Universidad Politécnica de Tapachula

"Innovación y Tecnología al servicio de la sociedad"

Nombre de los alumnos – Matriculas

Zarate Velázquez Mónica Lizbeth – 203097
Juan Carlos Monzón Hernández -222003
Roberto Escobar Córdova - 193021

Materia.

Minería de datos.

Cuatrimestre / periodo escolar

9º. U (MAYO – AGOSTO/ 2023)

Unidad - Tema

Unidad1

Nombre de la práctica / proyecto.

Actividad en kaggle

Tipo de evidencia

Evidencia

Plan de estudios.

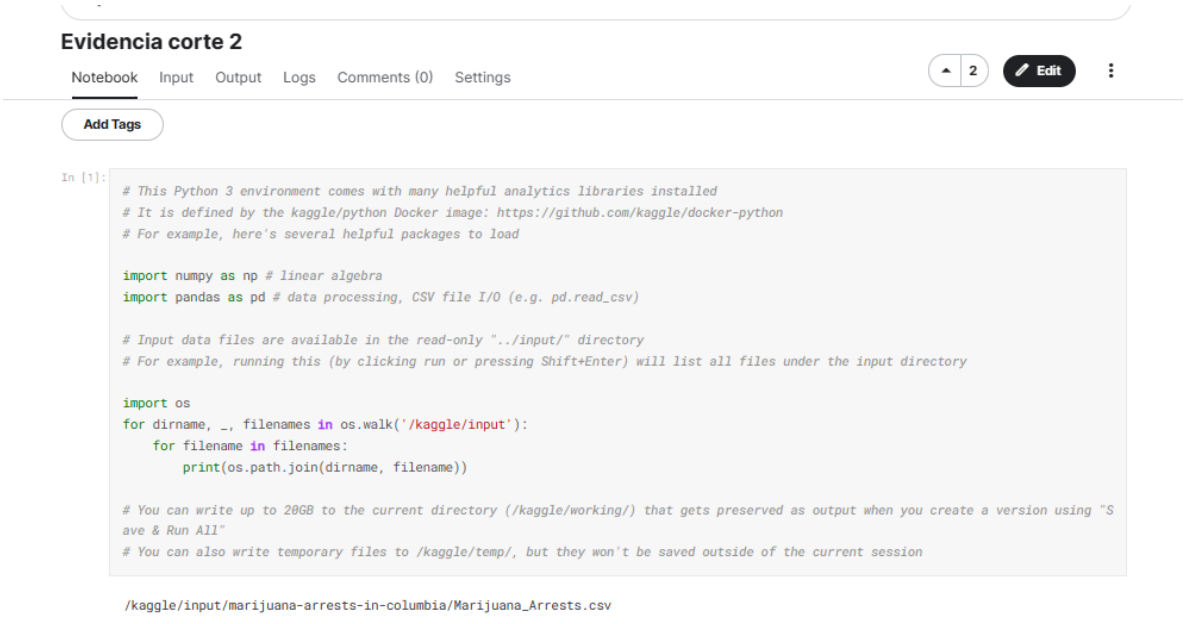
Ingeniería de Software.

Catedrático.

Karina Cancino Villatoro

Tapachula, Chiapas; 23 de junio de 2023

La base de datos que se utilizó para esta actividad es: Marijuana Arrest in Columbia, en esta activa se tuvieron que desarrollar los siguientes puntos para obtener ciertas cifras, números o tablas, etc.



Evidencia corte 2

Notebook Input Output Logs Comments (0) Settings 2 Edit

Add Tags

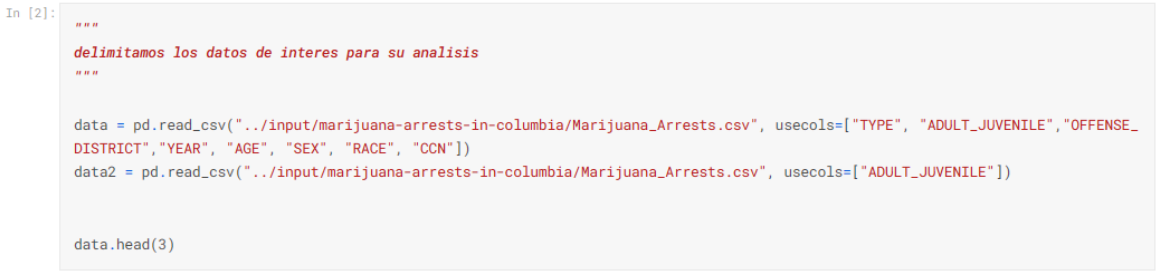
In [1]:

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/marijuana-arrests-in-columbia/Marijuana_Arrests.csv
```

El primer punto que se desarrollo es: Crear un nuevo conjunto de datos a partir de la BD Marijuana_Arrests.csv

En nuestro caso, decidimos delimitar de la siguiente manera la BD, debido a que hay ciertos requerimientos que son más de interés que otros, ya que debemos tener en mente que solo nos servirán aquellos datos con cifras, que nos den la idea principal de cuantos arrestos se obtienen.



In [2]:

```
"""
delimitamos los datos de interes para su analisis
"""

data = pd.read_csv("../input/marijuana-arrests-in-columbia/Marijuana_Arrests.csv", usecols=["TYPE", "ADULT_JUVENILE","OFFENSE_DISTRICT","YEAR", "AGE", "SEX", "RACE", "CCN"])
data2 = pd.read_csv("../input/marijuana-arrests-in-columbia/Marijuana_Arrests.csv", usecols=["ADULT_JUVENILE"])


data.head(3)
```

Out[2]:

	TYPE	ADULT_JUVENILE	YEAR	CCN	AGE	OFFENSE_DISTRICT	RACE	SEX
0	Possession	Adult	2012	b';\xc8k~\xa4iJ'	20.0	5D	B	F
1	Possession	Adult	2012	b';\xc8k~\xa4iJ'	23.0	5D	B	M
2	Possession	Adult	2012	b't6\xa0\xac\xec`\xa4'	46.0	7D	B	M

Para el segundo punto, debemos determinar y mostrar si existen registros sin datos y eliminarlos en caso que existan. En caso de que existan deberá proyectar cuántos registros son y cuáles son.

Por lo tanto, se determinó los conjuntos sin datos que lleva cada apartado de la tabla, sin olvidar cuales son las columnas necesarias que deberían, pero en este caso no se obtiene nada de importancia.

Aunque, se encontró un dato interesante que es que los juveniles son quienes no tienen datos, sin embargo no podemos eliminarlos, debido a que más delante nos servirán.

In [3]:

```
"""
1. Determinar y mostrar si existen registros sin datos y eliminarlos en caso que
existan. En caso de que existan deberá proyectar cuántos registros son y cuáles
son.


"""
dataNull = data["RACE"].isna()
data[dataNull]
```

Out[3]:

	TYPE	ADULT_JUVENILE	YEAR	CCN	AGE	OFFENSE_DISTRICT	RACE	SEX
42	Possession with intent to distribute	Juvenile	2012	NaN	NaN	7D	NaN	NaN
56	Possession	Juvenile	2012	NaN	NaN	3D	NaN	NaN
75	Distribution	Juvenile	2012	NaN	NaN	1D	NaN	NaN
76	Distribution	Juvenile	2012	NaN	NaN	1D	NaN	NaN
83	Possession	Juvenile	2012	NaN	NaN	7D	NaN	NaN
...	...	...	...	...	...	...	...	...
13058	Possession with intent to distribute	Juvenile	2021	NaN	NaN	4D	NaN	NaN
13059	Possession with intent to distribute	Juvenile	2021	NaN	NaN	7D	NaN	NaN
13060	Public consumption	Juvenile	2021	NaN	NaN	4D	NaN	NaN
13061	Possession	Juvenile	2021	NaN	NaN	6D	NaN	NaN
13062	Possession with intent to distribute	Juvenile	2021	NaN	NaN	4D	NaN	NaN

553 rows × 8 columns

In [4]:
```
dataNull.sum()
```

Out[4]:
```
553
```

In [5]:
```
"""
Eliminacion de los datos NaN

la funcion .dropna() elimina los registros NaN
en este caso, los arrestos que son juveniles no tienen registro de CCN, Edad, Raza, Sexo
"""
data = data.dropna(subset=["ADULT_JUVENILE"])
data.head()
```

Out[5]:

	TYPE	ADULT_JUVENILE	YEAR	CCN	AGE	OFFENSE_DISTRICT	RACE	SEX
0	Possession	Adult	2012	b';\xc8k~\xa4iJ'	20.0	5D	B	F
1	Possession	Adult	2012	b';\xc8k~\xa4iJ'	23.0	5D	B	M
2	Possession	Adult	2012	b't6\xa0\xac\xec`\xa4'	46.0	7D	B	M
3	Possession	Adult	2012	b'\xbe\x1d\xa7\xf5\xffWx'	30.0	6D	B	M
4	Possession with intent to distribute	Adult	2012	b'\xbb\xc0\x8e\x94\x81\xac\xcd'	29.0	6D	U	M

Después, tenemos el tercer paso que nos indica lo siguiente:

Determinar y mostrar existen registros duplicados y eliminarlos en caso que exista. En caso de que existan deberá proyectar cuántos registros son y cuáles son.

Aquí, podemos observar lo que se relató anteriormente, los juveniles no tienen datos que nos puedan mostrar, esto debido a que por ejemplo, por ser menores de edad no cuentan con una identificación oficial.

In [6]:
```
"""
3. Determinar y mostrar existen registros duplicados y eliminarlos en caso que
existan. En caso de que existan deberá proyectar cuántos registros son y cuáles
son.

"""
# sacamos datos duplicados
duplicados = data[data.duplicated(subset=["TYPE","ADULT_JUVENILE", "OFFENSE_DISTRICT", "YEAR","CCN", "AGE", "SEX", "RACE"])]
# guardamos los datos NO duplicados
data = data[~data.duplicated(subset=["TYPE","ADULT_JUVENILE", "OFFENSE_DISTRICT", "YEAR","CCN", "AGE", "SEX", "RACE"])]



duplicados.head()
```

Out[6]:

	TYPE	ADULT_JUVENILE	YEAR	CCN	AGE	OFFENSE_DISTRICT	RACE	SEX
24	Possession	Adult	2012	b'\x91re\xe5N\x1fT'	44.0	1D	B	F
76	Distribution	Juvenile	2012	NaN	NaN	1D	NaN	NaN
87	Possession	Juvenile	2012	NaN	NaN	7D	NaN	NaN
107	Distribution	Juvenile	2012	NaN	NaN	1D	NaN	NaN
135	Possession	Adult	2012	b'l\x99\xe2B#*G'	18.0	3D	B	M

In [7]:
```
data.count()
```

Out[7]:
```
TYPE                12451
ADULT_JUVENILE      12451
YEAR                12451
CCN                 12294
AGE                 12250
OFFENSE_DISTRICT    12451
RACE                12294
SEX                 12294
dtype: int64
```

Luego de ello, en el paso 4 se agrupan conjuntos por edades:

In [8]:
```
"""
4. Agrupar el conjunto de datos por edades.
"""

dataAgrup = data.groupby("AGE")
dataAgrup.size()
```

Out[8]:
```
AGE
18.0    552
19.0    652
20.0    713
21.0    789
22.0    775
       ...
74.0      1
75.0      1
76.0      1
77.0      1
81.0      1
Length: 61, dtype: int64
```

Después de ello, se ordena el conjunto de datos por años y tipo de arresto de manera descendente. Teniendo en cuenta que mostramos los primeros 5 registros y los últimos 5,

debido a que decidimos que se esa forma podemos verificar que los registros estas de manera descendente:

```
5. Ordenar el conjunto de datos por años y tipo de arresto de manera descendente

mostramos los primeros 5 registros y los ultimos 5 :3

"""

dataOrdenada = data.sort_values(by=["YEAR","TYPE"], ascending=False)

dataOrdenada[5:-5]
```

Out[9]:

	TYPE	ADULT_JUVENILE	YEAR	CCN	AGE	OFFENSE_DISTRICT	RACE	SEX
12790	Public consumption	Adult	2021	65fee0efdbac2252	24.0	1D	B	M
12797	Public consumption	Adult	2021	01d3f77d6ee699ea	47.0	1D	B	M
12799	Public consumption	Adult	2021	c5c31dd8eb908c44	26.0	1D	B	M
12803	Public consumption	Adult	2021	499d8ed6fb8013ed	32.0	1D	B	M
12804	Public consumption	Adult	2021	aad986e4892c3193	33.0	1D	B	M
...	...	...	...	...	...	...	...	...
3351	Distribution	Adult	2012	b'\t\xae\x86k\xf8\xe6.'	23.0	4D	B	M
3419	Distribution	Adult	2012	b'\x9bV\xee\xb8J2\xf9'	21.0	6D	B	M
3420	Distribution	Adult	2012	b'\x9bV\xee\xb8J2\xf9'	26.0	6D	B	M
3426	Distribution	Adult	2012	b'\xc71\x8a\xc0=\x80\xec'	18.0	6D	B	M
3429	Distribution	Adult	2012	b'\xe2\xcf\xb3\x13\xe1Fu'	42.0	6D	B	M

12441 rows × 8 columns

El punto 6: nos dice que debemos mostrar el número de arrestos generados en cada año clasificados por tipo de posesión.
Por lo tanto, aquí se nos muestra este tipo de agrupación, sin embargo no únicamente con ‘possession’, si no tambien con ‘possession with intent to distribute’, esto debido a que la petición no nos encaja únicamente que sean por posesión, si no que nos da la idea que la clasificación tenga el tipo posesión pero puede estar unido a cualquier otro tipo como la distribución.

```
6. Mostrar el numero de arrestos generados en cada año clasificados por tipo de
posesión.
"""

arrestosAnioPosesion = data.groupby(['YEAR', 'TYPE']).size()

print(arrestosAnioPosesion)
```

```
YEAR  TYPE
2012  Distribution                            131
      Possession                             2452
      Possession with intent to distribute    749
2013  Distribution                             76
      Possession                             1879
      Possession with intent to distribute    648
2014  Distribution                            107
      Possession                             1377
      Possession with intent to distribute    497
      Public consumption                       95
2015  Distribution                             74
      Possession                               23
      Possession with intent to distribute    135
      Public consumption                       82
2016  Distribution                            207
      Possession                               14
      Possession with intent to distribute    160
      Public consumption                      259
```

2017	Distribution	378
	Possession	18
	Possession with intent to distribute	231
	Public consumption	252
2018	Distribution	279
	Possession	24
	Possession with intent to distribute	478
	Public Consumption	218
2019	Cultivation	1
	Distribution	337
	Manufacture	1
	Possession	23
	Possession with intent to distribute	351
	Public consumption	97
2020	Distribution	68
	Manufacture	1
	Possession	14
	Possession with intent to distribute	298
	Public consumption	73
2021	Cultivation	1
	Distribution	41
	Possession	15
	Possession with intent to distribute	252
	Public consumption	35

dtype: int64

Del resultado anterior mostrar cuantos son adultos y cuantos son juveniles. Aquí, se hace únicamente un conteo de la posesión de drogas:

In [11]:
```
data2 = data2[data2["ADULT_JUVENILE"] != "Unknown"]


arrestosAnioPosesion2 = data2["ADULT_JUVENILE"].value_counts()
print(arrestosAnioPosesion2)
```

```
Adult       12471
Juvenile      553
Name: ADULT_JUVENILE, dtype: int64
```

El punto 8, nos dice que debemos clasificar los arrestos por tipo de raza ordenándolos de manera descendente.
En este caso, únicamente nos muestra a los adultos, por lo que ya habíamos dicho anteriormente y solo mostramos los primeros 5 registros debido a que en la columna años nos muestra cómo están ordenados.

In [12]:
```
"""8. Clasificar los arrestos por tipo de raza ordenándolos de manera descendente

los datos solo son mostrados si son adultos ya que si son juveniles no se muestran ya que no
tienen sus registro

"""

dataOrdenada2 = data.sort_values(by=["RACE"], ascending=False)
dataOrdenada2.head()
```

Out[12]:

	TYPE	ADULT_JUVENILE	YEAR	CCN	AGE	OFFENSE_DISTRICT	RACE	SEX
3248	Possession	Adult	2012	b'\xe9\x94\n\xd2wi\xbd'	30.0	3D	W	M
2751	Possession	Adult	2012	b'\xe7\x8cQ\xa4x\x87@'	32.0	6D	W	F
5034	Possession	Adult	2013	b'p\xc0\x141\rC]'	25.0	3D	W	F
7367	Possession	Adult	2014	b'\xc7\x7f\xfdb\x08is'	61.0	7D	W	F
8157	Possession	Adult	2014	b"a(\xb1\xe2\xa5{"	20.0	3D	W	M

Continuando, se tiene que mostrar los primeros 5 años en donde más y menor arrestos se han obtenido de acuerdo a las tablas:

In [13]:
```
"""
9 Proyectar el top 5 de los años con mayor arrestos
"""
top5_mayor_arrestos = data["YEAR"].value_counts().head(5)

print(top5_mayor_arrestos)
```

```
2012    3332
2013    2603
2014    2076
2018     999
2017     879
Name: YEAR, dtype: int64
```

In [14]:
```
"""9 Proyectar el top 5 de los años con menor arrestos"""

top5_menor_arrestos = data["YEAR"].value_counts().tail(5)

print(top5_menor_arrestos)
```

```
2019    810
2016    640
2020    454
2021    344
2015    314
Name: YEAR, dtype: int64
```

Finalmente, debemos proyectar el top 5 de las edades y géneros en donde se presenta el mayor y menor número de arrestos

In [15]:

```
"""10. Proyectar el top 5 de las edades y géneros en donde se presenta el mayor y
menor número de arrestos

NOTA: al ser juveniles no tienen registro de CCN, Edad, Raza, Sexo por lo tanto el analisis se
aplicaran en adultos
"""


"""Top 5 edades y géneros con mayor número de arrestos:"""
top5_mayor_arrestos = data.groupby(["AGE", "SEX"]).size().nlargest(5)

print(top5_mayor_arrestos)
```

```
AGE   SEX
21.0  M      707
23.0  M      695
22.0  M      694
20.0  M      639
24.0  M      624
dtype: int64
```

In [16]:

```
"""10. Proyectar el top 5 de las edades y géneros en donde se presenta el mayor y
menor número de arrestos

NOTA: al ser juveniles no tienen registro de CCN, Edad, Raza, Sexo por lo tanto el analisis se
aplicaran en adultos
"""


"""Top 5 edades y géneros con menor número de arrestos:"""
top5_menor_arrestos = data.groupby(["AGE", "SEX"]).size().nsmallest(5)

print(top5_menor_arrestos)
```

```
AGE   SEX
19.0  U      1
21.0  U      1
23.0  U      1
25.0  U      1
31.0  U      1
dtype: int64
```