Capstone project

For

IBM Data Science Specialization

**Urban Happiness**

Ayushi Sharma

[sharmayu20@gmail.com](mailto:sharmayu20@gmail.com)

**Introduction/Business Problem**

Relocating from one place to another is in itself a very difficult task for a person, he/she has to decide whether the neighborhood has the required venues that he/she likes and on top of that whether the neighborhood is safe.

What if a family man wants to move and is unaware of the environment around his selected neighborhood? This is where the project Urban Happiness comes in.

Urban Happiness is a project written in Python that can help a user decide his/her favorable neighborhood based on the venue that he requires and keeping in mind the crime rate of the neighborhood in city of San Francisco.

Urban Happiness presents a map to the user with the crime ranges of the locality and the clustered neighborhood based on the venues. It also outputs the closest neighborhood from the venue given by the user.

The map marks two neighborhoods that are close to the venue given by the user.

The project clusters the neighborhoods based on all the venues and the crime rate in that neighborhood into three clusters and shows it to the user via a map.

**Data**

The data that will be required for the project would be the crimes rate in San Francisco based on the neighborhoods, the json file that contains the coordinates of the neighborhoods in the form of a polygon which can then be used by folium to mark the neighborhoods, then the postal code or the pincode of the neighborhoods which is then used to gather the exact latitude and the longitude of the neighborhood, then the venue list that is present in the given locality which can be fetched using the foursquare API.

The following section has a detailed description of all the data that will be used for the completion of the project.

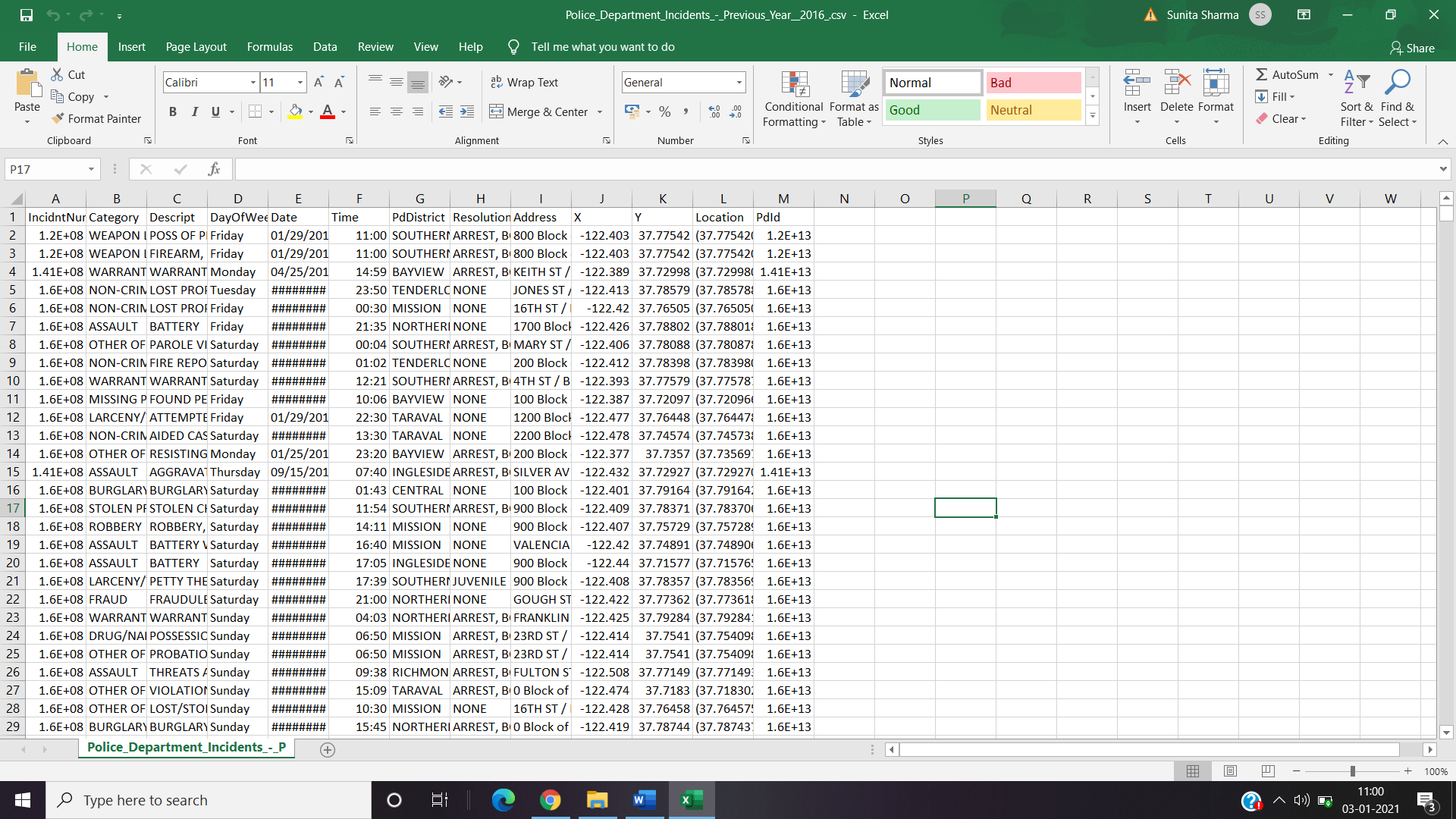
1. Crime rate in San Francisco

This data file is a csv file that contains the arrests or crimes that were committed in different neighborhood of San Francisco.

The important columns that are considered for the project are:

* District
* Count of crimes in each district

Snapshot of the csv file used.

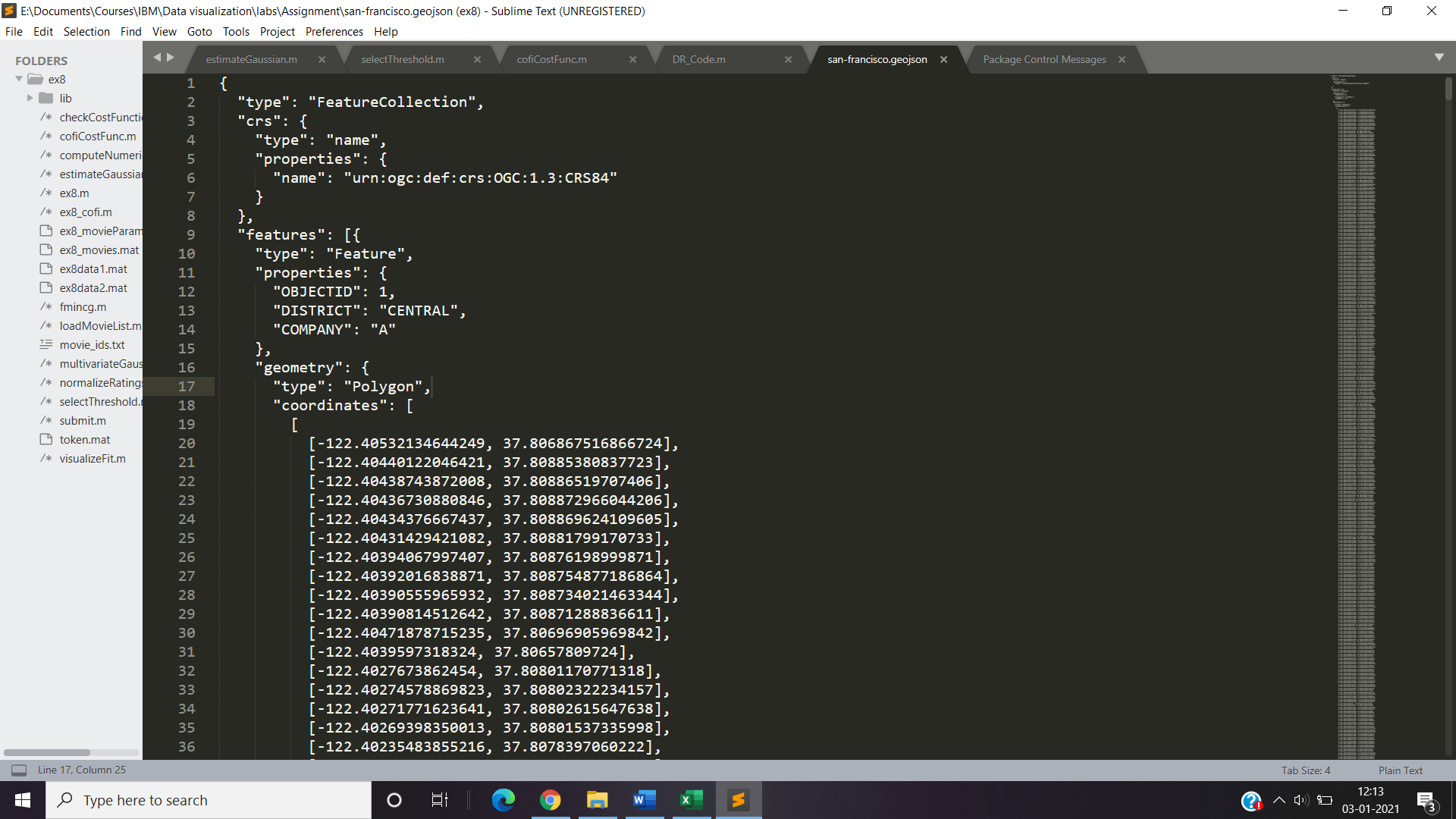


1. GeoJson file that contains information on San Francisco

This GeoJson file contains the information on San Francisco in the form of key value pairs. The file contains the coordinates of different neighborhoods in San Francisco. These coordinates are given to the folium’s geo\_data attribute of the Chloropleth class.

This geo\_data is responsible for marking the districts and the colors for the folium map.

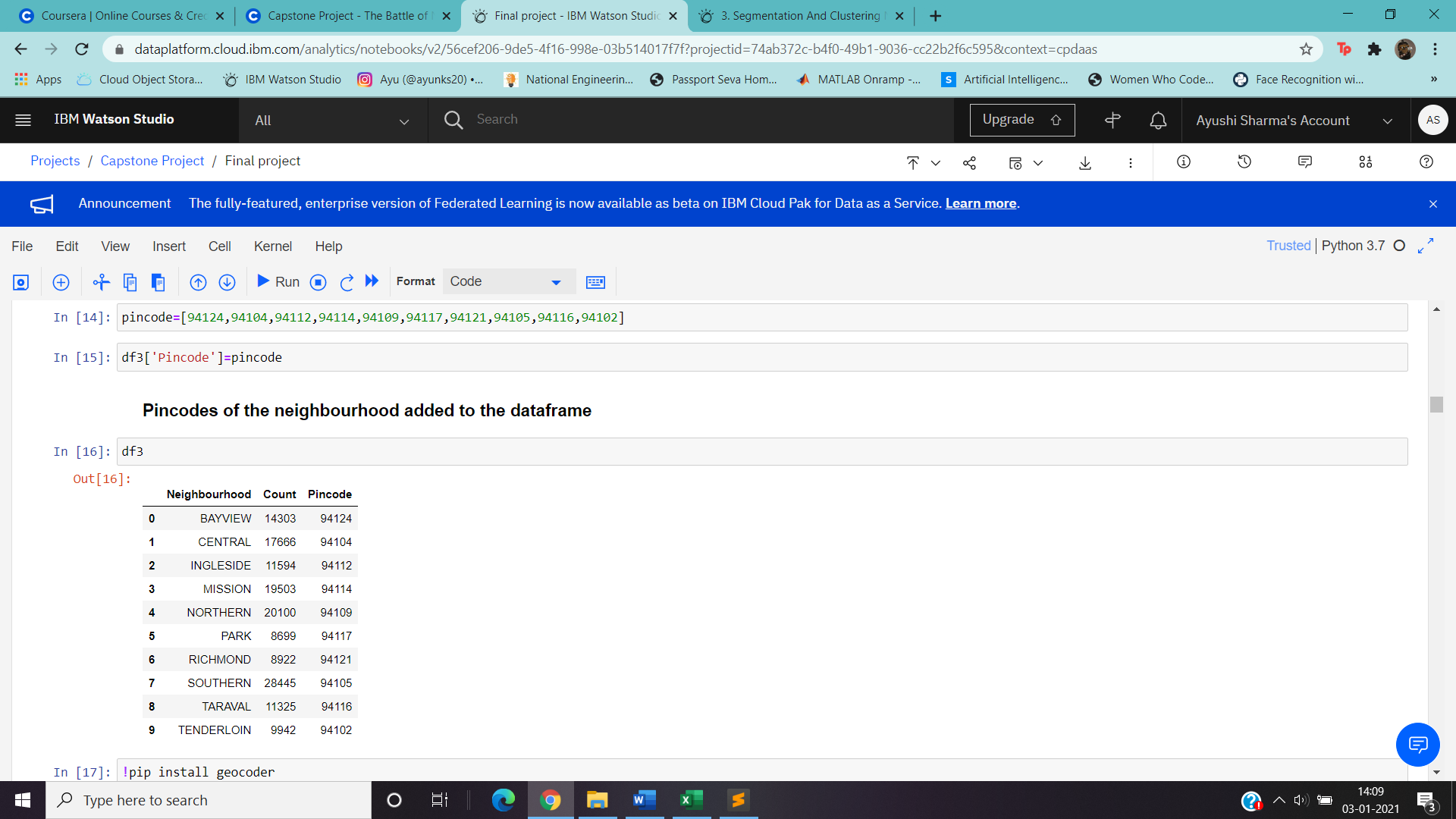
Snapshot of the geoJson file



1. Postal codes

Since there was no website that provided the postal codes of the neighborhood, the postal codes were manually added to the data frame by adding another column to the data frame. Postal codes were manually searched from the in internet and added to the data frame.

Snapshot of the modified data frame.

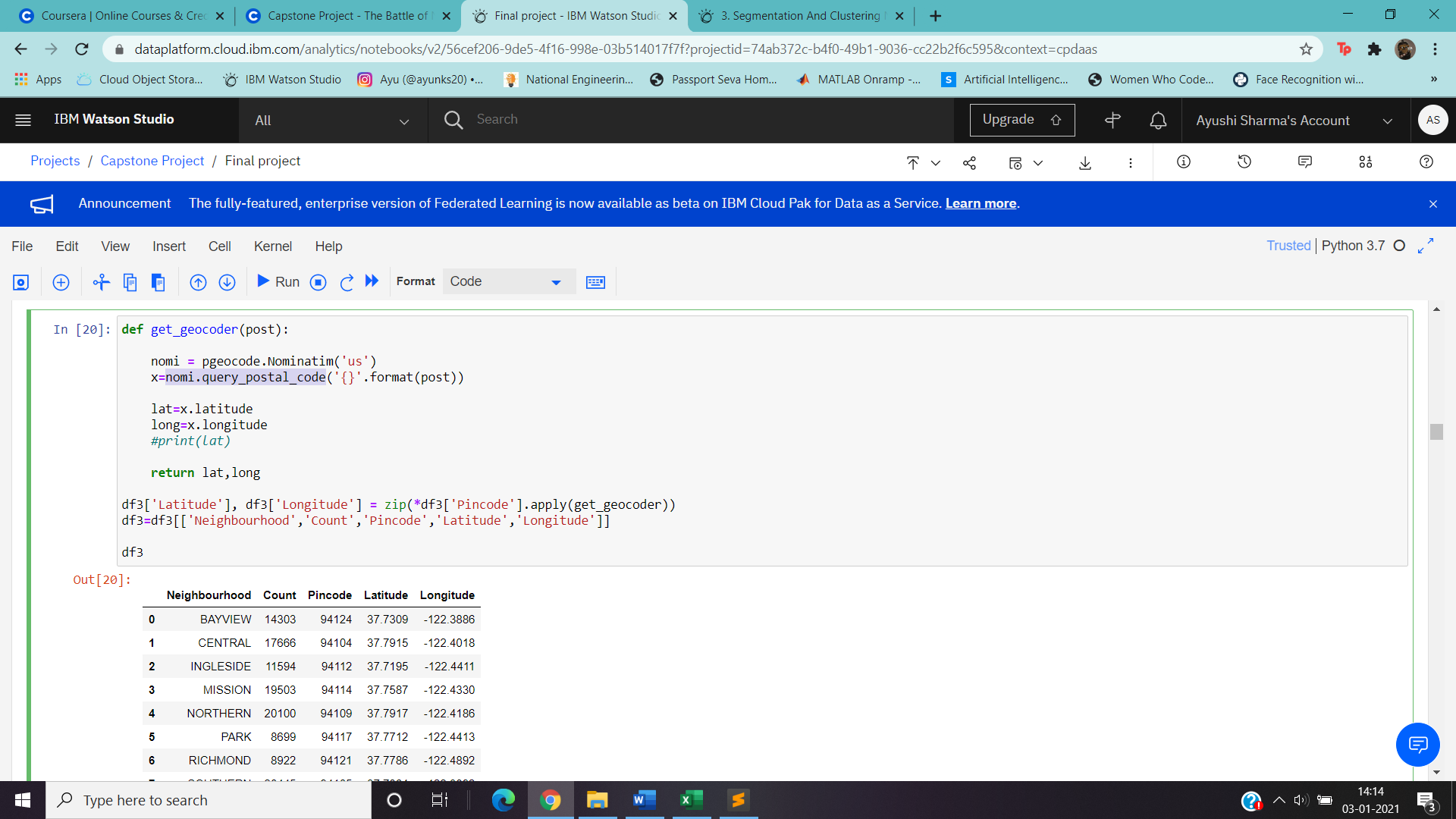


1. GeoCoder:

Geocoder package is used to fetch the latitudes and the longitudes of place by passing the postal code to it.

This package has a function nomi.query\_postal\_code() which takes postal code as a input and outputs the longitude and the latitude of the place

Snapshot of the function used to access the longitude and the latitude and the result.



1. FourSquare API:

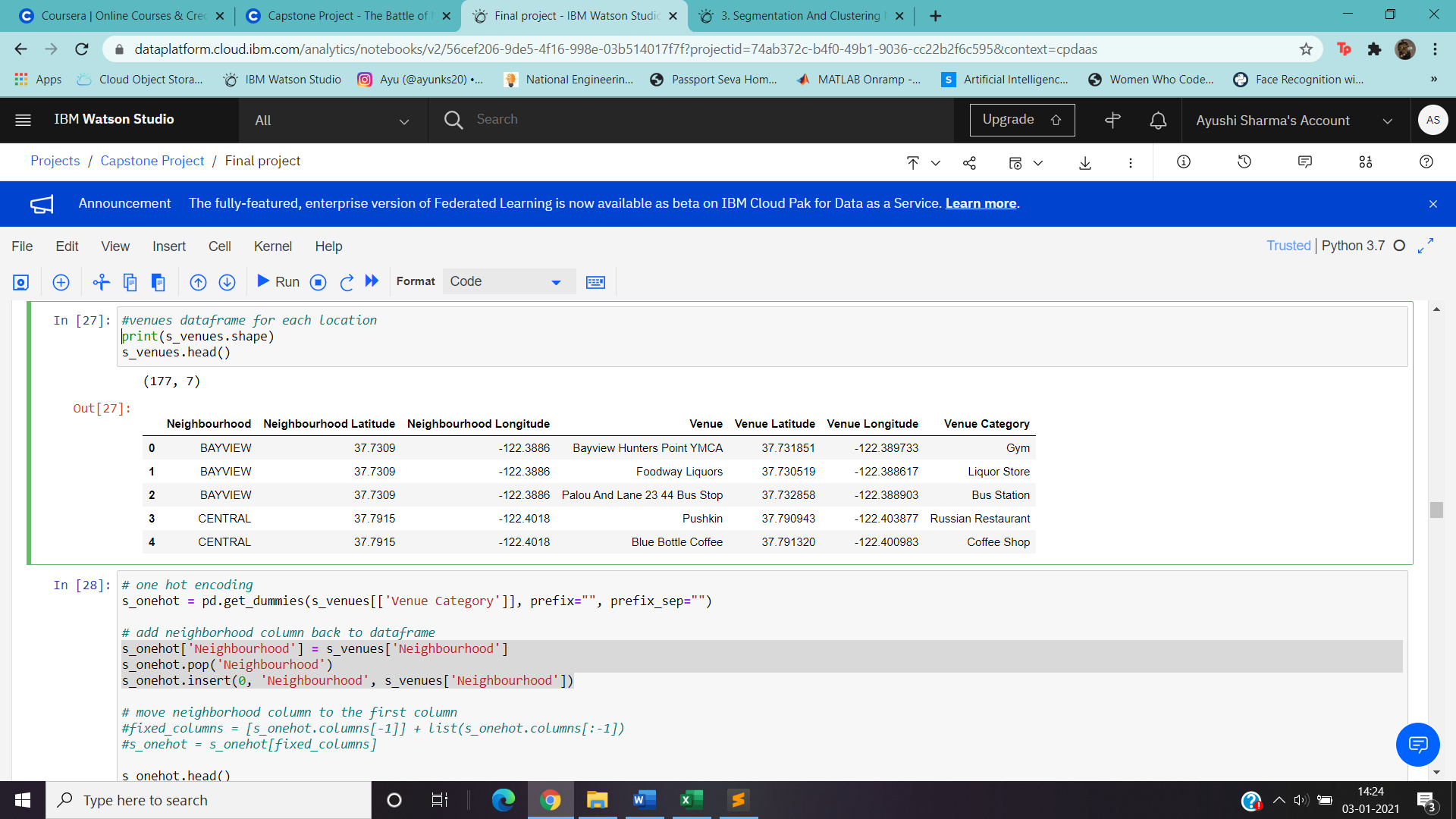
The foursquare API is used to fetch the list of venues that are close to the given latitude and the longitude. The API uses the client ID and the Client Secret to fetch the details.

The url is then used on a get request method to the API, the url contains the client id, client secret, version of the Foursquare, latitude and longitude of the location, radius to be considered around the location and the limit as to fetch how many venues around the location.

The response is then stored in the form of json object. The response can contain details of the venue such as name, latitude and longitude of the venue, category of the venue, or rating or tip of the venues.

The response can then be converted to a pandas data frame and then be used for further operations.

Snapshot of the result after converting to pandas data frame:



|  |
| --- |
|  |