IBM Data Science

Professional Certificate

Capstone Report

Spring 2021

Authored by: Derek

Table of Contents

Summary

Introduction / Business Problem Summary

1. Looking into the Ratings of Bagel Venues in New York City using FourSquare's API and ranking them from highest to lowest (Top 5).

2. Plot out all Bagel Store locations in New York City on an interactive map to visualize the closest locations.

Discussion / Results

1. Successfully created objective described in the Introduction

2. Objects can be reused in this code to pass different variables through, such as different locations or types of venues.

3. Original Scope was limited due to the restrictions of the free FourSquare API functionality vs. Premium (paid) features.

Introduction

Original Business Problem & Proposal

- When traversing Long Island and interacting with the local residents, you will find that almost all of them share pride in – their food. This is especially true for both their pizza and bagels.

- If a local Long Islander travels somewhere and is served a pizza or bagel, they cannot resist saying, “This is good. But, not nearly as good as the ones that we have on Long Island. You see, our water has a lower concentration of calcium and magnesium which makes them better than everywhere else.”

- Having lived on Long Island for the majority of my life, I wanted to prove this element to be true. So, I would like to analyze all of New York State using FourSquare’s API. I would like to take a look into the Ratings of Bagel Stores, sort the venue ratings by County, and discover which County has the best rated Bagel Stores.

New Business Problem & Proposal

- (a necessity to purchase a Premium FourSquare Developer Account prevented me to accomplishing the Original Proposal)

- Friends of mine who have never been to Long Island are coming to visit. After bragging about the local area having the best bagels in the world, I need to ensure I take them to the highest rated Bagel Shops.

- I would like to take a look into the Ratings of Bagel Stores in New York City using FourSquare’s API and rank them from highest to lowest. Then, I would like to plot out all Bagel Store locations in New York City on an interactive map so that we are able to see if one is in close range to us.

- Audience: myself and friends touring New York for the first time. The results are important to the audience for we all wish to find Bagel Venues in NYC, as well as which ones are the highest rated.

Version Control

This section outlines the details of each Version Update.

Date	Revision Author	Version	Changes
05/06/21	Derek	0.0	Initial Conceptualization
05/07/21	Derek	0.1	Data Research / Collection
05/08/21	Derek	0.2	Planning
05/09/21	Derek	1.0	Coding of Imported Libraries
05/09/21	Derek	1.1	Importing and Documenting Data Resources
05/10/21	Derek	1.2	Concatenation, Merging, and Cleaning of Data
05/10/21	Derek	2.0	Connection to and Use of FourSquare API
05/11/21	Derek	3.0	Manipulation and Analysis of Data from FourSquare API
05/11/21	Derek	3.1	Interpret Results and Establish Useful Visualization Roadmap
05/12/21	Derek	4.0	Fully Completed Code
05/12/21	Derek	4.1	Added Markdown Cells to Jupyter Notebook Explaining Methology
05/13/21	Derek	5.0	Uploaded to GitHub for Grading

Components

This section outlines the key components, processes, and objects used in the project.

#	Name
1	**Reusable Code Objects:** __iter__ getVenueInfo
2	**Libraries:** pandas numpy requests geocoder Nominatim folium os seaborn matplotlib (cm, color, pyplot)
3	**Development Platform:** IBM Watson Studio
4	**API Connectors:** FourSquare API
5	**Internet Hosting:** GitHub
6	**Input Data:** US-CITY-STATE.csv US-COORDINATES.csv NY-COUNTY.csv
7	**Output Data:** BAGEL-RATINGS.csv
8	**Credentials:** FourSquare API Developer

Component Description

Reusable Code Objects

#	Name	Description
1.1	__iter__	This code (within a hidden cell) accesses files within the IBM Cloud Object Storage. It will be used to import the CSV data files into the Jupyter Lab for the project.

```
def __iter__(self): return 0

if os.environ.get('RUNTIME_ENV_LOCATION_TYPE') == 'external':
    endpoint_XXXXXXXXXXXXXXXXXXXX = 'https://s3-api.us-geo.objectstorage.softlayer.net'
else:
    endpoint_XXXXXXXXXXXXXXXXXXXX = 'https://s3-api.us-geo.objectstorage.service.networklayer.com'

client_XXXXXXXXXXXXXXX = ibm_boto3.client(service_name='s3',
    ibm_api_key_id = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url=endpoint_XXXXXXXXXXXXXX)
```

1.2	getVenueInfo	This code passes through parameters, credentials, and requests to the FourSquare API and receives data back from the database.

```
def getVenueInfo(names, latitudes, longitudes, radius=500):

    venues_list=[ ]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&11={},{}radius={}'.format(Client_ID, Client_Secret, Version, lat, lng, radius)

        results = requests.get(url).json()["response"]['groups'][0]['items']

        venues_list.append([(name, lat, lng, v['venue']['name'], v['venue']['id'], v['venue']['categories'][0]['name']) for v in results])

    venue_info = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    venue_info.columns = ['City', 'Latitude', 'Longitude', 'Venue', 'Venue_Id', 'Venue Category' ]

    return(venue_info)
```

Libraries

#	Name	Description
2.1	pandas	pandas is a software library writer for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.
2.2	numpy	NumPy is a software library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
2.3	requests	Requests is a Python HTTP library, released under the Apache License 2.0. The goal of the project is to make HTTP requests simpler and more human-friendly.
2.4	geocoder	Simple and consistent geocoding library written in Python. Deals with a multitude of geocoding providers, such as Google, Bing, OSM & more.
2.5	Nominatim	A tool to search OSM data by name and address and to generate synthetic address of OSM points (reverse geocoding).
2.6	folium	Library that builds on the data wrangling strengths of the Python ecosystem. Simplifies data visualization into interactive leaflet maps, enabling both the binding of data to a map for choropleth visualizations as well as passing rich vector/raster/HTML visualizations as markets on the map. The library has a number of built-in tilesets from OpenStreetMap, Mapbox, and Stamen
2.7	os	A module that provides a portable way of using operating system dependent functionality.
2.8	seaborn	A Python data visualization library based on matplotlib. It provides a high-level interface for drawing informative statistical graphics.
2.9	matplotlib	A comprehensive library for creating static, animated, and interactive visualizations in Python.
2.10	matplotlib.cm	Built-in colormaps, colormap handling utilities, and the ScalarMappable mixin.
2.11	matplotlib.color	A module for converting numbers or color arguments to RGB or RGBA. This module includes functions and classes for color specification conversions, and for mapping numbers to colors in a 1-D array of colors called a colormap.
2.12	matplotlib.pyplot	A state-based interface to matplotlib, mainly intended for interactive plots and simple cases of programmic plot generation

Development Platform

#	Name	Description
3.1	IBM Watson Studio	IBM Watson Studio is an integrated environment designed to make it easy to develop, train, manage models, and deploy AI-powered applications and is a SaaS solution delivered on the IBM Cloud. It is evolving Data Science Experience on IBM Cloud with lot of new features to build AI applications.

API Connectors

#	Name	Description
4.1	FourSquare API (Developers)	An API to access data from FourSquare to get access to global POI data and rich content from over 100k trusted sources and driven by millions of consumers. Connection to the API allows for users to search, discover, and rank venues and get real-time data access.

Internet Hosting

#	Name	Description
5.1	GitHub	A provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuous integration, and wikis for every project.

Input Data

- This module requires geographical location data, specifically of New York State.
- Attributes such as Zip Code, State, City, Latitude, Longitude, County, etc. serve as the main driving force as the key features to answering the Business Problem.

#	Name	Website	Description
6.1	US-CITY-STATE.csv	Open DataSoft	Table containing attributes 'Zip', 'City', 'Timezone', 'Daylight savings time flag', 'Latitude', and 'Longitude'. (data manipulated/divided in CSV to practice merging/appending in Python) Zip = 5 Digit Postal Code State = String Name of State City = String Name of City Timezone = Integer of UTC Timezone Daylight savings time flag = Flag with 0 for False, 1 for True in terms of participation in Daylight Savings Time
6.2	US-COORDINATES.csv	Open DataSoft	Table containing attributes 'Zip', 'City', 'Timezone', 'Daylight savings time flag', 'Latitude', and 'Longitude'. (data manipulated/divided in CSV to practice merging/appending in Python) Zip = 5 Digit Postal Code Latitude = Degree Latitude Coordinates with 6 decimal places Longitude = Degree Longitude Coordinates with 6 decimal places
6.3	NY-COUNTY.csv	Zip Codes to Go	Table containing attributes 'Zip Code', 'City', 'County', and 'Zip Code Map (Link)' Zip Code = 5 Digit Postal Code City = String Name of City County = String Name of County Zip Code Map (Link) = Hyperlink to Google Maps Postal Code Location

Links (since table is picture): (1) Open DataSoft, (2) Open DataSoft, (3) Zip Codes to Go

Output Data

#	Name	Website	Description
7.1	BAGEL-RATINGS.csv	FourSquare API	Table containing attributes 'Venue', 'Rating', 'Ratings Count', and 'Tip'. (data manipulated/divided in CSV to practice merging/appending in Python) Venue = String Name of Individual Venue Rating = Average Rating for a Store out of 10 Ratings Count = The number of ratings contributed to a particular location Tips = The number of tips that were commented on a particular location

Credentials

#	Name	Application	Description
8.1	Client_ID	FourSquare API	A 46-Digit Code that is required to access the FourSquare API. This component acts as the "Username" to access the API.
8.2	Client_Secret	FourSquare API	A 46-Digit Code that is required to access the FourSquare API. This component acts as the "Password" to access the API.

Methodology

Summary

SET-UP

- Import Libraries
- Set-Up Objects
- Import Credentials

DATA MANIPULATION

- Import Data
- Manipulate Data
- Visualize/Evaluate Data
- Readjust Data

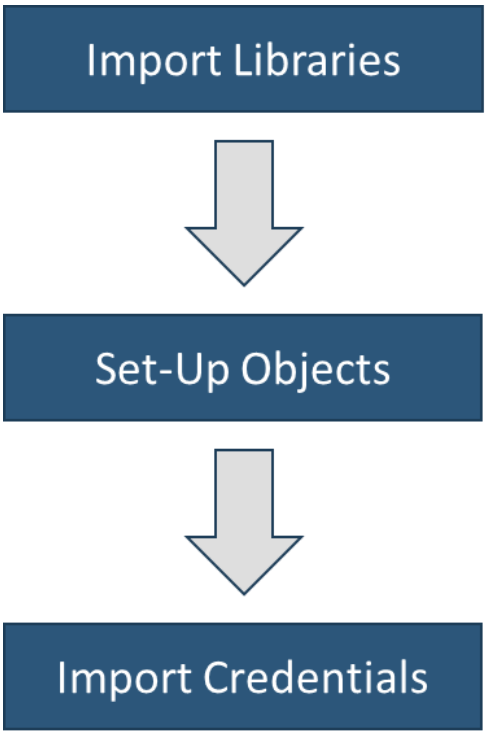
FOURSQUARE API

- Run API

ANALYSIS

- Clean Results
- Evaluate
- Readjust
- Filter
- Append Ratings
- Output 1
- Output 2

Set-Up



pandas, numpy, requests, geocoder, Nominatim, folium, os, seaborn, matplotlib (cm, color, pyplot)

- Import and install all libraries that will be used throughout the Jupyter Notebook.
- Some installs/imports requirements may already be satisfied, but duplication is added to ensure all proper libraries are present and can be referred to.

__iter__ , getVenueInfo

- Set up "Objects" or lines of code that can be called with different variables passed through each time.
- The Objects are…
  - to import Data into the module
  - to proper use the FourSquare API to collect information and pass it into the module

Client_ID, Client_Secret

- Credentials to be used to access the FourSquare API

Data Manipulation

Import Data

US-CITY-STATE.csv as df_citystate
US-COORDINATES.csv as df_coordinates

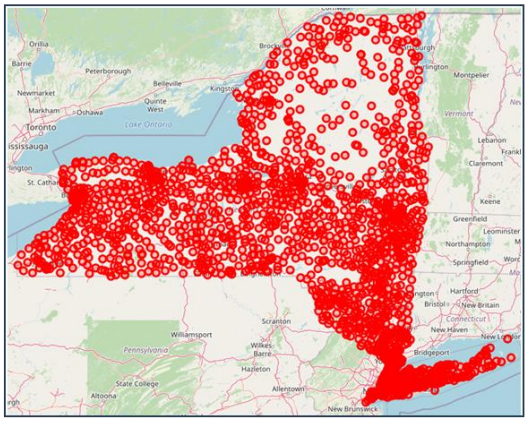
- Data from the two CSV files is imported into the Jupyter Notebook as DataFrames.

Manipulate Data

Join df_citystate + df_coordinates on Zip Code

- A glimpse into all of the Cities of New York that are within the DataFrame.

Visualize/Evaluate Data

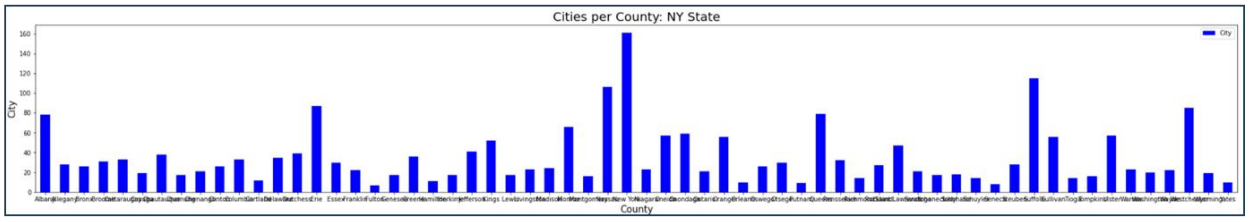


- A glimpse into all of the Cities of New York that are within the DataFrame.

Readjust Data

NY-COUNTY.csv as df_ny_county

- Data imported into the Jupyter Notebook as DataFrame.
- Reset index
- Join new DataFrame with existing DataFrame on Zip Code (adding in County)
- Cities per County graph for visualization purposes:

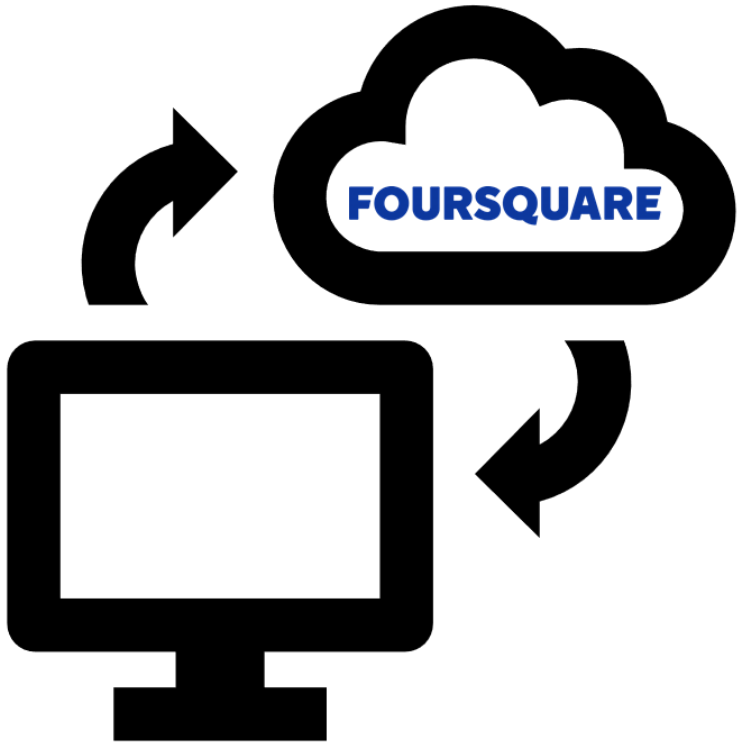


- Reset index
- Drop N/A values in 'County'
- Create a new DataFrame that only contains "New York", "Queens", "Nassau", and "Suffolk"

FourSquare API

Run API

Run Object “getVenueInfo” to use the API and retrieve data from FourSquare



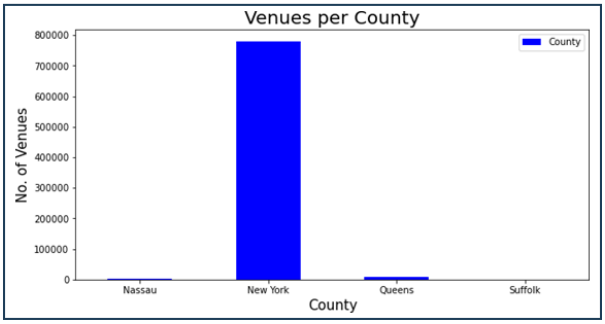
Analysis

Clean Results

Add in missing components, such as County.

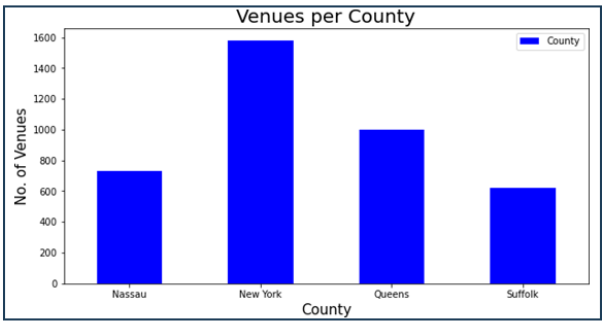
Evaluate

- Taking a view of "Venues per County".
- It appears that some cities may have had too large of a radius when grabbing from the API, so venues can be counted multiple times.



Readjust

- Duplicate venues removed
- Checking on the current venue allocation, appears to be more accurate.



Readjust Data

df_Bagel as df Venue Category = 'Bagel Shop'

- Pulling Bagel Shops for New York, Queens, Nassau, and Suffolk County

Manipulate Data

Join df_NYC_Bagel + df_Ratings on Venue

- Import CSV file that previously was downloaded using the FourSquare API function

Output 1

Top 5 Bagel Shops in New York by Rating

- A table with Venue name and Rating in order of Rating from top to bottom

Output 2

Map of all Bagel Shops locations in New York

- A map with datapoints of Bagel Shops location in New York

End

Results

Output 1

Top 5 Bagel Shops in New York by Rating

	Venue	Rating
0	Sadelle's	9.1
1	Absolute Bagels	9.1
2	Tompkins Square Bagels	9.0
3	Bo's Bagels	9.0
4	Best Bagel & Coffee	9.0

A table with Venue Name and Rating in order of Rating from Top to Bottom

Output 2

Bagel Shop Locations in New York City

A map with datapoints of Bagel Shop locations in New York

Discussion

Observation

- The code works! We have successfully discovered the Top 5 Bagel Shops in New York City by FourSquare Rating.
- Additionally, we have made a successful interactive map.
- Objects can be reused in this code to pass different variables through. For example, different locations or different types of venues to rank would also work within this code.

Recommendations

- As a recommendation for improvement, I would really like to use an upgraded version of FourSquare API in order to collect a vaster amount of data.
- With an increased amount of data, we would be able to use some machine learning algorithms to really crack into different types of analyses.
- Also, assuming more data present, you can refine results. For example, in one data set, it has the number of ratings that make up the average rating of a venue. You can set this to >30 for number of ratings to ensure an accurate representation of the rating is present.

Conclusion

- If you're visiting the New York Area, visit Sadelle's or Absolute Bagels for the best selection of Bagels!