

**ACTIVIDAD APRENDIZAJE 3**

```

1 import pandas as pd
2 import numpy as np
3 import csv
4
5
6
7
8
9
10 def leer_datos():
11     df = pd.read_csv("Autos.csv")
12     print(df)
13     df
14     # Crear una matriz en numpy
15     matriz = np.array(df.values)
16     #print(matriz)
17     return matriz
18
19
20 def main():
21     m = leer_datos()
22     print(m)
23
24 main()
25

```

	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	2017	12000	1	15944	5	150	57.7	1.0
1	2018	14000	2	9083	5	150	57.7	1.0
2	2017	13000	2	12456	5	150	57.7	1.0
3	2019	17500	2	10460	5	145	40.3	1.5
4	2019	16500	1	1482	5	145	48.7	1.0
...	...	...	...	...	...	...	...	...
17957	2017	8999	2	16700	5	150	47.1	1.4
17958	2014	7499	2	40700	5	30	57.7	1.0
17959	2015	9999	2	7010	1	20	67.3	1.6
17960	2018	8299	2	5007	5	145	57.7	1.2
17961	2015	8299	2	5007	5	22	57.7	1.0

```

[17962 rows x 8 columns]
[[2.017e+03 1.200e+04 1.000e+00 ... 1.500e+02 5.770e+01 1.000e+00]
 [2.018e+03 1.400e+04 2.000e+00 ... 1.500e+02 5.770e+01 1.000e+00]
 [2.017e+03 1.300e+04 2.000e+00 ... 1.500e+02 5.770e+01 1.000e+00]
 ...
 [2.015e+03 9.999e+03 2.000e+00 ... 2.000e+01 6.730e+01 1.600e+00]
 [2.018e+03 8.299e+03 2.000e+00 ... 1.450e+02 5.770e+01 1.200e+00]
 [2.015e+03 8.299e+03 2.000e+00 ... 2.200e+01 5.770e+01 1.000e+00]]

```

```

1 def precio_maximo(m):
2     arreglo = (m[:, 1]) # Extrae la columna 5 de impuestos (tax)
3     print(arreglo)
4     res = np.max(arreglo)
5     return res
6
7 def main():
8     m = leer_datos()
9     # print(m)
10    print("\nPRECIO MÁXIMO DE UN AUTO")
11    res = precio_maximo(m)
12    print("El precio máximo es:", res)
13
14 main()

```

```

PRECIO MÁXIMO DE UN AUTO
[12000. 14000. 13000. ... 9999. 8299. 8299.]
El precio máximo es: 54995.0

```

```

1 def promedio_impuestos(m):
2     arreglo = (m[:, 5]) # Extrae la columna 5 de impuestos (tax)
3     print(arreglo)
4     res = np.mean(arreglo)

```

```

5     return res
6
7 def main():
8     m = leer_datos()
9     print("\nPROMEDIO DE IMPUESTOS")
10    res = promedio_impuestos(m)
11    print("El promedio de impuestos es %.2f" % res)
12
13 main()

```



```

PROMEDIO DE IMPUESTOS
[150. 150. 150. ... 20. 145. 22.]
El promedio de impuestos es 113.31

```

```

1 def suma_millage(m):
2     arreglo = (m[:, 3]) # Extrae la columna 5 de impuestos (tax)
3     print(arreglo)
4     res = np.sum(arreglo)
5     return res
6
7 def main():
8     m = leer_datos()
9     # print(m)
10    print("\nSUMA DE MILLAGE")
11    res = suma_millage(m)
12    print("La suma del millage fue:", res)
13
14 main()

```



```

SUMA DE MILLAGE
[15944. 9083. 12456. ... 7010. 5007. 5007.]
La suma del millage fue: 419547822.0

```

```

1 def cantidad_autos(m):
2     arreglo = (m[:, 0])
3     lista = list(set(arreglo))
4     lista.sort()
5     #print(lista)
6     for ele in lista:
7         condicion = (m[:, 0] == ele)
8         matriz = m[condicion]
9         #print(matriz)
10        longitud = matriz.shape[0]
11        print("%i : %i" % (int(ele), longitud))
12
13 def main():
14     m = leer_datos()
15     # print(m)
16     print("\nCANTIDAD DE AUTOS POR AÑO")
17     cantidad_autos(m)
18
19 main()

```



```

CANTIDAD DE AUTOS POR AÑO
2002 : 3
2003 : 3
2004 : 4
2005 : 16
2006 : 13
2007 : 32
2008 : 57
2009 : 91
2010 : 67
2011 : 94
2012 : 115
2013 : 609
2014 : 805
2015 : 1368

```

```

2016 : 2331
2017 : 4888
2018 : 4014
2019 : 3194
2020 : 258

```

```

1 def promedio_mpg(m):
2     fuelType = {1: "Disel" , 2 : "Eléctrico", 3 : "Híbrido", 4 : "Otro", 5 : "Gasolina"}
3     print("fuelType : promedio mpg")
4     for key in fuelType:
5         condicion = ((m[:, 4]) == key)
6         matriz = m[condicion]
7         #print(matriz)
8         arreglo = matriz[:, 6]
9         #print(arreglo)
10        if arreglo.size > 0:
11            promedio = np.mean(arreglo)
12        else:
13            promedio = 0
14        print("%s. %s \t %.2f mpg" % (str(key), fuelType[key], promedio))
15
16 def main():
17     m = leer_datos()
18     # print(m)
19     print("\nPROMEDIO DE MILLAGE POR GALON DE CADA TIPO DE COMBUSTIBLE")
20     promedio_mpg(m)
21
22 main()

```



```

PROMEDIO DE MILLAGE POR GALON DE CADA TIPO DE COMBUSTIBLE
fuelType : promedio mpg
1. Disel          62.23 mpg
2. Eléctrico      67.30 mpg
3. Híbrido        88.77 mpg
4. Otro           67.30 mpg
5. Gasolina       55.81 mpg

```

```

1
2 def imprime_matriz(m):
3     lista = ["Automático", "Manual", "Semi-auto"]
4     cont = 0
5     print("\t\t MPG promedio Motor más grande Suma de impuestos")
6     for ren in range (m.shape[0]): #renglones
7         print(lista[cont], end= " ")
8         cont = cont + 1
9         for col in range (m.shape[1]): #columnas
10            print(" \t %.1f " % m[ren, col], end=" ")
11        print()
12
13 def calculos(m, year):
14     matrizf = []
15     listaT = [1, 2, 3]
16     #print("Lista[precio promedio, suma_productos, tamaño máximo]")
17     for elemento in listaT:
18         #print()
19         #print(str(elemento) + " " + str(year))
20         condicion = (m[:, 2] == elemento) & (m[:, 0] == year)
21         # Genera una nueva matriz que cumple con la condicion
22         matriz = m[condicion]
23         #print(matriz)
24         lista = []
25
26         # MPG PROMEDIO
27         arreglo = matriz[:, 6]
28         #print(arreglo)
29         if arreglo.size > 0:
30             promedio = np.mean(arreglo)

```

```

31     else:
32         promedio = 0
33         #print("Promedio mpg: ", promedio)
34         lista.append(promedio)
35         #print(lista)
36
37     # TAMAÑO DEL MOTOR MÁS GRANDE
38     arreglo = matriz[:,7]
39     #print(arreglo)
40     if arreglo.size > 0:
41         res = np.max(arreglo)
42     else:
43         res = 0
44     #print("Tamaño de motor más grande es: ", res)
45     lista.append(res)
46     #print(lista)
47
48     # SUMA DE IMPUESTOS
49     arreglo = matriz[:, 5]
50     #print(arreglo)
51     if arreglo.size > 0:
52         suma = np.sum(arreglo)
53     else:
54         suma = 0
55     #print("La suma de millages es: ", suma)
56     lista.append(suma)
57     #print(lista)
58     matrizf.append(lista)
59     return matrizf
60
61
62 def main():
63     m = leer_datos()
64     #print(m)
65
66     print("\nAÑO: 2011")
67     matriz = calculos(m, 2011)
68     #print(matriz)
69     #CONVERTIR LA MATRIZ A UNA MATRIZ DE NP
70     matriznp = np.array(matriz, dtype = "float")
71     imprime_matriz(matriznp)
72
73     print("\nAÑO: 2016")
74     matriz = calculos(m, 2016)
75     #print(matriz)
76     #CONVERTIR LA MATRIZ A UNA MATRIZ DE NP
77     matriznp = np.array(matriz, dtype = "float")
78     imprime_matriz(matriznp)
79
80     print("\nAÑO: 2018")
81     matriz = calculos(m, 2018)
82     #print(matriz)
83     #CONVERTIR LA MATRIZ A UNA MATRIZ DE NP
84     matriznp = np.array(matriz, dtype = "float")
85     imprime_matriz(matriznp)
86 main()

```



AÑO: 2011			
	MPG	promedio Motor más grande	Suma de impuestos
Automático	45.2	2.0	2370.0
Manual	50.7	2.5	11075.0
Semi-auto	0.0	0.0	0.0

  

AÑO: 2016			
	MPG	promedio Motor más grande	Suma de impuestos
Automático	52.3	5.0	27070.0
Manual	62.4	5.0	106585.0
Semi-auto	53.1	5.0	18530.0

AÑO: 2018

	MPG promedio	Motor más grande	Suma de impuestos
Automático	49.5	5.0	31155.0
Manual	59.6	5.0	527580.0
Semi-auto	52.2	5.0	26440.0

```

1 def imprime_matriz(m):
2     lista = ["Automático", "Manual", "Semi-auto"]
3     cont = 0
4     print("\t\t Precio promedio Suma millajes Impuesto máximo")
5     for ren in range(m.shape[0]): #renglones
6         print(lista[cont], end= " ")
7         cont = cont + 1
8         for col in range(m.shape[1]): #columnas
9             print(" \t %.1f " % m[ren, col], end=" ")
10        print()
11
12 def calculos(m, year):
13     matrizf = []
14     listaT = [1, 2, 3]
15     #print("Lista[precio promedio, suma_productos, tamaño máximo")
16     for elemento in listaT:
17         #print()
18         #print(str(elemento) + " " + str(year))
19         condicion = (m[:, 2] == elemento) & (m[:, 0] == year)
20         # Genera una nueva matriz que cumple con la condicion
21         matriz = m[condicion]
22         #print(matriz)
23         lista = []
24
25         # Precio PROMEDIO
26         arreglo = matriz[:, 1]
27         #print(arreglo)
28         if arreglo.size > 0:
29             promedio = np.mean(arreglo)
30         else:
31             promedio = 0
32         #print("Precio promedio: ", promedio)
33         lista.append(promedio)
34         #print(lista)
35
36         # SUMA DE MILLAJE
37         arreglo = matriz[:, 3]
38         #print(arreglo)
39         if arreglo.size > 0:
40             res = np.sum(arreglo)
41         else:
42             res = 0
43         #print("La suma del millage es: ", res)
44         lista.append(res)
45         #print(lista)
46
47         # IMPUESTO MÁXIMO
48         arreglo = matriz[:, 5]
49         #print(arreglo)
50         if arreglo.size > 0:
51             res = np.max(arreglo)
52         else:
53             res = 0
54         #print("EL impuesto máximo es: ", res)
55         lista.append(res)
56         #print(lista)
57         matrizf.append(lista)
58     return matrizf
59
60
61 def main():

```

```

62 m = leer_datos()
63 #print(m)
64
65 print("\nAÑO: 2016")
66 matriz = calculos(m, 2016)
67 #print(matriz)
68 #CONVERTIR LA MATRIZ A UNA MATRIZ DE NP
69 matriznp = np.array(matriz, dtype = "float")
70 imprime_matriz(matriznp)
71
72 print("\nAÑO: 2018")
73 matriz = calculos(m, 2018)
74 #print(matriz)
75 #CONVERTIR LA MATRIZ A UNA MATRIZ DE NP
76 matriznp = np.array(matriz, dtype = "float")
77 imprime_matriz(matriznp)
78 main()

```

```

↗
   year price transmission mileage fuelType tax mpg engineSize
0    2017  12000             1   15944         5  150  57.7      1.0
1    2018  14000             2    9083         5  150  57.7      1.0
2    2017  13000             2   12456         5  150  57.7      1.0
3    2019  17500             2   10460         5  145  40.3      1.5
4    2019  16500             1    1482         5  145  48.7      1.0
...    ...    ...          ...    ...         ...    ...    ...
17957 2017   8999             2   16700         5  150  47.1      1.4
17958 2014   7499             2   40700         5   30  57.7      1.0
17959 2015   9999             2    7010         1   20  67.3      1.6
17960 2018   8299             2    5007         5  145  57.7      1.2
17961 2015   8299             2    5007         5   22  57.7      1.0

```

[17962 rows x 8 columns]

AÑO: 2016

	Precio promedio	Suma millajes	Impuesto máximo
Automático	12906.4	5943277.0	570.0
Manual	10249.5	59611316.0	580.0
Semi-auto	12564.2	4107638.0	570.0

AÑO: 2018

	Precio promedio	Suma millajes	Impuesto máximo
Automático	16958.0	3584203.0	150.0
Manual	12768.9	56462493.0	150.0
Semi-auto	16443.4	3004389.0	150.0

1 Comienza a programar o [generar](#) con IA.