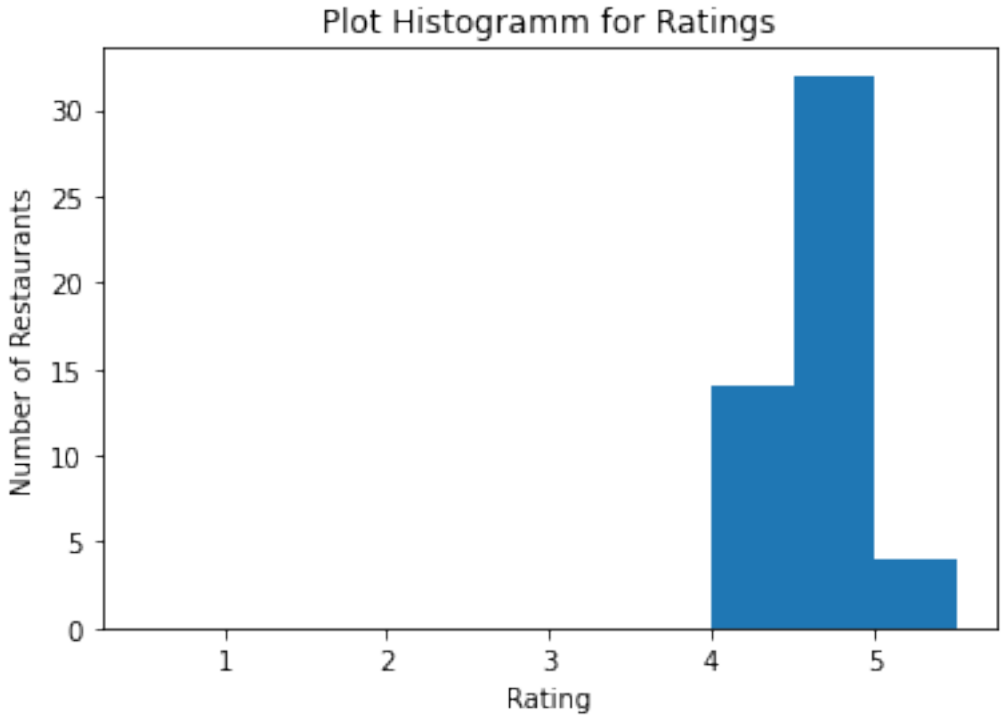


[104]:
```
# histogram graph to diplay Rating vs Nr of Restaurants
rat1 = dff['rating']

# Rating Percentages
# Rating Values 4,5 (64%) - 4,0 (28%) - 5,0 (8%)
print('Rating values are', '\n', rat1.value_counts(normalize=True))

plt.title("Plot Histogramm for Ratings")
plt.xlabel('Rating')
plt.ylabel('Number of Restaurants')

plt.hist(rat1, range=(0.5, 5.5), bins=10)
```

```
Rating values are
 4.5    0.64
```

```
4.0    0.28
5.0    0.08
Name: rating, dtype: float64
```

[104]:
```
(array([ 0.,  0.,  0.,  0.,  0.,  0.,  0., 14., 32.,  4.]),
 array([0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. , 5.5]),
 <BarContainer object of 10 artists>)
```



[106]:
```
# dff1.head()
df_flat4=dff1
# df_flat4.head()
dff2=dff
dff1.head(10)
```

[106]:

	cat1	cat2	cat3
0	Bavarian	Beer Garden	NaN
1	Spanish	Tapas/Small Plates	NaN
2	German	NaN	NaN
3	Pan Asian	NaN	NaN
4	Vietnamese	NaN	NaN
5	Mexican	Cocktail Bars	NaN
6	Thai	NaN	NaN
7	Italian	NaN	NaN
8	Vietnamese	Wok	NaN

```
9  Italian     NaN                 NaN
```

[124]:
```
# Extract from dff1 - cat 1/2/3 Columns all relevant categroisation information
df_flat4['fl1'] = dff1.cat1.str.extract(r'\b(\w+)$', expand=True)
df_flat4['fl2'] = dff1.cat2.str.extract(r'\b(\w+)$', expand=True)
df_flat4['fl3'] = dff1.cat3.str.extract(r'\b(\w+)$', expand=True)
df_flat4 = dff1.fillna('')


# New CAT Column to store all previously extracted - relevant categroisation␣
 ↪information
df_flat4['CAT'] = df_flat4['cat1'] +' - '+ df_flat4['cat2']+' - '+␣
 ↪df_flat4['cat3']
df_flat4.head(50)
```

[124]:

	cat1	cat2	cat3	\
0	Bavarian	Beer Garden		
1	Spanish	Tapas/Small Plates		
2	German			
3	Pan Asian			
4	Vietnamese			
5	Mexican	Cocktail Bars		
6	Thai			
7	Italian			
8	Vietnamese	Wok		
9	Italian			
10	German	Bavarian	Gastropubs	
11	Pan Asian			
12	International			
13	German	Dive Bars		
14	Pancakes			
15	Vietnamese			
16	Italian	Pizza	Seafood	
17	Mexican	American (Traditional)		
18	Kebab			
19	Bavarian	Austrian		
20	Italian			
21	Cafes	Breakfast & Brunch		
22	Breakfast & Brunch	Cafes		
23	Korean			
24	Bavarian			
25	Cocktail Bars	Steakhouses	Burgers	
26	Cafes			
27	Cafes	American (Traditional)	Breakfast & Brunch	
28	Austrian			
29	Tapas Bars			
30	Barbeque	Steakhouses		

```
31  Greek
32  Chinese
33  Bavarian            Brewpubs
34  Greek
35  Bavarian
36  Ramen
37  Thai
38  Italian
39  Chinese
40  Greek               Seafood                 Gluten-Free
41  German
42  Chinese
43  Italian
44  Barbeque            Smokehouse
45  Greek               Mediterranean
46  Vegan               Vegetarian              Food Stands
47  Italian
48  Pan Asian
49  Cafes               Breakfast & Brunch

              fl1             fl2         fl3  \
0   Bavarian        Garden
1   Spanish         Plates
2   German
3   Asian
4   Vietnamese
5   Mexican         Bars
6   Thai
7   Italian
8   Vietnamese      Wok
9   Italian
10  German          Bavarian        Gastropubs
11  Asian
12  International
13  German          Bars
14  Pancakes
15  Vietnamese
16  Italian         Pizza           Seafood
17  Mexican
18  Kebab
19  Bavarian        Austrian
20  Italian
21  Cafes           Brunch
22  Brunch          Cafes
23  Korean
24  Bavarian
25  Bars            Steakhouses     Burgers
```

```
26  Cafes
27  Cafes                       Brunch
28  Austrian
29  Bars
30  Barbeque       Steakhouses
31  Greek
32  Chinese
33  Bavarian       Brewpubs
34  Greek
35  Bavarian
36  Ramen
37  Thai
38  Italian
39  Chinese
40  Greek          Seafood      Free
41  German
42  Chinese
43  Italian
44  Barbeque       Smokehouse
45  Greek          Mediterranean
46  Vegan          Vegetarian   Stands
47  Italian
48  Asian
49  Cafes          Brunch
```

```
                                                CAT
0   Bavarian - Beer Garden -
1   Spanish - Tapas/Small Plates -
2   German -  -
3   Pan Asian -  -
4   Vietnamese -  -
5   Mexican - Cocktail Bars -
6   Thai -  -
7   Italian -  -
8   Vietnamese - Wok -
9   Italian -  -
10  German - Bavarian - Gastropubs
11  Pan Asian -  -
12  International -  -
13  German - Dive Bars -
14  Pancakes -  -
15  Vietnamese -  -
16  Italian - Pizza - Seafood
17  Mexican - American (Traditional) -
18  Kebab -  -
19  Bavarian - Austrian -
20  Italian -  -
```

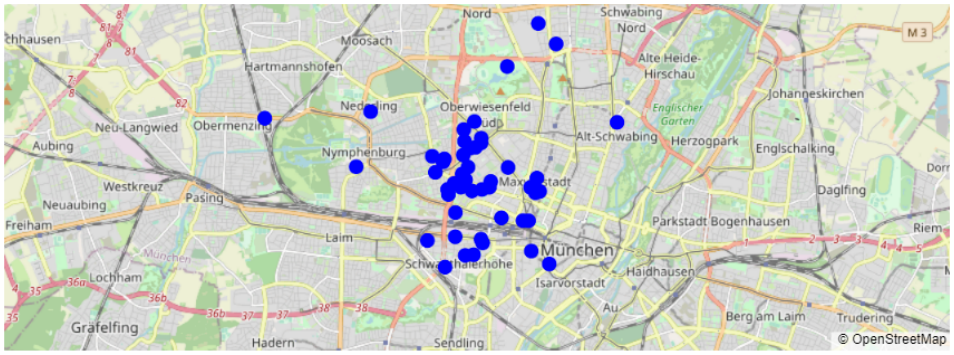
```
21  Cafes - Breakfast & Brunch -
22  Breakfast & Brunch - Cafes -
23  Korean -  -
24  Bavarian -  -
25  Cocktail Bars - Steakhouses - Burgers
26  Cafes -  -
27  Cafes - American (Traditional) - Breakfast & Brunch
28  Austrian -  -
29  Tapas Bars -  -
30  Barbeque - Steakhouses -
31  Greek -  -
32  Chinese -  -
33  Bavarian - Brewpubs -
34  Greek -  -
35  Bavarian -  -
36  Ramen -  -
37  Thai -  -
38  Italian -  -
39  Chinese -  -
40  Greek - Seafood - Gluten-Free
41  German -  -
42  Chinese -  -
43  Italian -  -
44  Barbeque - Smokehouse -
45  Greek - Mediterranean -
46  Vegan - Vegetarian - Food Stands
47  Italian -  -
48  Pan Asian -  -
49  Cafes - Breakfast & Brunch -
```

[125]:
```
C1=df_flat4['CAT']
# Assign C1 - to category column - which now contains all categorization␣
 ↪paramters
dff2['category']= C1
# dff2.drop(['price', 'catflag', 'cat1','cat2', 'cat3'], axis=1, inplace=True)
# Sort dff Alphabetically - Column Name
dff2.sort_values(by=['name'], ascending=True)
dff2.head()
```

[125]:
```
   ID                  name                                          address  \
0  0   Augustiner-Keller     [Arnulfstr. 52, 80335 Munich, Germany]
1  1   Marbella              [Horemansstr. 30, 80636 Munich, Germany]
2  2   Neuhauser Augustiner  [Hübnerstr. 23, 80637 Munich, Germany]
3  3   Yee Chino Restaurant  [Helene-Weber-Allee 19, 80637 Munich, Germany]
4  4   An An                 [Elvirastr. 12, 80636 Munich, Germany]

           phone  rating  review_count   latitude  longitude price  \
```

```
0  +4989594393    4.0     633           48.143500  11.551950  €€
1  +498912779753  4.5     38            48.153883  11.541679  €€
2  +49891202130   4.5     36            48.159200  11.540600  €€
3  +498915988587  4.5     49            48.163221  11.543674  €€
4  +498955286459  5.0     12            48.149020  11.542900

               catflag        cat1                cat2 cat3  \
0  Continental          Bavarian    Beer Garden
1  Continental          Spanish     Tapas/Small Plates
2  Conditional Value 1  German
3  Continental          Pan Asian
4  Conditional Value 1  Vietnamese

                          category
0  Bavarian - Beer Garden -
1  Spanish - Tapas/Small Plates -
2  German -  -
3  Pan Asian -  -
4  Vietnamese -  -
```

[126]:
```
# Create interactive plotly express map to display 50 restaurants in Munich,␣
 ↪with following data:
# name lat, long, address, category, review count

import plotly.express as px
fig = px.scatter_mapbox(dff2, lat="latitude", lon="longitude",
                        hover_name='name', hover_data=['address', 'category',␣
 ↪'review_count'], zoom=11, size_max=40)
fig.update_layout(mapbox_style="open-street-map")
fig.update_traces(marker=dict(size=14, color='blue'))
fig.show()
```



4.3.1 1.4 Results - We have various Results for 50 Restaurants in Munich

Results are visualized using Pie Chart, Interactive Map and Histogram for Rating - The Pie Chart visualizes categories with relevant percentages. Interactive Maps using the Plotly library/ StreetMap displays Restaurant information in the Popup Menu (Name, Address, Coordinates, Category…) it also displays location in the city Map, this is useful to asses where the might be demand for a particular Restaurant category. Histogram Rating - displays that there are no ratings below 4,0 - most of them are 4,5 (64%), followed by 4,0 (28%) and 5,0 (8%) ### 1.5 Discussion - section where you discuss any observations you noted and any recommendations you can make based on the results. The Data Science analysis in this project has given us an insight for 50 Restaurants in Munich - covering Data Requirements, Collection Preparation and later Data Understanding - looking at the Pie Chart above in the Notebook, we can estimate that there is room for more Vegan/Vegeterian, Thai, Tapas, International, Latin American, Austrian, Pan-Asian, Vietnamese and Chinese Restaurants. The interactive Map displays a centered distribution of the Restaurants, there should be more demand for international cuisine in the suburbs East/West/North/South. ### 1.6 Conclusion - section where you conclude the report. This assignment gives a complete overview over the Data Science Methodology - Business Understanding, Analytic Approach initially. After that the "Data Chain" is evaluated: Data Requirements, Collection, Understanding and Data Preparation is assessed and implemented. Here we have Software Engineering constraints and limitations that depend on the various projects and technologies. In this assignment I chose to work with the Yelp API, its limitation of 50 Search Results is obviously a constraint for this project, but it was chosen also to limit the length of this assignment. But, we have access with Python and its libraries to excellent Data Science Tools, for visualization, evaluation end estimation. Another API with more search results could improve data science evaluation quality and accuracy. Obviously more analysis, CPU Power and Memory and data processing should be applied.

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: