Escuela Politécnica Nacional

Análisis de Datos

2024-B

Nombre: Luis Adrián Ramos Guzmán

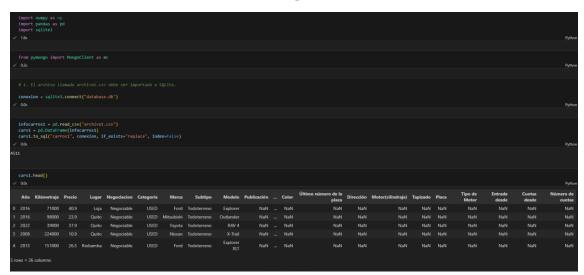
Curso: GR1

DESARROLLO

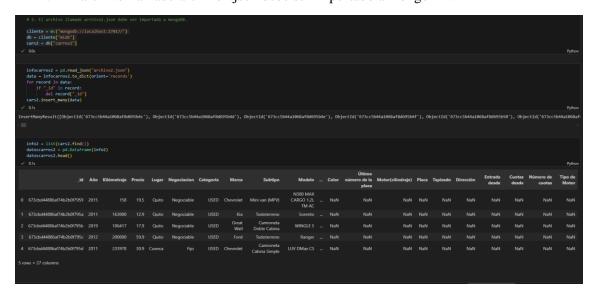
Carga de datos. (10 puntos)

Implementar el proceso de carga de datos, cada item corresponde al número de la ilustración.

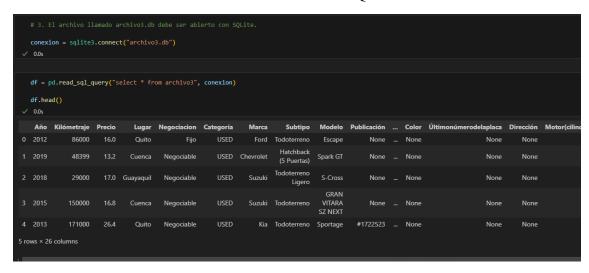
1. El archivo llamado archivo1.csv debe ser importado a SQLite.



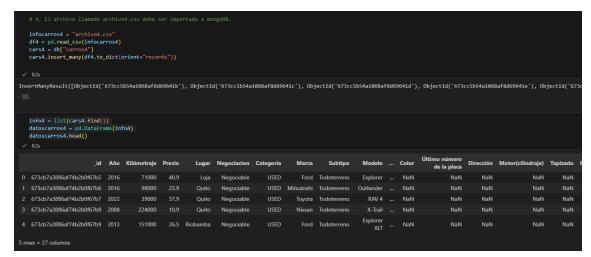
2. El archivo llamado archivo2.json debe ser importado a mongoDB.



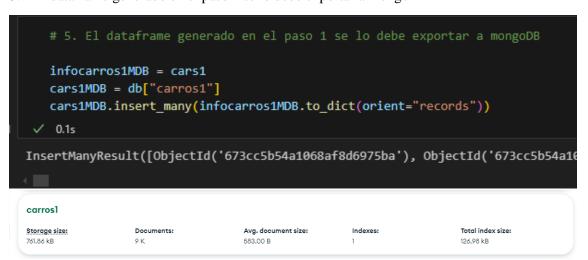
3. El archivo llamado archivo 3.db debe ser abierto con SQLite.



4. El archivo llamado archivo4.csv debe ser importado a mongoDB.



5. El dataframe generado en el paso 1 se lo debe exportar a mongoDB



6. El dataframe generado en el paso 3 se lo debe exportar a mongoDB

```
# 6. El dataframe generado en el paso 3 se lo debe exportar a mongoDB

infocarros2MDB = df
cars2MDB = db["carros3"]
cars2MDB.insert_many(infocarros2MDB.to_dict(orient= 'records'))

✓ 0.0s

InsertManyResult([ObjectId('673cc5b64a1068af8d698759'), ObjectId('673cc5b64a1068af8d6

carros3

Storage.size:
364.54 kB

Documents:
Avg.document size:
1ndexes:
Total index size:
81.92 kB
```

7. El dataframe generado en el paso 2 se lo debe exportar a SQLite

```
# 7. El dataframe generado en el paso 2 se lo debe exportar a SQLite
    data = list(cars2.find())
    for record in data:
        record['_id'] = str(record['_id'])
    cars2SQL = pd.DataFrame(data)
    cars2SQL.to_sql("carros2", conexion, if_exists="replace", index=False)
  ✓ 0.1s
 4510
Name
                                                            Schema
                                             Type
Tables (4)
   > III archivo3
                                                            CREATE TABLE "archivo3" ( "A
   > III carros1
                                                            CREATE TABLE "carros1" ( "Af
                                                            CREATE TABLE "carros2" ( "_ic
     arros2
                                                            CREATE TABLE "carros4" ( "_ic
   > III carros4
```

8. El dataframe generado en el paso 4 se lo debe exportar a SQLite

```
# 8. El dataframe generado en el paso 4 se lo debe exportar a SQLite
    data2 = list(cars4.find())
    cars4SQL = pd.DataFrame(data)
    cars4SQL.to_sql('carros4', conexion, if_exists="replace", index=False)

√ 0.2s

 4510
Name
                                           Type
                                                          Schema
Tables (4)
                                                         CREATE TABLE "archivo3" ( "A
  archivo3
                                                         CREATE TABLE "carros1" ( "Af
   > III carros1
                                                         CREATE TABLE "carros2" ( "_ic
     arros2
   > III carros4
                                                         CREATE TABLE "carros4" ( "_ic
```

9. Se debe exportar los datos a un dataframe.

```
# 9. Se debe exportar los datos a un dataframe.
   data = pd.concat([cars2SQL, cars4SQL], ignore_index=True)
   carsMDB = db["carros"]
   data_dict = data.to_dict(orient="records")
    for record in data_dict:
        if " id" in record:
             del record["_id"]
    carsMDB.insert many(data dict)

√ 0.2s

InsertManyResult([ObjectId('673ccfd14a1068af8d699028'), ObjectId('673ccfd1
carros
                 Documents:
                                 Avg. document size:
                                                  Indexes:
                                                                   Total index size:
397.31 kB
                14 K
                                 589.00 B
                                                                   69,63 kB
```

10. Se debe exportar los datos a un dataframe.

```
data1 = list(cars1MDB.find())
   data2 = list(cars2MDB.find())
   for record in data1: record["_id"] = str(record["_id"])
   for record in data2: record["_id"] = str(record["_id"])
   df1 = pd.DataFrame(data1)
   df2 = pd.DataFrame(data2)
   carsSQL = pd.concat([df1, df2], ignore_index=True)
   conexion = sqlite3.connect('database.db', timeout=30)
   with conexion:
       carsSQL.to_sql("carros", conexion, if_exists="replace", index=False)
 ✓ 0.4s
Tables (3)
                                                            CREATE TABLE "carros" ( "_ic
     arros
   > III carros1
                                                            CREATE TABLE "carros1" ( "in
                                                            CREATE TABLE "carros2" ( "in
     arros2
```

Limpieza de datos. (4 puntos)

11. Debe unificar y limpiar los datos de tal manera que estén listos para analizarlos

```
df_Carros1 = carsSQL.drop_duplicates()
   df_Carros1 = carsSQL.dropna()
   df_Carros1.isnull().sum()
Kilómetraje
Precio
Lugar
Negociacion
Categoría
Marca
Subtipo
Modelo
Publicación
Transmisión
Combustible
Sistema de climatización
Último número de la placa 0
Dirección
Motor(cilindraje)
Tapizado
Placa
Tipo de Motor
Entrada desde
TipodeMotor
Entradadesde
Cuotasdesde
dtype: int64
Output is truncated. View as a <u>scrollable element</u> or open in a <u>text editor</u>. Adjust cell output <u>settings</u>...
```

```
data = list(carsMDB.find())
   carrosMDB = pd.DataFrame(data)
   df_Carros2 = carrosMDB.drop_duplicates()
df_Carros2 = carrosMDB.dropna()
   df_Carros2.isnull().sum()
_id
Año
Kilómetraje
Lugar
Negociacion
Categoría
Marca
Subtipo
Modelo
Publicación
Recorrido
Combustible
Sistema de climatización
Tracción
Color
Último número de la placa
Motor(cilindraje)
Placa
Tapizado
Dirección
Cuotas desde
Número de cuotas
Tipo de Motor
dtype: int64
```

Análisis de datos. (10 puntos)
Con pandas debe contestar a las siguientes preguntas:
a. Los autos que tienen los años más recientes y los autos que tienen los autos más
antiguos.
b. El promedio de kilometraje de los autos de Quito o Guayaquil.
c. El promedio de kilometraje de los autos de Cuenca.
d. Los autos que son negociables y fijos en porcentaje, campo negociación.
e. Porcentaje de autos nuevos y usados.
f. Las marcas que más se repiten.
g. Modelos que más se repiten.
h. Porcentaje de subtipos.
i. Auto más caro y más barato, con sus características.
j. Auto de Guayaquil que menos kilometraje tiene y más caro es.

Parte teórica (6 puntos)

Contestar a las siguientes preguntas:

a) Defina cada etapa de la pirámide del conocimiento.

La pirámide del conocimiento es un esquema que representa un paso a paso de como procesar y analizar la información, empezando por los datos que es la parte mas básica de la pirámide, pasando por la información que nos proporcionan esos datos. Llegando al conocimiento una vez analizado los datos y por ultimo la sabiduría una vez obtenido el conocimiento.

b) Indique lo que es un archivo .json

Un archivo json es un formato de transmisión de información basado en javascript. Su sintaxis es parecida a un objeto en el mismo lenguaje. Es altamente usado en el desarrollo web y bases de datos no sql.

c) Explique las diferencias entre Bases de Datos relacionales y NoSQL.
 Las bases de datos sql o relacionales se escriben en xon la sintaxis de SQL y están relacionadas por tablas, mientras que las no sql se basan en json para guardar sus datos

Entregables:

Archivo ipynb y el pdf con capturas de pantalla indicando el procedimiento.

En el pdf puede poner las respuestas de la parte teórica.

Importante:

Tomar en consideración:

- El uso de dataframes en cada etapa.
- MongoDB y SQLite tienen las opciones de importación y exportación.