

Kevin Israel Lopez Mendoza 1719110155 29 de octubre

En la universidad "ACME" se desea calcular el promedio de cada uno de los alumnos, y el área de control escolar genero el archivo `calificaciones_v2.csv` para calcular el promedio por alumno, y tener un archivo con formato similar al archivo `calificaciones_promedio_v2.csv`.

Librerias

```
In [59]: import pandas as pd  
import numpy as np  
import csv
```

```
In [60]: np.__version__
```

```
Out[60]: '1.23.3'
```

```
In [61]: pd.__version__
```

```
Out[61]: '1.5.0'
```

```
In [62]: csv.__version__
```

```
Out[62]: '1.0'
```

Abrir el Archivo csv con Pandas

```
In [63]: file = pd.read_csv("calificaciones_v2.csv")
```

```
In [64]: file.head(10)
```

```
Out[64]: matricula 11111111
```

0	nombre	Alumno 1
1	grupo	TI11
2	Materia 1	1
3	Materia 2	8
4	Materia 3	1
5	Materia 4	7
6	Materia 5	2
7	Materia 5	2
8	estado	regular
9	matricula	11111112

Eliminar los registros duplicados

```
In [65]: file = file.drop([7],axis =0)
file = file.drop([431],axis =0)
file = file.drop([106],axis=0)
file = file.drop([107],axis=0)
```

```
In [66]: file = file.reset_index(drop=True)
```

```
In [67]: file.head(10)
```

```
Out[67]: matricula 11111111
```

	matricula	11111111
0	nombre	Alumno 1
1	grupo	TI11
2	Materia 1	1
3	Materia 2	8
4	Materia 3	1
5	Materia 4	7
6	Materia 5	2
7	estado	regular
8	matricula	11111112
9	nombre	Alumno 2

```
In [68]: file.tail(12)
```

```
Out[68]: matricula 11111111
```

	matricula	11111111
417	Materia 4	3
418	Materia 5	9
419	estado	regular
420	matricula	11111158
421	nombre	Alumno 48
422	grupo	IDGS71
423	Materia 1	4
424	Materia 2	4
425	Materia 3	10
426	Materia 4	6
427	Materia 5	8
428	estado	regular

Reindexar el ID

```
In [69]: file = file.reset_index(drop=True)
```

```
In [70]: exportacion = file.to_csv("indexado.csv")
```

```
In [71]: file.head(8)
```

```
Out[71]: matricula 11111111
```

	nombre	Alumno 1
0	grupo	TI11
1	Materia 1	1
2	Materia 2	8
3	Materia 3	1
4	Materia 4	7
5	Materia 5	2
6	estado	regular

Cambiar los valores NaN a 0

```
In [72]: file = file.fillna(0)
```

dividir en dos dataframe

```
In [73]: file_1 = file.iloc[:231,:]  
file_1
```

```
Out[73]: matricula 11111111
```

	nombre	Alumno 1
0	grupo	TI11
1	Materia 1	1
2	Materia 2	8
3	Materia 3	1
4
226	grupo	TI41
227	Materia 1	0
228	Materia 2	5
229	Materia 3	7
230	Materia 4	2

231 rows × 2 columns

```
In [74]: file_2 = file.iloc[231:,:]  
file_2
```

```
Out[74]:      matricula  11111111
  231    estado    regular
  232  matricula  11111137
  233    nombre  Alumno 27
  234    grupo     TI41
  235  Materia 1        0
...
  424  Materia 2        4
  425  Materia 3       10
  426  Materia 4        6
  427  Materia 5        8
  428    estado    regular
```

198 rows × 2 columns

Agregar materia Faltante del dataframe 1 (file_1)

```
In [75]: file_1.loc[231]=["Materia 5",0]
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_41980\1948037545.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
file_1.loc[231]=["Materia 5",0]
```

```
In [76]: file_1
```

```
Out[76]:      matricula  11111111
  0    nombre  Alumno 1
  1    grupo     TI11
  2  Materia 1        1
  3  Materia 2        8
  4  Materia 3        1
...
  227  Materia 1        0
  228  Materia 2        5
  229  Materia 3        7
  230  Materia 4        2
  231  Materia 5        0
```

232 rows × 2 columns

Concatenar los dos datafrane

```
In [77]: frames = [file_1,file_2]
file = pd.concat(frames,sort=False)
```

```
In [78]: file
```

```
Out[78]:      matricula  11111111
```

	matricula	11111111
0	nombre	Alumno 1
1	grupo	TI11
2	Materia 1	1
3	Materia 2	8
4	Materia 3	1
...
424	Materia 2	4
425	Materia 3	10
426	Materia 4	6
427	Materia 5	8
428	estado	regular

430 rows × 2 columns

```
In [79]: file = file.reset_index(drop=True)
file
```

```
Out[79]:      matricula  11111111
```

	matricula	11111111
0	nombre	Alumno 1
1	grupo	TI11
2	Materia 1	1
3	Materia 2	8
4	Materia 3	1
...
425	Materia 2	4
426	Materia 3	10
427	Materia 4	6
428	Materia 5	8
429	estado	regular

430 rows × 2 columns

```
In [80]: new_file = file.to_csv("indexado.csv")
```

Separar los dataframe en 2

```
In [81]: file_1 = file.iloc[:322,:]  
file_1
```

```
Out[81]:      matricula  11111111  
              0    nombre  Alumno 1  
              1    grupo    TI11  
              2  Materia 1       1  
              3  Materia 2       8  
              4  Materia 3       1  
             ...        ...  ...  
            317  Materia 1       7  
            318  Materia 2       1  
            319  Materia 3       4  
            320  Materia 4       0  
            321  Materia 5       6
```

322 rows × 2 columns

```
In [82]: file_2 = file.iloc[322:,:]  
file_2
```

```
Out[82]:      matricula  11111111  
            322  matricula  11111147  
            323    nombre  Alumno 37  
            324    grupo    IDGS91  
            325  Materia 1       6  
            326  Materia 2       6  
            ...        ...  ...  
            425  Materia 2       4  
            426  Materia 3      10  
            427  Materia 4       6  
            428  Materia 5       8  
            429    estado    regular
```

108 rows × 2 columns

Añadir estado faltante

```
In [83]: file_1.loc[322]=["estado","regular"]
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_41980\2453132803.py:1: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy  
file_1.loc[322]=["estado","regular"]
```

Concatenar los dos dataframes

```
In [84]: frames = [file_1,file_2]  
file = pd.concat(frames,sort=False)
```

```
In [85]: file = file.reset_index(drop=True)  
file
```

```
Out[85]:
```

	matricula	11111111
0	nombre	Alumno 1
1	grupo	TI11
2	Materia 1	1
3	Materia 2	8
4	Materia 3	1
...
426	Materia 2	4
427	Materia 3	10
428	Materia 4	6
429	Materia 5	8
430	estado	regular

431 rows × 2 columns

Exportar a CSV

```
In [86]: archive = file.to_csv("indexado.csv")
```

Abrir archivo con Numpy

```
In [87]: with open("indexado.csv") as file:  
    datos = list(csv.reader(file,delimiter=','))
```

```
In [88]: type(datos)
```

```
Out[88]: list
```

Convertir a array

```
In [89]: datos = np.array(datos)

In [90]: type(datos)

Out[90]: numpy.ndarray

In [91]: datos[:10]

Out[91]: array([['', 'matricula', '11111111'],
   ['0', 'nombre', 'Alumno 1'],
   ['1', 'grupo', 'TI11'],
   ['2', 'Materia 1', '1'],
   ['3', 'Materia 2', '8'],
   ['4', 'Materia 3', '1'],
   ['5', 'Materia 4', '7'],
   ['6', 'Materia 5', '2'],
   ['7', 'estado', 'regular'],
   ['8', 'matricula', '11111112']], dtype='<U9')
```

Obtener las calificaciones

```
In [92]: filtro = np.array([False, True, True])
datos = datos[:, filtro]

In [93]: datos[:10]

Out[93]: array([['matricula', '11111111'],
   ['nombre', 'Alumno 1'],
   ['grupo', 'TI11'],
   ['Materia 1', '1'],
   ['Materia 2', '8'],
   ['Materia 3', '1'],
   ['Materia 4', '7'],
   ['Materia 5', '2'],
   ['estado', 'regular'],
   ['matricula', '11111112']], dtype='<U9')
```

Cabeceras

```
In [94]: cabeceras = datos.T[0]

In [95]: cabeceras[:10]

Out[95]: array(['matricula', 'nombre', 'grupo', 'Materia 1', 'Materia 2',
   'Materia 3', 'Materia 4', 'Materia 5', 'estado', 'matricula'],
  dtype='<U9')
```

Contar cabeceras

```
In [96]: cabeceras.size

Out[96]: 432

In [97]: np.unique(cabeceras)

Out[97]: array(['Materia 1', 'Materia 2', 'Materia 3', 'Materia 4', 'Materia 5',
   'estado', 'grupo', 'matricula', 'nombre'], dtype='<U9')
```

```
In [98]: np.unique(cabeceras).size
```

```
Out[98]: 9
```

```
In [99]: cabeceras = cabeceras[:9]
```

```
In [100...]: cabeceras
```

```
Out[100]: array(['matricula', 'nombre', 'grupo', 'Materia 1', 'Materia 2',  
                 'Materia 3', 'Materia 4', 'Materia 5', 'estado'], dtype='<U9')
```

Calcular el numero de registros

```
In [101...]: no_cabecera = cabeceras.size  
no_cabecera
```

```
Out[101]: 9
```

```
In [102...]: no_registros = datos.T[0].size  
no_registros
```

```
Out[102]: 432
```

```
In [103...]: filas = no_registros/no_cabecera  
filas
```

```
Out[103]: 48.0
```

Eliminar las cabeceras del array

```
In [104...]: filtro = np.array([False,True])  
datos_sin_cabeceras = datos[:,filtro]
```

```
In [105...]: datos_sin_cabeceras[:10]
```

```
Out[105]: array([[['11111111'],  
                  ['Alumno 1'],  
                  ['TI11'],  
                  ['1'],  
                  ['8'],  
                  ['1'],  
                  ['7'],  
                  ['2'],  
                  ['regular'],  
                  ['11111112']], dtype='<U9')
```

```
In [106...]: datos_sin_cabeceras.shape
```

```
Out[106]: (432, 1)
```

Cambiar la forma del array

```
In [107...]: datos_sin_cabeceras = datos_sin_cabeceras.reshape(int(filas),int(no_cabecera))
```

```
In [108...]: datos_sin_cabeceras.shape
```

```
Out[108]: (48, 9)
```

```
In [109...]: datos_sin_cabeceras[:5]
```

```
Out[109]: array([['11111111', 'Alumno 1', 'TI11', '1', '8', '1', '7', '2',
   'regular'],
   ['11111112', 'Alumno 2', 'TI11', '10', '7', '1', '8', '5',
   'regular'],
   ['11111113', 'Alumno 3', 'TI11', '0', '1', '1', '3', '0',
   'regular'],
   ['11111114', 'Alumno 4', 'TI11', '0', '4', '3', '2', '7',
   'regular'],
   ['11111115', 'Alumno 5', 'TI11', '9', '10', '3', '9', '2',
   'regular']], dtype='<U9')
```

Separar los datos de los alumnos

```
In [110...]: datos_alumnos = np.array(datos_sin_cabeceras[:, :3])
```

```
In [111...]: datos_alumnos[:10]
```

```
Out[111]: array([['11111111', 'Alumno 1', 'TI11'],
   ['11111112', 'Alumno 2', 'TI11'],
   ['11111113', 'Alumno 3', 'TI11'],
   ['11111114', 'Alumno 4', 'TI11'],
   ['11111115', 'Alumno 5', 'TI11'],
   ['11111116', 'Alumno 6', 'TI11'],
   ['11111117', 'Alumno 7', 'TI11'],
   ['11111118', 'Alumno 8', 'TI11'],
   ['11111119', 'Alumno 9', 'TI11'],
   ['11111120', 'Alumno 10', 'TI11']], dtype='<U9')
```

Obtener las calificaciones de los alumnos

```
In [112...]: calificaciones = np.array(datos_sin_cabeceras[:, 3:8], dtype=float)
```

```
In [113...]: calificaciones[:10]
```

```
Out[113]: array([[ 1.,  8.,  1.,  7.,  2.],
   [10.,  7.,  1.,  8.,  5.],
   [ 0.,  1.,  1.,  3.,  0.],
   [ 0.,  4.,  3.,  2.,  7.],
   [ 9., 10.,  3.,  9.,  2.],
   [ 1.,  3.,  8.,  1.,  7.],
   [ 4.,  7.,  1.,  6.,  5.],
   [10.,  1.,  7., 10.,  0.],
   [ 5.,  5.,  4.,  0.,  3.],
   [ 5.,  7.,  3.,  3.,  4.]])
```

Calcular el promedio

```
In [115...]: promedio = []
for index in range(int(filas)):
    promedio.append(["regular", calificaciones[index].sum()/5])
promedio = np.array(promedio)
```

```
In [116...]: promedio[:5]
```

```
Out[116]: array([['regular', '3.8'],
   ['regular', '6.2'],
   ['regular', '1.0'],
   ['regular', '3.2'],
   ['regular', '6.6']], dtype='<U32')
```

```
In [117... promedio.shape
```

```
Out[117]: (48, 2)
```

Concatenar valor a Cabeceras

```
In [124... cabeceras = np.concatenate((cabeceras, ["Promedio"]), axis=0, dtype=str)
```

```
In [125... cabeceras
```

```
Out[125]: array(['matricula', 'nombre', 'grupo', 'Materia 1', 'Materia 2',
   'Materia 3', 'Materia 4', 'Materia 5', 'estado', 'Promedio'],
  dtype='<U9')
```

Concatener los arreglos

```
In [119... todos = np.concatenate((datos_alumnos, calificaciones, promedio), axis=1, dtype=str)
```

```
In [120... todos[:5]
```

```
Out[120]: array([['11111111', 'Alumno 1', 'TI11', '1.0', '8.0', '1.0', '7.0',
   '2.0', 'regular', '3.8'],
   ['11111112', 'Alumno 2', 'TI11', '10.0', '7.0', '1.0', '8.0',
   '5.0', 'regular', '6.2'],
   ['11111113', 'Alumno 3', 'TI11', '0.0', '1.0', '1.0', '3.0',
   '0.0', 'regular', '1.0'],
   ['11111114', 'Alumno 4', 'TI11', '0.0', '4.0', '3.0', '2.0',
   '7.0', 'regular', '3.2'],
   ['11111115', 'Alumno 5', 'TI11', '9.0', '10.0', '3.0', '9.0',
   '2.0', 'regular', '6.6']], dtype='<U32')
```

Dataframe final

```
In [126... dataframe = pd.DataFrame(data = todos, columns = cabeceras , dtype=str)
```

```
In [128... dataframe.head(10)
```

Out[128]:

	matricula	nombre	grupo	Materia 1	Materia 2	Materia 3	Materia 4	Materia 5	estado	Promedio
0	11111111	Alumno 1	TI11	1.0	8.0	1.0	7.0	2.0	regular	3.8
1	11111112	Alumno 2	TI11	10.0	7.0	1.0	8.0	5.0	regular	6.2
2	11111113	Alumno 3	TI11	0.0	1.0	1.0	3.0	0.0	regular	1.0
3	11111114	Alumno 4	TI11	0.0	4.0	3.0	2.0	7.0	regular	3.2
4	11111115	Alumno 5	TI11	9.0	10.0	3.0	9.0	2.0	regular	6.6
5	11111116	Alumno 6	TI11	1.0	3.0	8.0	1.0	7.0	regular	4.0
6	11111117	Alumno 7	TI11	4.0	7.0	1.0	6.0	5.0	regular	4.6
7	11111118	Alumno 8	TI11	10.0	1.0	7.0	10.0	0.0	regular	5.6
8	11111119	Alumno 9	TI11	5.0	5.0	4.0	0.0	3.0	regular	3.4
9	11111120	Alumno 10	TI11	5.0	7.0	3.0	3.0	4.0	regular	4.4

Exportar a CSV

In []: `file = dataframe.to_csv("calificaciones_finales_v2.csv")`