

IMPLEMENTACIÓN K-MEANS

REDUCCIÓN DEL ARCHIVO ORIGINAL

Archivo: source/reducirCSV.ipynb

```

import pandas as pd
data = pd.read_csv('yellow_tripdata_2009-12.csv')
data = data.drop(["vendor_name", "Trip_Pickup_DateTime", "Trip_Dropoff_DateTime",
"Passenger_Count", "Trip_Distance", "Rate_Code", "store_and_forward", "End_Lon",
"End_Lat", "Payment_Type", "Fare_Amt", "surcharge", "mta_tax", "Tip_Amt", "Tolls_Amt",
"Total_Amt"], axis = 1)
data = data.iloc[:500000,]
print(data.info())
data.to_csv("../build/datos_resumidos.csv", index=False, encoding='utf8')

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500000 entries, 0 to 499999
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Start_Lon    500000 non-null  float64
1   Start_Lat    500000 non-null  float64
dtypes: float64(2)
memory usage: 7.6 MB
None

```

GENERACIÓN DE CLUSTER CON K-MEANS

Archivo: source/main.cpp

```

Consola de depuración de Microsoft Visual Studio
Insertar cantidad de ITERACIONES
5
Insertar cantidad de CLUSTERS
5
D:\Documentos\UCSP\CCOMP\2021-01\ESTRUCTURA de Datos Avanzadas\Laboratorio\TRABAJOS_github\EDA_trabajos\K means_implemen
tation\build\Debug\k_means.exe (proceso 12916) se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .

build > coordenadas.csv
1 longitud,latitud,agrupacion
2 -73.9879,40.7379,0
3 -73.956,40.7796,4
4 -73.9557,40.6895,3
5 -73.984,40.7546,2
6 -73.9591,40.7693,4
7 -73.9822,40.7831,2
8 0,0,4
9 -73.9821,40.7768,2
10 -73.9848,40.7419,1
11 0,0,4
12 -73.9912,40.73,0

```

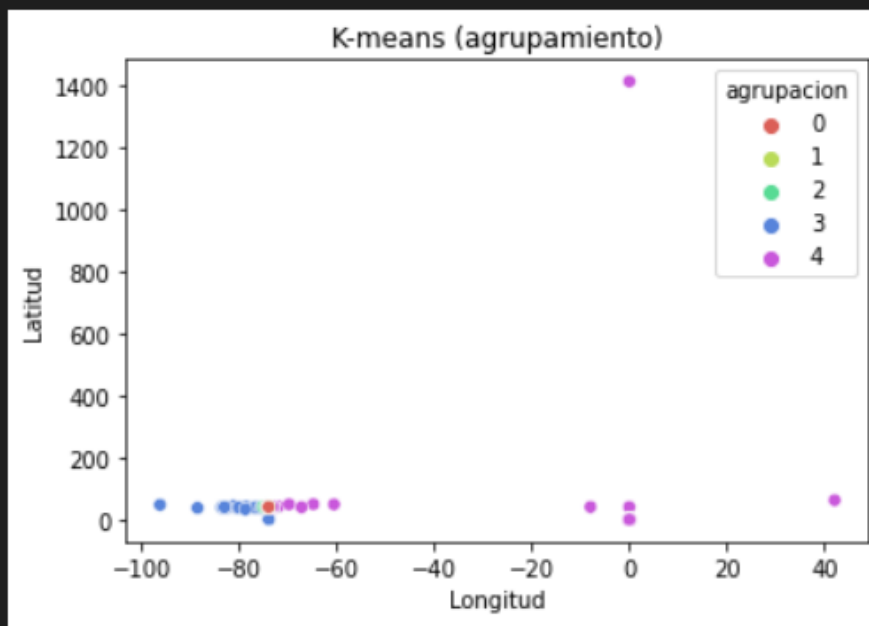
VISUALIZACIÓN DE RESULTADOS

Archivo: source/visualización.ipynb

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

plt.figure()
df = pd.read_csv("../build/coordenadas.csv")
sns.scatterplot(x=df.longitud, y=df.latitud,
                hue=df.agrupacion,
                palette=sns.color_palette("hls", n_colors=5))
plt.xlabel("Longitud")
plt.ylabel("Latitud")
plt.title("K-means (agrupamiento)")

plt.show()
```



OTRAS PRUEBAS

Cantidad de datos: 1000000.

```
import pandas as pd
data = pd.read_csv('yellow_tripdata_2009-12.csv')
data = data.drop(["vendor_name", "Trip_Pickup_DateTime", "Trip_Dropoff_DateTime",
"Passenger_Count", "Trip_Distance", "Rate_Code", "store_and_forward", "End_Lon",
"End_Lat", "Payment_Type", "Fare_Amt", "surcharge", "mta_tax", "Tip_Amt", "Tolls_Amt",
"Total_Amt"], axis = 1)
data = data.iloc[:1000000,]
print(data.info())
data.to_csv("../build/datos_resumidos.csv", index=False, encoding='utf8')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000000 entries, 0 to 999999
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Start_Lon    1000000 non-null  float64
1   Start_Lat    1000000 non-null  float64
dtypes: float64(2)
memory usage: 15.3 MB
None
```

10 iteraciones y 10 clúster. Visualizando con otra gráfica.

```
plt.figure()
df = pd.read_csv("../build/coordenadas.csv")
sns.relplot(x="longitud", y="latitud", data = df, kind = "scatter",
            hue="agrupacion", alpha = 0.4)
#sns.scatterplot(x=df.longitud, y=df.latitud,
#                hue=df.agrupacion,
#                palette=sns.color_palette("hls", n_colors=10))
plt.xlabel("Longitud")
plt.ylabel("Latitud")
plt.title("K-means (agrupamiento)")

plt.show()
```

<Figure size 432x288 with 0 Axes>

