Location Searching based on features of choice and distance

Nicolas Shattaoula

February 21st 2020

Introduction:

Nowadays people are looking to rent a house, pick a location for a business in their hometown, country or overseas.

Problem:

How will they make this choice in terms of defining the closest place to the features they would choose?

Interest:

This will be useful for anyone that would want to select a location for any purpose, buying/renting a house starting business or even selecting some location based on the least of distances of the locations he or she desires.

Data acquisition and cleaning:

Source:

I will use Foursquare Api to derive all the location data of the locations

Data cleaning :

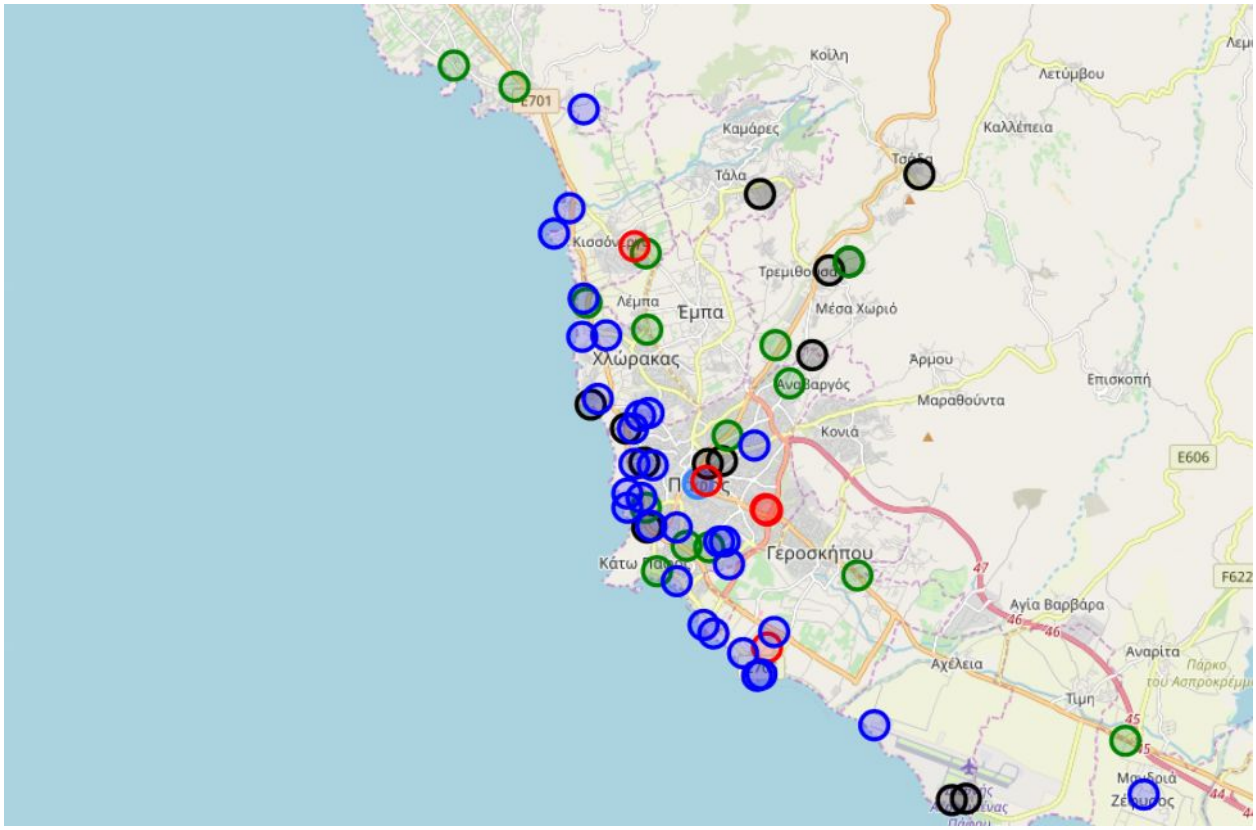
Using pandas dataframe we will shape the data into a dataframe of combinations of the various locations.

Feature selection:

We will use shapely and polygon in order to gauge distance and select the top results with the least amount of distance.

Exploratory Data Analysis:

After gathering all the needed data we can use folium to visualize all the nearby locations (colored by category)



Now the challenge is how to define the methodology to gauge what will be the metric to choose the location.
After we gathered the data and used a full outer join of coordinates of the categorial locations we created a table of combinations of coordinates, from there will estimate the distance by calculating the perimeter of the polygon that is constructed from the locations.

Methodology

We use shapely and length of the polygon to achieve this

This is to produce a dataframe with all the features combinatios!

15]:
```
#GroceryCoordinates.join(SwimmingPoolCoordinates, how='outer')
GroceryCoordinates['key'] = 0
SwimmingPoolCoordinates['key'] = 0
StadiumPoolCoordinates['key'] = 0
SchoolPoolCoordinates['key'] = 0

df_cartesian = GroceryCoordinates.merge(SwimmingPoolCoordinates, how='outer')
df1_cartesian= df_cartesian.merge(StadiumPoolCoordinates, how='outer')
coordinates_combinations= df1_cartesian.merge(SchoolPoolCoordinates, how='outer')
```

16]:
```
from shapely.geometry import Polygon
```

Create a function that will calculate the perimeter of the polygon of the 4 features!

22]:
```
def calculate_length(row):
    return Polygon([(row['locationlatGrocery'], row['locationlngGrocery']), (row['locationlatSwim'], row['locatio
```

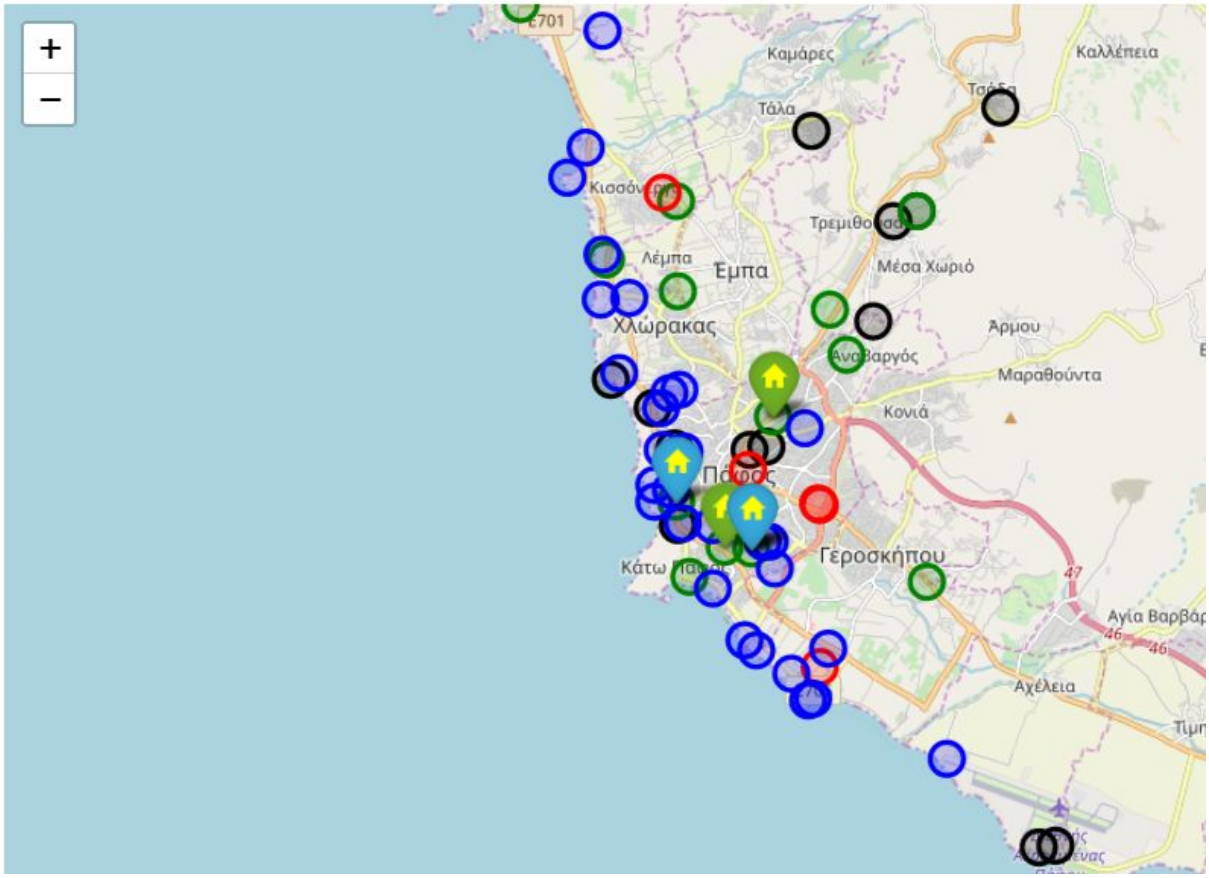
We apply the function to calculate the perimeter of each group of features,and choose some top results!

23]:
```
coordinates_combinations['area'] = coordinates_combinations.apply(calculate_length, axis=1)
Top10=coordinates_combinations.sort_values('area', ascending=True).head(9)
```

Prediction/Results of Model:

```
# display map
venues_map
```

7]:



In this project we defined locations based on least distances between features of our choosing and visualised it

Future directions:

Use a different approach to select the metric, like finding the centroid of the locations and take the average distance between them (maybe using knn algorithm) . Also we can use higher complexity to the problem by adding weight of feature i.e. its more important to have a school closed than a church, or impose a limit in the maximum distance of the amenity.