ÍNDICE GENERAL

ÍNDICE DE FIGURAS

TRABAJO CDR'S DE MILÁN

Para profundizar en el uso de MongoDB para el tratamiento de datos procedentes de los registros de datos o CDR (Call Detail Record) almacenados por las operadoras de telecomunicaciones, utilizaremos los datos de la ciudad de Milán tomados por Telecom Italia desde el 1 de noviembre de 2013 hasta el 1 de enero de 2014 disponibles en el siguiente enlace:

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EGZHFV

Cada alumno elegirá una semana completa de datos que incluya la totalidad de los datos tomados de lunes a domingo y realizará las siguientes acciones:

Ejercicio 1

Descargar los ficheros ".txt" correspondientes y crear un programa en Python para cargar los datos en la colección "Milan_CDR_c" de la base de datos "Milan_CDR_db".

He seleccionado la semana del 23/12/2013 al 29/12/2013.

```
import os
import numpy as np
import json
import pymongo
import warnings
warnings.filterwarnings('ignore')

FILES_PATH = r"C:/Users/carlo/Desktop/Carlos/MasterLocal/ADPLocal/JSONs/"
client = pymongo.MongoClient('mongodb://localhost:27017')
database = client['Milan_CDR_db']

collist = database.list_collection_names()
if "Milan_CDR_c" in collist:
    print("The collection Milan_CDR_c exists.")
    Milan_CDR_c = database.get_collection("Milan_CDR_c")
else:
    # Primero creo la colección Milan_CDR_c
    database.create_collection("Milan_CDR_c")

    Milan_CDR_c = database.get_collection("Milan_CDR_c")
```

```
    for file in os.listdir(FILES_PATH):
        # Inserto cada archivo en la colección
        try:
            with open(FILES_PATH + file, encoding="utf-8") as f:
                file_data = json.load(f)
                Milan_CDR_c.insert_many(file_data)
        except UnicodeDecodeError as e:
            print(f"Error de codificación en el archivo: {file} - {e}")
        except json.JSONDecodeError as e:
            print(f"Error en el formato JSON del archivo: {file} - {e}")
```

Figura 1: Código Cargar Datos en Colección

Ejercicio 2

Encuentra los países con los que se interactúa.

Se interactúa con los siguientes países: [0, 1, 7, 20, 27, 30, 31, 32, 33, 34, 36, 39, 40, 41, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 64, 65, 66, 81, 82, 84, 86, 90, 91, 92, 93, 94, 95, 98, 211, 212, 213, 216, 218, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 260, 261, 262, 263, 264, 265, 267, 291, 297, 298, 299, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 370, 371, 372, 373, 374, 375, 376, 377, 378, 380, 381, 382, 385, 386, 387, 389, 420, 421, 423, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 670, 672, 673, 675, 676, 677, 678, 679, 680, 681, 685, 686, 687, 689, 690, 852, 853, 855, 856, 870, 880, 881, 886, 960, 961, 962, 963, 964, 965, 966, 967, 968, 970, 971, 972, 973, 974, 975, 976, 977, 992, 993, 994, 995, 996, 998, 1204, 1214, 1226, 1242, 1246, 1250, 1268, 1289, 1306, 1340, 1345, 1403, 1416, 1418, 1438, 1450, 1473, 1506, 1514, 1519, 1579, 1587, 1604, 1613, 1647, 1649, 1670, 1671, 1684, 1705, 1709, 1721, 1758, 1767, 1778, 1780, 1784, 1787, 1807, 1808, 1809, 1819, 1829, 1849, 1902, 1905, 1907, 1924, 1927, 1929, 1939, 7700, 7701, 7702, 7705, 7707, 7711, 7712, 7713, 7714, 7717, 7725, 7726, 7727, 7728, 7771, 7775, 7776, 7777, 7778, 8816, 12684, 12687, 14413, 14415, 18092, 18093, 18094, 18096, 18097, 18098, 18099, 18682, 18683, 18684, 18762, 18763, 18764, 18765, 18768, 18769, 29774, 50931, 50936, 50937, 50938, 50947, 88216, 88232, 88239, 97259]

```
cliente = MongoClient("mongodb://localhost:27017/")
db = cliente["Milan_CDR_db"]
```

```
coleccion = db["Milan_CDR_c"]

pipeline = [
    {"$group": {"_id": "$Country_code"}},
]

diff_countries = coleccion.aggregate(pipeline)
diff_countries = [doc['_id'] for doc in diff_countries]

diff_countries = sorted(diff_countries)

print("Países con los que se interactúa:", diff_countries)
```

Figura 2: Código Encontrar Países

Ejercicio 3

Encuentra que país es con el que más se interactúa además de Italia

País con el que más se interactúa excluyendo Italia: 49, Alemania.

Para lograr obtener este resultado, primero, el operador $match elimina los documentos con los códigos excluidos. Luego, $group agrupa los datos por Country_code, contando la cantidad de interacciones para cada país. A continuación, $sort ordena los resultados en orden descendente según la cantidad de interacciones. Finalmente, $limit selecciona solo el país con más interacciones, el primero de todos ya que hemos ordenado con sort.

```
codes_excluded = [39, 0]

cliente = MongoClient("mongodb://localhost:27017/")
db = cliente["Milan_CDR_db"]
coleccion = db["Milan_CDR_c"]

pipeline_ej3 = [
    {"$match": {"Country_code": {"$nin": codes_excluded}}},
    {"$group": {"_id": "$Country_code", "count": {"$sum": 1}}},
    {"$sort": {"count": -1}},
    {"$limit": 1}
]

print(list(coleccion.aggregate(pipeline_ej3))[0]['_id'])
```

Figura 3: Código Segundo País con más Interacciones

Ejercicio 4

¿Qué celda comunica más con el extranjero?

La celda 6165.

```
from bson.son import SON
from pymongo import MongoClient
cliente = MongoClient("mongodb://localhost:27017/")
db = cliente["Milan_CDR_db"]
coleccion = db["Milan_CDR_c"]

codes_excluded = [39, 0]

pipeline_ej4 = [
    {"$match": {"Country_code": {"$nin": codes_excluded}}},
    {"$group": {"_id": "$Square_id", "count": {"$sum": 1}}},
    {"$sort": SON([("count", -1)])},
    {"$limit": 1}
]

print(list(coleccion.aggregate(pipeline_ej4))[0]['_id'])
```

Figura 4: Código Celda Extranjero

Ejercicio 5

Encuentra la celda con más actividad de smsin, smsout callin, callout, internet y la total

- SMS-in_activity: 5161
- SMS-out_activity: 5059
- Call-in_activity: 5161
- Call-out_activity: 5059
- Internet_traffic_activity: 5161
- Actividad total: 5161

```
#-------- SMS-in_activity --------
pipeline_SMS_in_activity = [
    {"$group": {"_id": "$Square_id", "total_SMS-in_activity": {"$sum":
"$SMS-in_activity"}}},
    {"$sort": {"total_SMS-in_activity": -1}},
    {"$limit": 1}
]

most_active_SMS_in_activity_cell =
coleccion.aggregate(pipeline_SMS_in_activity)
```

```
most_active_SMS_in_activity_cell =
list(most_active_SMS_in_activity_cell)[0]['_id']

#-------- SMS-out_activity --------
pipeline_SMS_out_activity = [
    {"$group": {"_id": "$Square_id", "total_SMS-out_activity": {"$sum":
"$SMS-out_activity"}}},
    {"$sort": {"total_SMS-out_activity": -1}},
    {"$limit": 1}
]

most_active_SMS_out_activity_cell =
coleccion.aggregate(pipeline_SMS_out_activity)
most_active_SMS_out_activity_cell =
list(most_active_SMS_out_activity_cell)[0]['_id']

#-------- Call-in_activity --------
pipeline_Call_in_activity = [
    {"$group": {"_id": "$Square_id", "total_Call-in_activity": {"$sum":
"$Call-in_activity"}}},
    {"$sort": {"total_Call-in_activity": -1}},
    {"$limit": 1}
]

most_active_Call_in_activity_cell =
coleccion.aggregate(pipeline_Call_in_activity)
most_active_Call_in_activity_cell =
list(most_active_Call_in_activity_cell)[0]['_id']

#-------- Call-out_activity --------
pipeline_Call_out_activity = [
    {"$group": {"_id": "$Square_id", "total_Call-out_activity": {"$sum":
"$Call-out_activity"}}},
    {"$sort": {"total_Call-out_activity": -1}},
    {"$limit": 1}
]

most_active_Call_out_activity_cell =
coleccion.aggregate(pipeline_Call_out_activity)
most_active_Call_out_activity_cell =
list(most_active_Call_out_activity_cell)[0]['_id']

#-------- Internet_traffic_activity --------
pipeline_Internet_traffic_activity = [
    {"$group": {"_id": "$Square_id", "total_Internet_traffic_activity":
{"$sum": "$Internet_traffic_activity"}}},
    {"$sort": {"total_Internet_traffic_activity": -1}},
    {"$limit": 1}
]

most_active_Internet_traffic_activity_cell =
coleccion.aggregate(pipeline_Internet_traffic_activity)
most_active_Internet_traffic_activity_cell =
list(most_active_Internet_traffic_activity_cell)[0]['_id']

#-------- Actividad total --------
pipeline_total_activity = [
    {"$group": {
        "_id": "$Square_id",
```

```
        "total_activity": {"$sum": {"$sum": ["$SMS-in_activity", "$SMS-
out_activity", "$Call-in_activity", "$Call-out_activity",
"$Internet_traffic_activity"]}}
    }},
    {"$sort": {"total_activity": -1}},
    {"$limit": 1}
]

most_active_total_cell = coleccion.aggregate(pipeline_total_activity)
most_active_total_cell = list(most_active_total_cell)[0]['_id']

#-------- Resultados --------
print("SMS-in_activity:", most_active_SMS_in_activity_cell)
print("SMS-out_activity:", most_active_SMS_out_activity_cell)
print("Call-in_activity:", most_active_Call_in_activity_cell)
print("Call-out_activity:", most_active_Call_out_activity_cell)
print("Internet_traffic_activity:",
most_active_Internet_traffic_activity_cell)
print("Actividad total:", most_active_total_cell)
```

Figura 5: Código Actividad Celdas

Ejercicio 6

Crea una colección con un documento por celda en el que aparezcan los acumulados de los diferentes campos.

Podemos observar la nueva colección en la terminal:

```
Milan_CDR_db> show collections
accumulated_by_cell
Milan_CDR_c
Milan_CDR_db> db.accumulated_by_cell.find().pretty()
[
  {
    _id: 3816,
    'total_SMS-in_activity': 1824.0608869473417,
    'total_SMS-out_activity': 742.3453886903881,
    'total_Call-in_activity': 805.2661876273016,
    'total_Call-out_activity': 1082.9257924870876,
    total_Internet_traffic_activity: 14434.898716545877
  },
  {
    _id: 4305,
    'total_SMS-in_activity': 559.1962736735596,
    'total_SMS-out_activity': 308.2285214121279,
    'total_Call-in_activity': 300.40291566408786,
    'total_Call-out_activity': 347.5709965244656,
    total_Internet_traffic_activity: 11411.401053721862
```

Figura 6: Nueva Colección por Celdas

```
cliente = MongoClient("mongodb://localhost:27017/")
db = cliente["Milan_CDR_db"]
colleccion = db["Milan_CDR_c"]

pipeline_accumulated_by_cell = [
    {"$group": {"_id": "$Square_id",
                "total_SMS-in_activity": {"$sum": "$SMS-in_activity"},
                "total_SMS-out_activity": {"$sum": "$SMS-out_activity"},
                "total_Call-in_activity": {"$sum": "$Call-in_activity"},
                "total_Call-out_activity": {"$sum": "$Call-out_activity"},
                "total_Internet_traffic_activity": {"$sum":
"$Internet_traffic_activity"}}},
    {"$out": "accumulated_by_cell"}
]

colleccion.aggregate(pipeline_accumulated_by_cell)
```

Figura 7: Código Nueva Colección por Celdas

Ejercicio 7

Crea una colección con un documento por celda y hora en el que aparezcan los acumulados de los diferentes campos.

Al igual que en el ejercicio anterior, mediante la terminal podemos observar los cambios realizados:

```
Milan_CDR_db> show collections
accumulated_by_cell
accumulated_cell_and_hour
Milan_CDR_c
Milan_CDR_db> db.accumulated_cell_and_hour.find().pretty()
[
  {
    _id: { Square_id: 3384, hour: null },
    'total_SMS-in_activity': 5380.894955459235,
    'total_SMS-out_activity': 3339.4941841084446,
    'total_Call-in_activity': 2655.9399200208395,
    'total_Call-out_activity': 2929.657943004438,
    total_Internet_traffic_activity: 57471.76562926072
  },
  {
Milan_CDR_db>
    'total_SMS-in_activity': 358.60016037201314,
    'total_SMS-out_activity': 201.45394153349628,
    'total_Call-in_activity': 183.55998802093293,
    'total_Call-out_activity': 202.0808090443235,
    total_Internet_traffic_activity: 7977.860256711431
  },
  {
    _id: { Square_id: 5945, hour: null },
    'total_SMS-in_activity': 9641.298507685156,
    'total_SMS-out_activity': 4998.680886873173,
    'total_Call-in_activity': 5668.8385107433505
```

Figura 8: Nueva Colección por Celdas y Hora

```
pipeline_ej7 = [
    {"$group": {"_id": {"Square_id": "$Square_id", "hour": {"$hour":
{"$toDate": "$time"}}},
                "total_SMS-in_activity": {"$sum": "$SMS-in_activity"},
                "total_SMS-out_activity": {"$sum": "$SMS-out_activity"},
                "total_Call-in_activity": {"$sum": "$Call-in_activity"},
                "total_Call-out_activity": {"$sum": "$Call-out_activity"},
                "total_Internet_traffic_activity": {"$sum":
"$Internet_traffic_activity"}}},
    {"$out": "accumulated_cell_and_hour"}
]

coleccion.aggregate(pipeline_ej7)
```

Figura 9: Código Nueva Colección por Celdas y Hora

Ejercicio 8

Realiza un estudio de las celdas 4259 (Bocconi), 4456 (Navigli), 5060 (Duomo), 1419 (terreno agricula), 2436 (área industrial), 4990 (aeropuerto de Linate), 945 (residencial aislado) y 5048 (residencial céntrico).

Para realizar este estudio he obtenido la media, mediana y desviación típica de actividad de smsin, smsout callin, callout, internet y la total para cada celda.

Las celdas de Duomo (5060) y Residencial Céntrico (5048) presentan la mayor actividad en llamadas, SMS y tráfico de Internet, con altos valores medios y gran variabilidad, lo que nos indica una intensa actividad comercial y turística. Navigli (4456) también destaca en uso de datos, lo que sugiere un elevado número de usuarios conectados.

Por otra parte, las celdas de terreno agrícola (1419) y residencial aislado (945) muestran una mínima actividad en telecomunicaciones, confirmando su baja densidad poblacional.

El Aeropuerto de Linate (4990) experimenta altas fluctuaciones en llamadas y datos, reflejando los flujos de pasajeros. Por su parte, Bocconi (4259) presenta un consumo de Internet elevado con picos de uso, probablemente debido a la actividad universitaria.

En general, las zonas comerciales y turísticas concentran la mayor actividad, mientras que las áreas rurales y residenciales aisladas muestran un uso muy reducido.

```
client = MongoClient("mongodb://localhost:27017/")
db = client["Milan_CDR_db"]
coleccion = db["Milan_CDR_c"]

# Lista de celdas a analizar
celdas_estudio = [4259, 4456, 5060, 1419, 2436, 4990, 945, 5048]

pipeline_estadisticas = [
    {"$match": {"Square_id": {"$in": celdas_estudio}}},
    {"$group": {
        "_id": "$Square_id",
        "count": {"$sum": 1},
        "mean_SMS_in": {"$avg": "$SMS-in_activity"},
        "mean_SMS_out": {"$avg": "$SMS-out_activity"},
        "mean_call_in": {"$avg": "$Call-in_activity"},
        "mean_call_out": {"$avg": "$Call-out_activity"},
        "mean_internet": {"$avg": "$Internet_traffic_activity"},
        "std_SMS_in": {"$stdDevPop": "$SMS-in_activity"},
```

```
        "std_SMS_out": {"$stdDevPop": "$SMS-out_activity"},
        "std_call_in": {"$stdDevPop": "$Call-in_activity"},
        "std_call_out": {"$stdDevPop": "$Call-out_activity"},
        "std_internet": {"$stdDevPop": "$Internet_traffic_activity"},
        "all_SMS_in": {"$push": "$SMS-in_activity"},
        "all_SMS_out": {"$push": "$SMS-out_activity"},
        "all_call_in": {"$push": "$Call-in_activity"},
        "all_call_out": {"$push": "$Call-out_activity"},
        "all_internet": {"$push": "$Internet_traffic_activity"}
    }},
    {"$project": {
        "_id": 1,
        "count": 1,
        "mean_SMS_in": 1, "mean_SMS_out": 1, "mean_call_in": 1,
"mean_call_out": 1, "mean_internet": 1,
        "std_SMS_in": 1, "std_SMS_out": 1, "std_call_in": 1,
"std_call_out": 1, "std_internet": 1,
        "median_SMS_in": {"$arrayElemAt": ["$all_SMS_in", {"$floor":
{"$divide": ["$count", 2]}}]},
        "median_SMS_out": {"$arrayElemAt": ["$all_SMS_out", {"$floor":
{"$divide": ["$count", 2]}}]},
        "median_call_in": {"$arrayElemAt": ["$all_call_in", {"$floor":
{"$divide": ["$count", 2]}}]},
        "median_call_out": {"$arrayElemAt": ["$all_call_out", {"$floor":
{"$divide": ["$count", 2]}}]},
        "median_internet": {"$arrayElemAt": ["$all_internet", {"$floor":
{"$divide": ["$count", 2]}}]}
    }},
    {"$sort": {"_id": 1}}
]

resultados_estadisticas = list(coleccion.aggregate(pipeline_estadisticas))

print("\nEstadísticas de cada celda (Media, Mediana y Desviación Típica):")
print("=" * 50)
for r in resultados_estadisticas:
    print(f"Celda {r['_id']}:")
    print(f"  - SMS entrantes -> Media: {r['mean_SMS_in']:.2f}, Mediana:
{r['median_SMS_in']}, Desviación: {r['std_SMS_in']:.2f}")
    print(f"  - SMS salientes -> Media: {r['mean_SMS_out']:.2f}, Mediana:
{r['median_SMS_out']}, Desviación: {r['std_SMS_out']:.2f}")
    print(f"  - Llamadas entrantes -> Media: {r['mean_call_in']:.2f},
Mediana: {r['median_call_in']}, Desviación: {r['std_call_in']:.2f}")
    print(f"  - Llamadas salientes -> Media: {r['mean_call_out']:.2f},
Mediana: {r['median_call_out']}, Desviación: {r['std_call_out']:.2f}")
    print(f"  - Tráfico de Internet -> Media: {r['mean_internet']:.2f},
Mediana: {r['median_internet']}, Desviación: {r['std_internet']:.2f}")
    print("=" * 50)
```

Figura 10: Código Análisis

Resultados obtenidos en el análisis:

```
Estadísticas de cada celda (Media, Mediana y Desviación Típica):
==================================================
Celda 945:
  - SMS entrantes -> Media: 0.75, Mediana: null, Desviación: 0.85
  - SMS salientes -> Media: 0.75, Mediana: null, Desviación: 1.01
  - Llamadas entrantes -> Media: 0.75, Mediana: null, Desviación: 0.86
```

```
  - Llamadas salientes -> Media: 0.50, Mediana: 0.02283840435291939,
Desviación: 0.84
  - Tráfico de Internet -> Media: 13.98, Mediana: null, Desviación: 11.92
================================================================================
Celda 1419:
  - SMS entrantes -> Media: 0.22, Mediana: 0.03988471000100698, Desviación:
0.25
  - SMS salientes -> Media: 0.20, Mediana: 0.010099125743485763,
Desviación: 0.24
  - Llamadas entrantes -> Media: 0.25, Mediana: null, Desviación: 0.29
  - Llamadas salientes -> Media: 0.18, Mediana: 0.003965030995841482,
Desviación: 0.27
  - Tráfico de Internet -> Media: 2.61, Mediana: 2.2852856519027576,
Desviación: 2.47
================================================================================
Celda 2436:
  - SMS entrantes -> Media: 1.22, Mediana: null, Desviación: 1.35
  - SMS salientes -> Media: 1.61, Mediana: null, Desviación: 1.57
  - Llamadas entrantes -> Media: 1.29, Mediana: null, Desviación: 1.40
  - Llamadas salientes -> Media: 1.66, Mediana: null, Desviación: 1.93
  - Tráfico de Internet -> Media: 19.09, Mediana: 0.001715871504108279,
Desviación: 17.23
================================================================================
Celda 4259:
  - SMS entrantes -> Media: 2.66, Mediana: 5.702207890101282, Desviación:
3.31
  - SMS salientes -> Media: 2.48, Mediana: 3.21519697187897, Desviación:
2.72
  - Llamadas entrantes -> Media: 3.09, Mediana: 0.25509897519381863,
Desviación: 3.98
  - Llamadas salientes -> Media: 2.29, Mediana: 1.8437803920703508,
Desviación: 3.87
  - Tráfico de Internet -> Media: 47.31, Mediana: 55.70903431020007,
Desviación: 52.69
================================================================================
Celda 4456:
  - SMS entrantes -> Media: 6.73, Mediana: null, Desviación: 9.59
  - SMS salientes -> Media: 5.68, Mediana: null, Desviación: 7.72
  - Llamadas entrantes -> Media: 7.88, Mediana: null, Desviación: 11.51
  - Llamadas salientes -> Media: 5.33, Mediana: 0.24990804533372257,
Desviación: 9.88
  - Tráfico de Internet -> Media: 130.38, Mediana: null, Desviación: 173.79
================================================================================
Celda 4990:
  - SMS entrantes -> Media: 2.86, Mediana: 4.660398079838499, Desviación:
4.10
  - SMS salientes -> Media: 2.02, Mediana: null, Desviación: 3.15
  - Llamadas entrantes -> Media: 2.65, Mediana: null, Desviación: 4.06
  - Llamadas salientes -> Media: 3.08, Mediana: null, Desviación: 4.79
  - Tráfico de Internet -> Media: 26.99, Mediana: null, Desviación: 48.53
================================================================================
Celda 5048:
  - SMS entrantes -> Media: 8.15, Mediana: null, Desviación: 11.57
  - SMS salientes -> Media: 9.94, Mediana: null, Desviación: 11.82
  - Llamadas entrantes -> Media: 8.65, Mediana: null, Desviación: 12.05
  - Llamadas salientes -> Media: 6.11, Mediana: null, Desviación: 10.71
  - Tráfico de Internet -> Media: 127.70, Mediana: 0.06494965410011584,
Desviación: 143.27
================================================================================
Celda 5060:
```

```
  - SMS entrantes -> Media: 11.28, Mediana: 0.6709077240934839, Desviación:
21.93
  - SMS salientes -> Media: 8.41, Mediana: null, Desviación: 15.19
  - Llamadas entrantes -> Media: 10.47, Mediana: null, Desviación: 23.93
  - Llamadas salientes -> Media: 11.35, Mediana: null, Desviación: 27.44
  - Tráfico de Internet -> Media: 84.97, Mediana: 0.2697801990581178,
Desviación: 227.43
================================================================================
```

Figura 11: Resultados Análisis Ejercicio 8