Coursera Capstone Project: IBM Data Science

Mwimbi Shindola
mwimbi.shindola@gmail.com
Lusaka, Zambia

1. Introduction and Business Problem

Lusaka is the capital city of Zambia and is the Largest city by population in the county. As of 2020 It has a population of about 2,774,000 people according to Macprotrends.net. Being the most populous city there is need to properly plan how institutions of learning are located. One of the challenges parents face with growing traffic is finding the nearest schools to take their children to. This project intends to identify areas where schools are located further away. Information from this research can be used by families with school going children when planning which areas to settle, the government when identifying which areas to build more schools or private investors who may intend to venture into private schools to bridge the gap in neighborhoods without schools.

2. Data

This section describes how data for the project was retrieved and processed. Multiple sources and methods where used to make the data usable for analysis. The following subsections sections describe the data collected and how it was processed at different levels.

2.1. Neighborhoods

The data of the neighborhoods in Lusaka was collected from 2 different sources: Wikipedia page and macpro.info website then created a csv list of the neighborhoods was created and uploaded as a data asset on the Watson IBM project platform. This CSV file was then used to create a neighborhoods data frame in the project notebook.

2. Import CSV file and create a Dataframe for Lusaka Locations uploaded a the project assests

```
# @hidden_cell
# The project token is an authorization token that is used to access project resources like data sources, connecti
from project_lib import Project
project = Project(project_id='0fb9cca2-467f-4596-8794-c01c4453f927', project_access_token='p-078e0048fb252b796908
pc = project.project_context
```

```
# Fetch the file
my_file = project.get_file("Lusaka_neighborhoods.csv")

# Read the CSV data file from the object storage into a pandas DataFrame
my_file.seek(0)
lsk_df=pd.read_csv(my_file, ) #nrows=10
```

2.2. Geocoding

The file contents from Lusaka_nieghborhoods.csv are retrieved into a Pandas DataFrame. The latitude and longitude of the neighborhoods are retrieved using Google Maps Geocoding API. The geometric location values are then stored into the a dataframe which was later merged with the neighborhoods list dataframe.

3. Get the geographical coordinates of Lusaka Neighborhoods

```
# define a function to get coordinates
def get_latlng(neighborhood):
    # initialize the variable to None
    lat_lng_coords = None
    # loop until you get the coordinates
    while(lat_lng_coords is None):
        g = geocoder.arcgis('{}, Lusaka, Zambia'.format(neighborhood))
        lat_lng_coords = g.latlng
    return lat_lng_coords
```

```
# call the function to get the coordinates, store in a new list using list comprehension
coords = [ get_latlng(neighborhood) for neighborhood in lsk_df["Neighborhood"].tolist() ]
```

2.3. Venue Data

From the location data obtained, the venue data is found out by passing in the required parameters to the FourSquare API, and creating another DataFrame to contain all the venue details along with the respective neighborhoods.

Code

```
limit=100
radius=2000

search_query='School'
column_names=['Neighborhood','Latitude','Longitude','School','School_Latitude','School_Longitude','School_categor
mydf=pd.DataFrame(columns=column_names)
for name, lat, lng in zip(lsk_df['Neighborhood'],lsk_df['Latitude'],lsk_df['Longitude']):
    url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radiu
    results = requests.get(url).json()
    k=results['response']['venues']
    for school in k:
        if len(school['categories'])>0:
            mydf=mydf.append({'Neighborhood':name,'Latitude':lat,'Longitude':lng,'School':school['name'],'School_
        else:
            mydf=mydf.append({'Neighborhood':name,'Latitude':lat,'Longitude':lng,'School':school['name'],'School_
```