Hunt for Indian Restaurants in AUSTRIA

IBM Capstone Data Science Project

Authored By: Deepankar Singh

Date: 20th March 2019

1. INTRODUCTION

1.1 Background/Business Problem

The founder of one of the best Influencers and Food Community of India ‘SpicyTadka’ began her career with leading news channels like headlines today, Times Now and News X. After working for 7 years, she decided to pursue her passion for photography and blogging, eventually leading to Food Photography and Food Community. During her visit to Europe, she wanted to make a blog about the Indian food in Europe. And, they wanted to start their blog from the Centre of Europe – AUSTRIA.

So, the team of SpicyTadka requested our Data Science team to find out the Best Indian Restaurant (In terms of overall ratings) in AUSTRIA.

1.2 Interests

The Indian Food Bloggers and Food Explorers across the Europe would be very interested to know about the Best Indian Restaurant in Austria so that they can explore and can make their blogs as well in the future. Also, the Indian Community and travelers who are in Austria and searching for Indian Food would get to know the best Indian Restaurants.

2. DATA

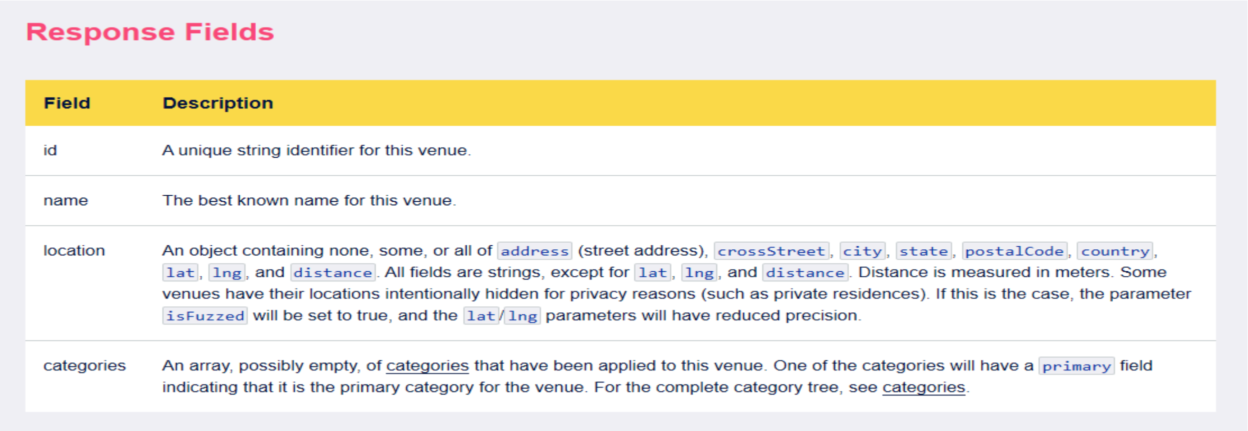
2.1 Simplemaps.com

Has a World Cities Database and it offers a simple, accurate and up-to-date database of the world's cities and towns, which they have built it from the ground up using authoritative sources such as the NGIA, US Geological Survey, US Census Bureau, and NASA. We have taken the data for 81 cities of Austria and their coordinates from the website and save it our Git Hub repository. Below are the Links for both.

Simplemaps.com : https://simplemaps.com/data/at-cities
Git Hub Repository : https://github.com/Ekdeep0712/Capstone-Project-Coursera/blob/master/at.csv

2.2 Foursquare API

For this project we are also using Foursquare API. Foursquare is a social location service that allows users to explore the world around them. One of the best characteristics of this API is, it can provide you the list of venues based on the coordinates and radius given. By using the “explore” feature with correct parameters, will give you a JASON file mentioning the below information.



Response Fields

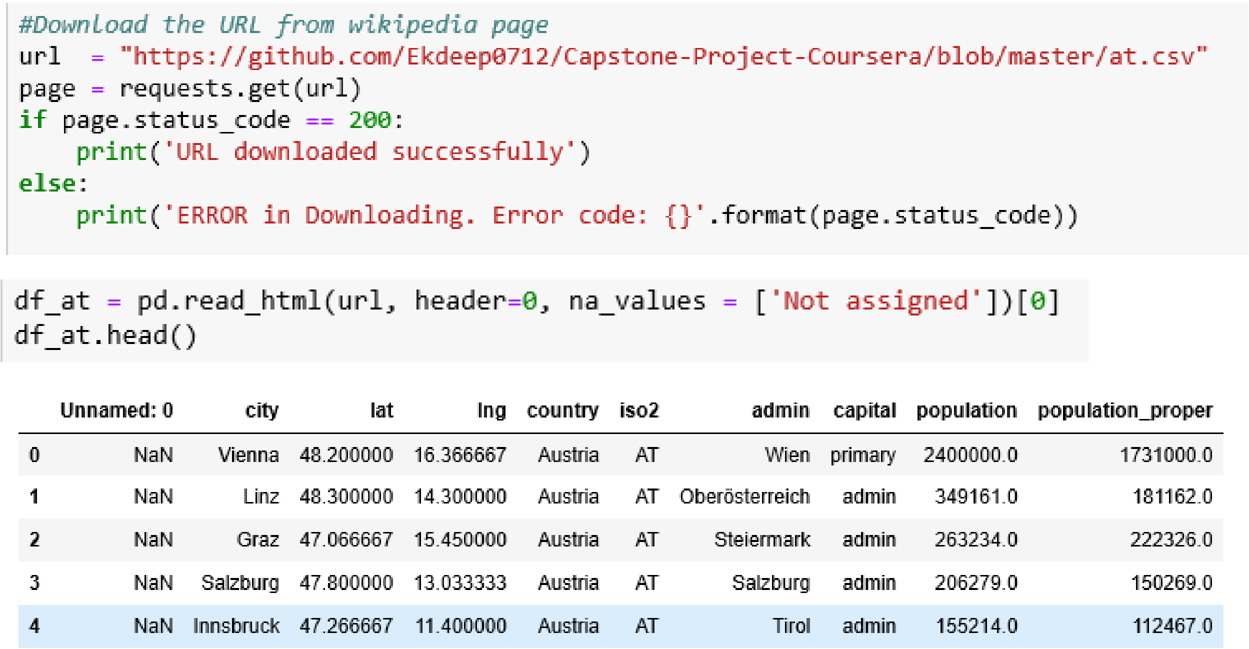
Field	Description
id	A unique string identifier for this venue.
name	The best known name for this venue.
location	An object containing none, some, or all of `address` (street address), `crossStreet`, `city`, `state`, `postalCode`, `country`, `lat`, `lng`, and `distance`. All fields are strings, except for `lat`, `lng`, and `distance`. Distance is measured in meters. Some venues have their locations intentionally hidden for privacy reasons (such as private residences). If this is the case, the parameter `isFuzzed` will be set to true, and the `lat`/`lng` parameters will have reduced precision.
categories	An array, possibly empty, of categories that have been applied to this venue. One of the categories will have a `primary` field indicating that it is the primary category for the venue. For the complete category tree, see categories.

We used Foursquare to find the list of Indian Restaurants in Austria and to find the ratings of the restaurants.

3. METHODOLOGY

3.1 Data Preprocessing

During the data preprocessing stage, we downloaded the data containing cities, population & coordinates of Austria (fig.1). After renaming the column names (fig.2) and remove the columns which we don’t need (fig.3). After preprocessing, we came up with the new data frame with all the information we needed.

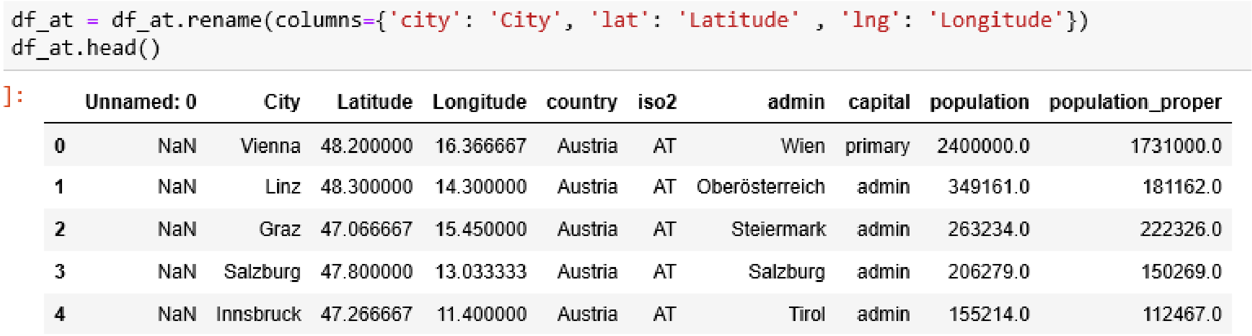


```
#Download the URL from wikipedia page
url  = "https://github.com/Ekdeep0712/Capstone-Project-Coursera/blob/master/at.csv"
page = requests.get(url)
if page.status_code == 200:
    print('URL downloaded successfully')
else:
    print('ERROR in Downloading. Error code: {}'.format(page.status_code))
```

```
df_at = pd.read_html(url, header=0, na_values = ['Not assigned'])[0]
df_at.head()
```

	Unnamed: 0	city	lat	lng	country	iso2	admin	capital	population	population_proper
0	NaN	Vienna	48.200000	16.366667	Austria	AT	Wien	primary	2400000.0	1731000.0
1	NaN	Linz	48.300000	14.300000	Austria	AT	Oberösterreich	admin	349161.0	181162.0
2	NaN	Graz	47.066667	15.450000	Austria	AT	Steiermark	admin	263234.0	222326.0
3	NaN	Salzburg	47.800000	13.033333	Austria	AT	Salzburg	admin	206279.0	150269.0
4	NaN	Innsbruck	47.266667	11.400000	Austria	AT	Tirol	admin	155214.0	112467.0

Figure.1 Dataset downloaded

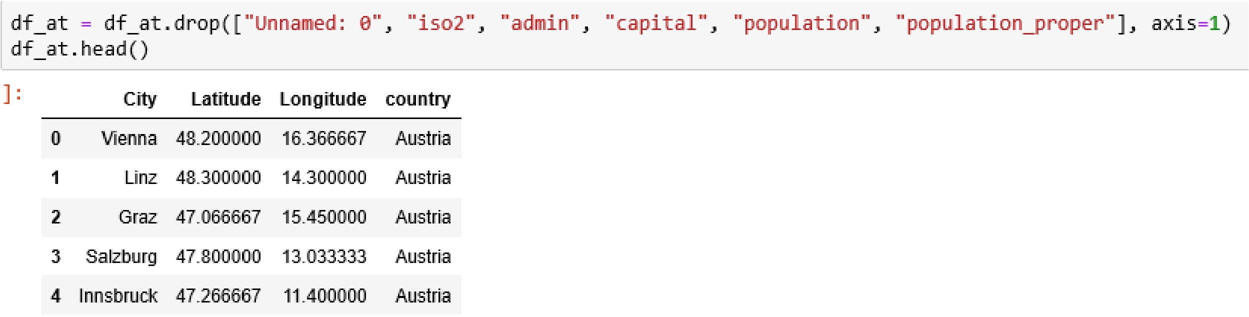


```
df_at = df_at.rename(columns={'city': 'City', 'lat': 'Latitude' , 'lng': 'Longitude'})
df_at.head()
```

]:

	Unnamed: 0	City	Latitude	Longitude	country	iso2	admin	capital	population	population_proper
0	NaN	Vienna	48.200000	16.366667	Austria	AT	Wien	primary	2400000.0	1731000.0
1	NaN	Linz	48.300000	14.300000	Austria	AT	Oberösterreich	admin	349161.0	181162.0
2	NaN	Graz	47.066667	15.450000	Austria	AT	Steiermark	admin	263234.0	222326.0
3	NaN	Salzburg	47.800000	13.033333	Austria	AT	Salzburg	admin	206279.0	150269.0
4	NaN	Innsbruck	47.266667	11.400000	Austria	AT	Tirol	admin	155214.0	112467.0

Figure.2 Rename the columns



```
df_at = df_at.drop(["Unnamed: 0", "iso2", "admin", "capital", "population", "population_proper"], axis=1)
df_at.head()
```

]:

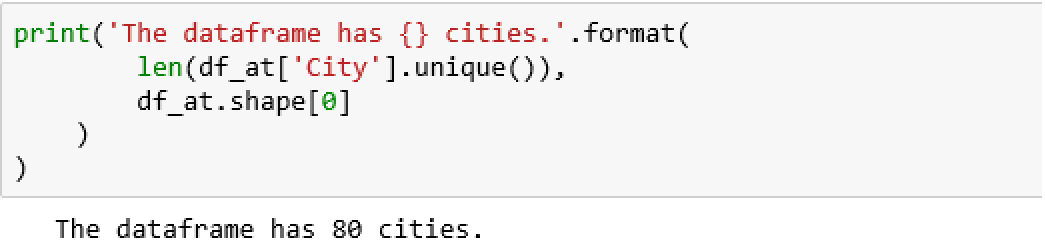
	City	Latitude	Longitude	country
0	Vienna	48.200000	16.366667	Austria
1	Linz	48.300000	14.300000	Austria
2	Graz	47.066667	15.450000	Austria
3	Salzburg	47.800000	13.033333	Austria
4	Innsbruck	47.266667	11.400000	Austria

Figure.3 Drop the unwanted columns

As shown in figure.3, during the data preprocessing stage we created the dataset containing the basic information we needed for Austria.

3.2 Exploratory Data Analysis

Let's check the number of cities we are working with (fig.4).

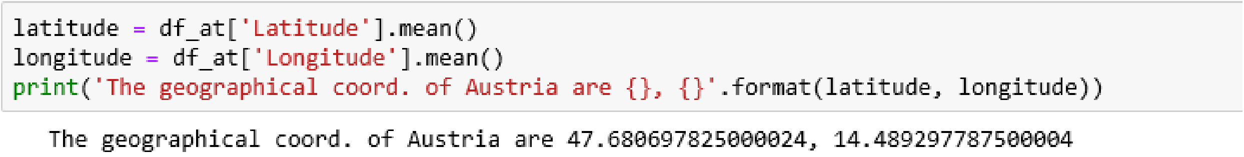


```
print('The dataframe has {} cities.'.format(
        len(df_at['City'].unique()),
        df_at.shape[0]
    )
)
```

```
The dataframe has 80 cities.
```

Figure.4 Total number of cities in Austria

Then, we will check the geographical coordinates of Austria (fig.5).

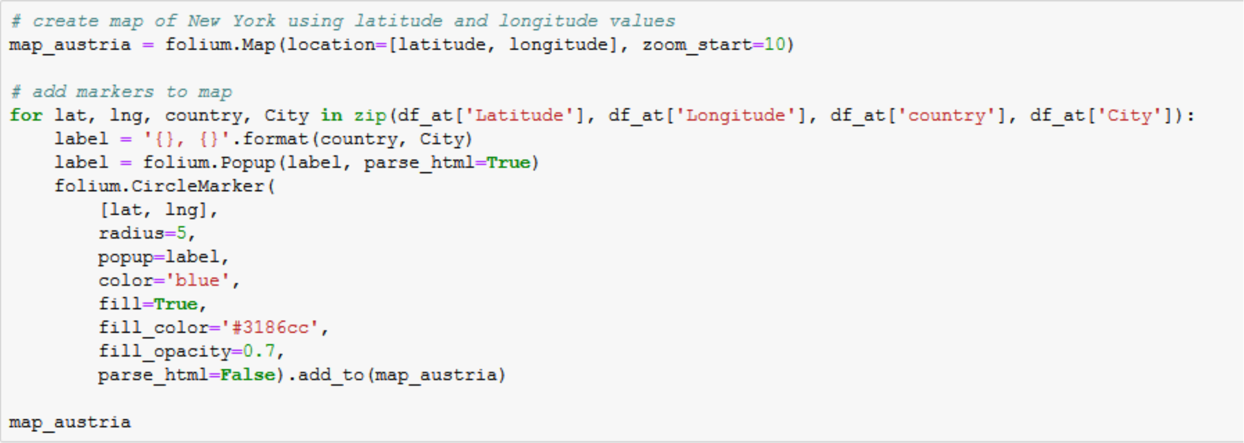


```
latitude = df_at['Latitude'].mean()
longitude = df_at['Longitude'].mean()
print('The geographical coord. of Austria are {}, {}'.format(latitude, longitude))
```

```
The geographical coord. of Austria are 47.680697825000024, 14.489297787500004
```

Figure.5 Geographical Coordinates

We used python **folium** library to visualize geographic details of Austria and its cities using the geographical coordinates (fig.6) and also created a map of Austria with cities superimposed on top (fig7).



```
# create map of New York using latitude and longitude values
map_austria = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, country, City in zip(df_at['Latitude'], df_at['Longitude'], df_at['country'], df_at['City']):
    label = '{}, {}'.format(country, City)
    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_austria)

map_austria
```

Figure.6 Using folium library

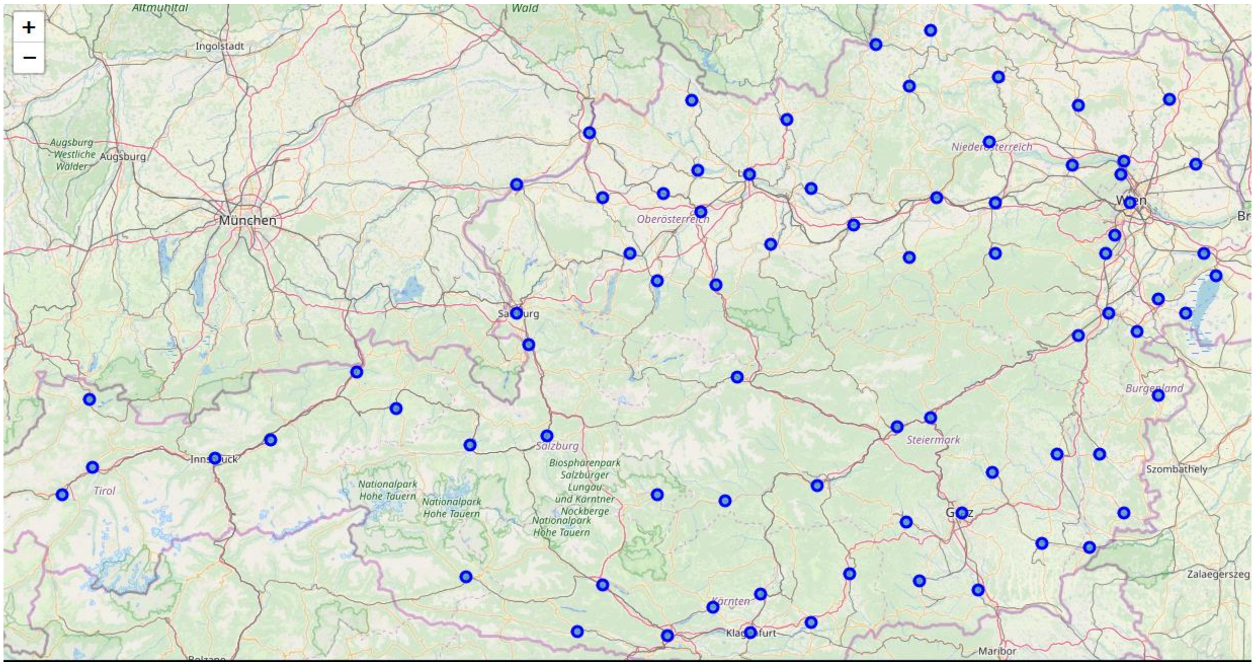
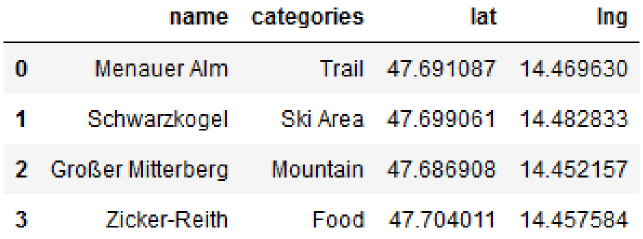
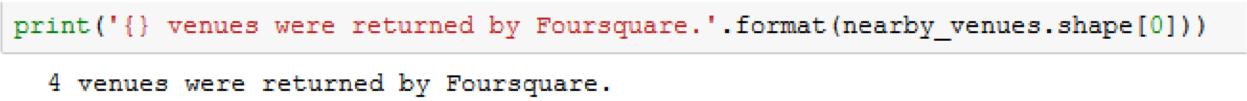


Figure.7 Map of Austria with cities

We utilized the Foursquare API to explore the cities and segment them. I designed the limit as **50 venues** and the radius **5000 meter** for each city from their given latitude and longitude information. Here is a figure of the list Venues name, category, latitude and longitude information from Foursquare API (fig.8). As we received only 4 venues by this operation.



	name	categories	lat	lng
0	Menauer Alm	Trail	47.691087	14.469630
1	Schwarzkogel	Ski Area	47.699061	14.482833
2	Großer Mitterberg	Mountain	47.686908	14.452157
3	Zicker-Reith	Food	47.704011	14.457584

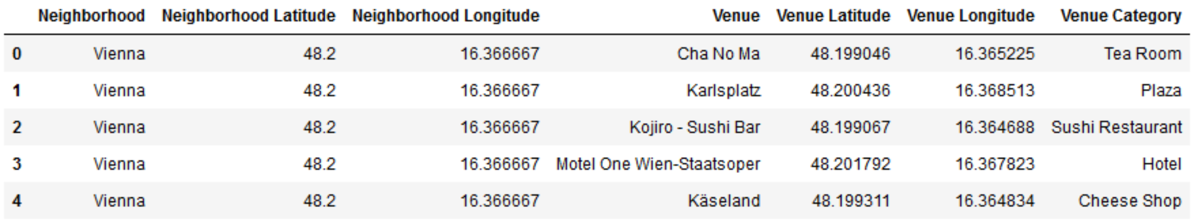


```
print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))
```

```
4 venues were returned by Foursquare.
```

Figure.8 Venues with geographical coordinates

Now, we will repeat the same function to check the venues in all the cities present in our data frame and will create a new dataset with all the information. Below is the head of our new merged data frame along with coordinates (fig.9).

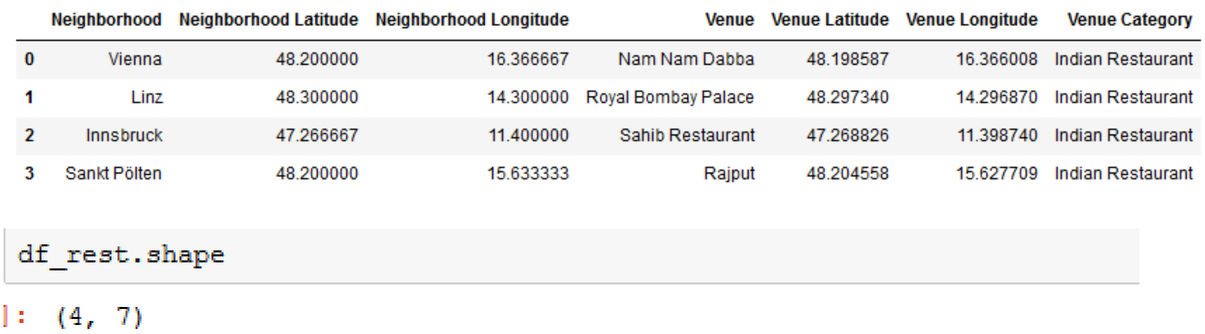


	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Vienna	48.2	16.366667	Cha No Ma	48.199046	16.365225	Tea Room
1	Vienna	48.2	16.366667	Karlsplatz	48.200436	16.368513	Plaza
2	Vienna	48.2	16.366667	Kojiro - Sushi Bar	48.199067	16.364688	Sushi Restaurant
3	Vienna	48.2	16.366667	Motel One Wien-Staatsoper	48.201792	16.367823	Hotel
4	Vienna	48.2	16.366667	Käseland	48.199311	16.364834	Cheese Shop

Figure.9 Merged data set with all the cities & venues with coordinates

As per above dataset we have total **1196** venue categories returned by Four square API.

Now as per our need, we filtered all the categories containing the “Indian” in its name as we are looking only for Indian Restaurants in Austria. And we received the below result (fig.10).



	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Vienna	48.200000	16.366667	Nam Nam Dabba	48.198587	16.366008	Indian Restaurant
1	Linz	48.300000	14.300000	Royal Bombay Palace	48.297340	14.296870	Indian Restaurant
2	Innsbruck	47.266667	11.400000	Sahib Restaurant	47.268826	11.398740	Indian Restaurant
3	Sankt Pölten	48.200000	15.633333	Rajput	48.204558	15.627709	Indian Restaurant

```
df_rest.shape
```

```
]:  (4, 7)
```

Figure.10 List of Indian Restaurants in Austria

We used again **folium** library to visualize geographic details of Indian Restaurants in Austria (fig.11).

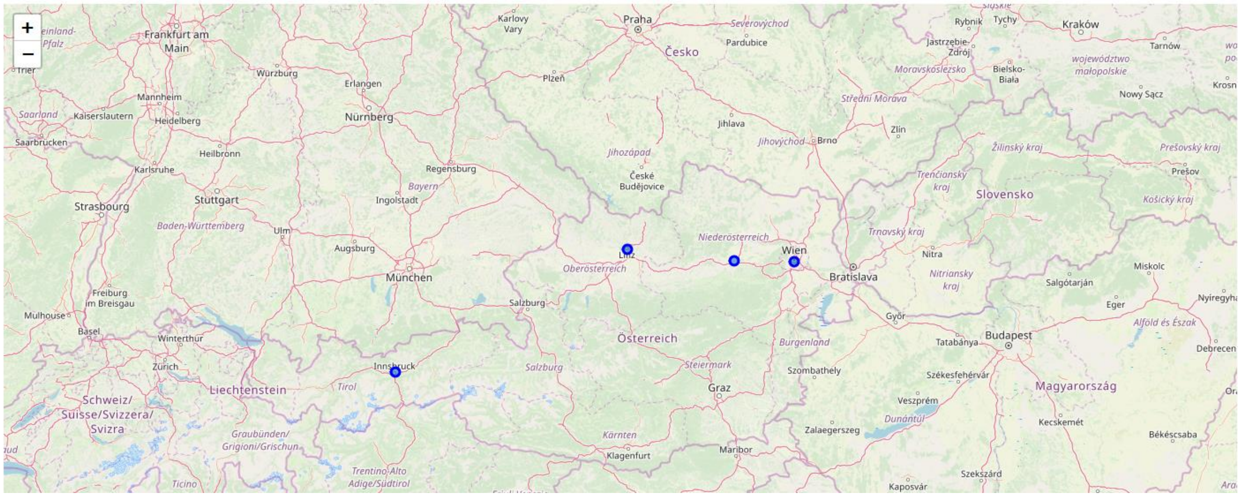
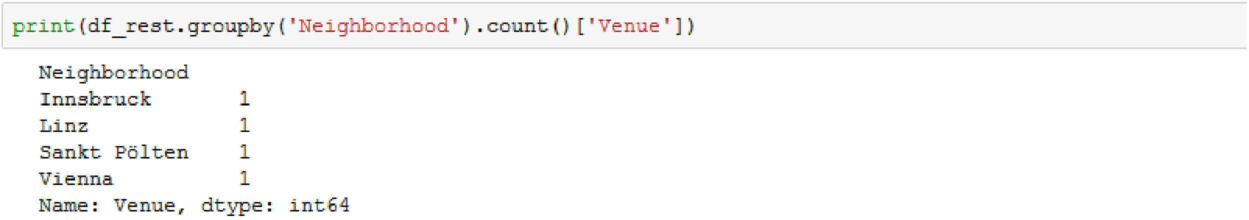


Figure.11 List of Indian Restaurants in Austria

As you see above, we only received the 4 Indian Restaurants in Austria according to Foursquare API.

4. RESULTS

Just to make sure the total numbers of Indian Restaurants we received per neighborhood, we used Group by function on our data set and we received the below result (fig.12).



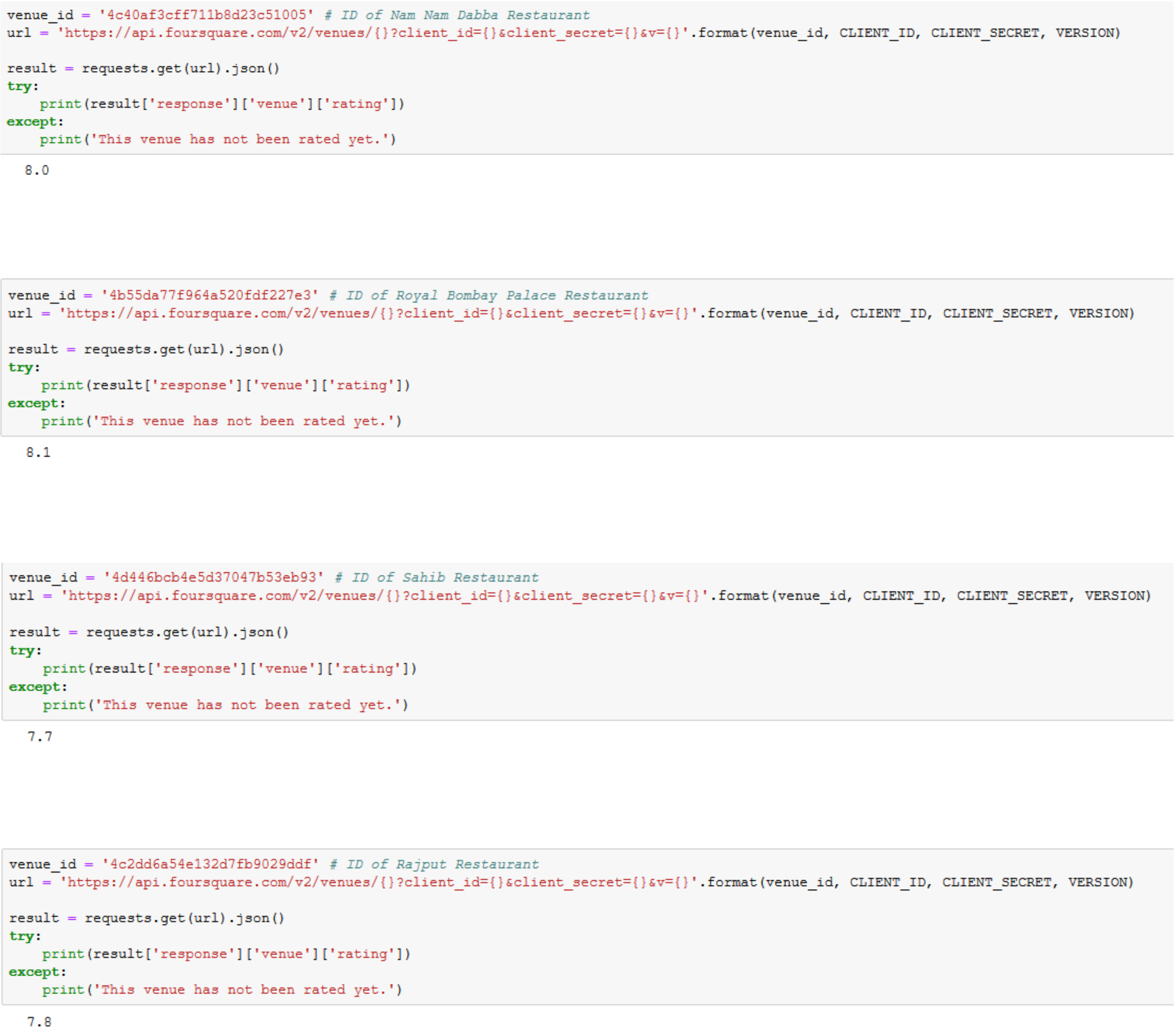
```
print(df_rest.groupby('Neighborhood').count()['Venue'])
```

```
Neighborhood
Innsbruck        1
Linz             1
Sankt Pölten     1
Vienna           1
Name: Venue, dtype: int64
```

Figure.12 Indian Restaurants per neighborhood

Now, our task is to find the best Indian Restaurant among them as per overall ratings.

So again, with the help of Foursquare API venue ID function, we fetch the overall ratings of all the restaurants (fig.13).



```
venue_id = '4c40af3cff711b8d23c51005' # ID of Nam Nam Dabba Restaurant
url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.format(venue_id, CLIENT_ID, CLIENT_SECRET, VERSION)

result = requests.get(url).json()
try:
    print(result['response']['venue']['rating'])
except:
    print('This venue has not been rated yet.')
```

```
8.0
```

```
venue_id = '4b55da77f964a520fdf227e3' # ID of Royal Bombay Palace Restaurant
url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.format(venue_id, CLIENT_ID, CLIENT_SECRET, VERSION)

result = requests.get(url).json()
try:
    print(result['response']['venue']['rating'])
except:
    print('This venue has not been rated yet.')
```

```
8.1
```

```
venue_id = '4d446bcb4e5d37047b53eb93' # ID of Sahib Restaurant
url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.format(venue_id, CLIENT_ID, CLIENT_SECRET, VERSION)

result = requests.get(url).json()
try:
    print(result['response']['venue']['rating'])
except:
    print('This venue has not been rated yet.')
```

```
7.7
```

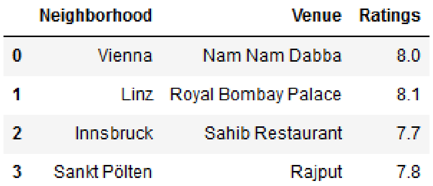
```
venue_id = '4c2dd6a54e132d7fb9029ddf' # ID of Rajput Restaurant
url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.format(venue_id, CLIENT_ID, CLIENT_SECRET, VERSION)

result = requests.get(url).json()
try:
    print(result['response']['venue']['rating'])
except:
    print('This venue has not been rated yet.')
```

```
7.8
```

Figure.13 Overall ratings of Indian Restaurants

Finally, we created a new dataset with overall rating of all four Indian Restaurants we found in Austria and below is the new data set with the final information we needed (fig.14).



	Neighborhood	Venue	Ratings
0	Vienna	Nam Nam Dabba	8.0
1	Linz	Royal Bombay Palace	8.1
2	Innsbruck	Sahib Restaurant	7.7
3	Sankt Pölten	Rajput	7.8

Figure.14 Final Data set of Indian Restaurants with Ratings

5. DISCUSSION

After creating a function to check the venues in all the cities present in our data frame and after creating a new dataset with all the information along with coordinates as shown in (fig.9), we were able to filter the venue categories and found the categories which have common name "Indian Restaurant" as shown in (fig.10).

As per Foursquare API, we found only 4 Indian Restaurants from 80 cities in Austria, so we had very less options for our client, however it was very easy for us to filter the best restaurant according to overall ratings.

We also visualized the map of restaurants we found in Austria with the cities they are located in.

6. CONCLUSION

As a result, from the above scenario, we were able to give best option to our client for the Indian Restaurants. So that, they will make their next food blog and can able to explore the different and best Indian dishes served in the best Indian Restaurant of Austria.

Also they can help the Indian Community and Indian travelers to find out the best place to eat through their blogs and let them know the best dishes they serve. They can also show the exact location and ambience through their videos and blogs.

During this project we have used different Data Wrangling methodologies with Pandas & Folium for map visualization.

For future projects with similar characteristics, it should be considered to expand the amount of data and machine learning algorithms. For. eg. K Means Clustering, DBSCAN etc.

Thank you
Deepankar Singh