1/22/24, 12:56 AM EDA

Calcular Distancias de Escuelas a establecimientos que venden alcohol mas cercanos

Chequen los PATHS

Primero se importan los scripts y scipy

```
In [ ]: from scripts.filter import filtrar_dataframe
    from scipy.spatial import KDTree
    import pandas as pd
```

Seccion de PATHS

```
In [ ]: # Lista de rutas de archivos Negocios DENUE
        paths negocios denue = [
            # AQUI AGREGUEN LOS PATHS DE SUS CSVs
            '/home/jay/repos/AI/feminicidios/denue_00_46111_csv/conjunto_de_datos/de
            '/home/jay/repos/AI/feminicidios/denue 00 46112-46311 csv/conjunto de da
            '/home/jay/repos/AI/feminicidios/denue_00_46321-46531_csv/conjunto_de_da
            '/home/jay/repos/AI/feminicidios/denue_00_46591-46911_csv/conjunto_de_da
            # Sin querer puse el PATH de las escuelas, xd que bueno que despues de q
            # Lo deje para debuggear
            # '/home/jay/repos/AI/feminicidios/denue 00 61 csv/conjunto de datos/den
        # Ruta de archivo Escuelas DENUE
        file_path_escuelas = '/home/jay/repos/AI/feminicidios/denue_00_61_csv/conjun
        # Ruta de archivo salud DENUE
        file_path_salud = '/home/jay/repos/GeoStats/denue/denue_00_62_csv/conjunto_d
        # Ruta de archivo parques DENUE
        file_path_parques = '/home/jay/repos/GeoStats/denue/denue_00_71_csv/conjunto
        # Ruta de archivo mayoreo DENUE
        file_path_mayoreo = '/home/jay/repos/GeoStats/denue/denue_00_43_csv/conjunto
```

Dado una lista de terminos en una lista de columnas se filtra.

```
In [ ]: columnas_para_filtrar = ['nombre_act', 'raz_social', 'nom_estab']
```

1/22/24, 12:56 AM EDA

```
In [ ]: # Filtramos escuelas
        df escuelas = filtrar dataframe(file path escuelas, columnas para filtrar, [
In [ ]: # Filtramos parques
        terminos_parques = ['parques', 'jardin', 'plaza', 'bosque']
        df parques = filtrar dataframe(file path parques, columnas para filtrar, ter
In [ ]: # Salud es complicado, porque queremos hospitales, publicas y privadas separ
        terminos_privadas = ['consultorio', 'privado']
        terminos_publicas = ['imss', 'publico', 'isste']
        terminos_hospitales = ['hospital', 'cirugia']
        df salud privadas = filtrar dataframe(file path salud, columnas para filtrar
        df salud publicas = filtrar dataframe(file path salud, columnas para filtrar
        df_salud_hospitales = filtrar_dataframe(file_path_salud, columnas_para_filtr
In []: # Para los negocios de alcohol tenemos que unir todos los dataframes al fina
        terminos_alcohol = ['alcohol', 'licor', 'cerveza', 'bar', 'cantina', 'vinos'
        df alcohol list = []
        for path in paths negocios denue:
            df filtrado = filtrar dataframe(path, columnas para filtrar, terminos al
            df_alcohol_list.append(df_filtrado)
        df_alcohol = pd.concat(df_alcohol_list, ignore_index=True)
In [ ]: # Fabricas
        terminos fabricas = ['fabrica', 'fabricacion', 'industria', 'industrias', 'i
        df_fabricas = filtrar_dataframe(file_path_mayoreo, columnas_para_filtrar, te
```

EDA

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))

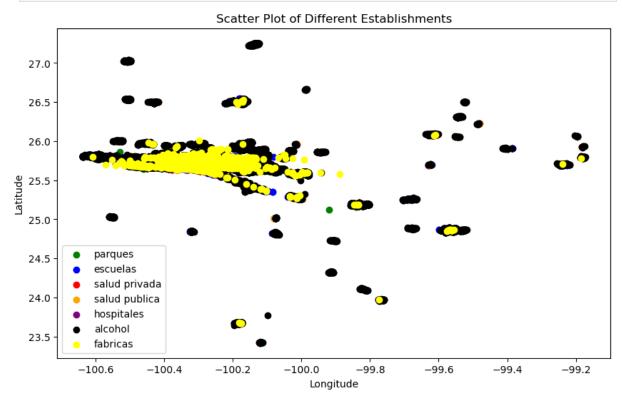
# Plot each dataframe with a different color
plt.scatter(df_parques['longitud'], df_parques['latitud'], color='green', la
plt.scatter(df_escuelas['longitud'], df_escuelas['latitud'], color='blue', l
plt.scatter(df_salud_privadas['longitud'], df_salud_privadas['latitud'], col
plt.scatter(df_salud_publicas['longitud'], df_salud_publicas['latitud'], col
plt.scatter(df_salud_hospitales['longitud'], df_salud_hospitales['latitud'],
plt.scatter(df_alcohol['longitud'], df_alcohol['latitud'], color='black', la
plt.scatter(df_fabricas['longitud'], df_fabricas['latitud'], color='yellow',

# Add a title and labels
plt.title('Scatter Plot of Different Establishments')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
```

1/22/24, 12:56 AM EDA

```
# Add a legend to identify the datasets
plt.legend()

# Show the plot
plt.show()
```



Todavia no queda, esta en testing

Se crea un KDTree

Se guarda el indice y la distancia del establecimiento mas cercano a la escuela

```
In []: # Encuentra el establecimiento más cercano a una coordenada dado un arbol de
    def find_nearest(row, tree, establishments):
        distance, index = tree.query([row['latitud'], row['longitud']])
        return establishments.iloc[index]['nom_estab'], distance

# Tree para aplicar busqueda
    alcohol_tree = KDTree(alcohol[['latitud', 'longitud']])

# simon
    schools['nearest_alcohol'], schools['distance_to_nearest_alcohol'] = zip(*sc
```