Domumentacion Proyecto 1

Proyecto 1 - OLC2 2S EVD 2025

Victor Abdiel Lux Juracán - 201403946

Victor Ronaldo Gomez Lara - 201114493

Carga de Datos

primero cargamos los datos del archivo csv para ver las columnas

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files
```

```
# Subir archivo
uploaded = files.upload()
filename = list(uploaded.keys())[0]
df = pd.read_csv(filename)
```

```
df
```

Pasos siguientes: Generar código con df | New interactive sheet

Se puede observar cuantas columnas y filas contiene el archivo lo primero que hacemos es verificar si estan las columnas necesarias

- Promedio actual del estudiante (promedio_actual).
- Porcentaje de asistencia a clases (asistencia_clases).
- Porcentaje de tareas entregadas (tareas_entregadas).
- Participación en clase (participacion_clase).
- Horas dedicadas al estudio (horas_estudio).
- Promedio en evaluaciones parciales (promedio_evaluaciones).
- Cantidad de cursos reprobados o retirados (cursos_reprobados).
- Participación en actividades extracurriculares(actividades_extracurriculares).
- Cantidad de reportes disciplinarios (reportes_disciplinarios).

esto en python lo hacemos asi

```
    columnas_presentes = set(df.columns)
    columnas_faltantes = [col for col in columnas_val if col not in colum
    if columnas_faltantes:
        return jsonify({
            "error": "Faltan columnas requeridas",
            "columnas_faltantes": columnas_faltantes
        }), 400
```

```
df.info();
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 14 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   asistencia_clases               200 non-null    object
 1   tareas_entregadas               200 non-null    int64
 2   participacion_clase             200 non-null    float64
 3   horas_estudio                   200 non-null    float64
 4   promedio_evaluaciones           200 non-null    float64
 5   cursos_reprobados               200 non-null    int64
```

```
 6   actividades_extracurriculares  200 non-null    object
 7   reportes_disciplinarios        200 non-null    int64
 8   riesgo                         200 non-null    object
 9   carnet                         175 non-null    object
 10  first_name                     200 non-null    object
 11  last_name                      200 non-null    object
 12  promedio_actual                174 non-null    float64
 13  gender                         200 non-null    object
dtypes: float64(4), int64(3), object(7)
memory usage: 22.0+ KB
```

```
df.describe()
```

primero vamos a eliminar datos duplicados

```
print(f"Duplicados antes: {df.duplicated().sum()}")
df = df.drop_duplicates().reset_index(drop=True)
print(f"Duplicados después: {df.duplicated().sum()}")
print(f"Filas finales: {df.shape[0]}")
```

```
Duplicados antes: 0
Duplicados después: 0
Filas finales: 200
```

Despues empezamos a convertir las columnas numericas a int, las columnas que vamos a idetificar como numericas son promedio_actual, asistencia_clases, tareas_entregadas, participacion_clase, horas_estudio, promedio_evaluaciones, cursos_reprobados, reportes_disciplinarios, los que son carne ni los categoricos como actividades_extracurriculares, generos, etc.

```
columnas_numericas = [ # primero pongo las columnas que voy a convertir
    'promedio_actual', 'asistencia_clases', 'tareas_entregadas',
    'participacion_clase', 'horas_estudio', 'promedio_evaluaciones',
    'cursos_reprobados', 'reportes_disciplinarios'
]
```

```
for col in columnas_numericas:
    df[col] = pd.to_numeric(df[col], errors='coerce')# luego los covierto
# visualizo cuales columnas se convirtieron
df.info();
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 14 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   asistencia_clases               199 non-null    float64
 1   tareas_entregadas               200 non-null    int64  
 2   participacion_clase             200 non-null    float64
 3   horas_estudio                   200 non-null    float64
 4   promedio_evaluaciones           200 non-null    float64
 5   cursos_reprobados               200 non-null    int64  
 6   actividades_extracurriculares   200 non-null    object 
 7   reportes_disciplinarios         200 non-null    int64  
 8   riesgo                          200 non-null    object 
 9   carnet                          175 non-null    object 
 10  first_name                      200 non-null    object 
 11  last_name                       200 non-null    object 
 12  promedio_actual                 174 non-null    float64
 13  gender                          200 non-null    object 
dtypes: float64(5), int64(3), object(6)
memory usage: 22.0+ KB
```

luego vamos a ver si hay valores faltantes y si existe lo remplazamos por NaN que es nulo luego usamos media (mean), median() o mode(), segun columna.

Analisis horas de estudio

```
import numpy as np
import matplotlib.pyplot as plt
data = df['horas_estudio'].dropna().values
media = np.mean(data)
mediana = np.median(data)

plt.figure()
plt.hist(data, bins=20)
plt.axvline(media, linestyle='--', label='Media')
plt.axvline(mediana, linestyle='-', label='Mediana')
plt.title('Distribución de Horas de Estudio')
plt.xlabel('Horas de estudio')
plt.ylabel('Número de estudiantes')
plt.legend()
plt.show()
```

Cerrar

```
Para ver como se comportan los datos, se
que pasan varias horas estudiando, y hay
para que los datos null podamos llenarlo
la mediana, para no tener extremos muy b
archivo en particular la media y la medi
es mejor la mediana.

---

#### Analisis reportes disciplinarios

De la misma manera que no hay tantos est
reportes, la mayoria de estudiantes no t
que la media, mediana y la moda se van h
que el valor de la media puedequedar muy
valores  0 y 1, por ello decidimos utili
de los datos.
```

Para ver como se comportan los datos, se puede observar que hay estudiantes que pasan varias horas estudiando, y hay algunos que no estudian casi nada, para que los datos null podamos llenarlos sin inflarlo tanto es mejor usar la mediana, para no tener extremos muy bajo y tampoco tan altos, para este archivo en particular la media y la mediana son casi iguales pero siempre es mejor la mediana.

Analisis reportes disciplinarios

De la misma manera que no hay tantos estudiantes con una catidad grande de reportes, la mayoria de estudiantes no tiene reportes y podemos visualizar que la media, mediana y la moda se van hacia cero, usaremos la mediana ya que el valor de la media puedequedar muy disperso al existir muchos valores 0 y 1,

por ello decidimos utilizar el valor que represante el 50% de los datos.

```
data_1 = df['reportes_disciplinarios'].dropna().values

media_1 = np.mean(data_1)
mediana_1 = np.median(data_1)

plt.figure()
plt.hist(data, bins=20)
plt.axvline(media_1, linestyle='--', label='Media')
plt.axvline(mediana_1, linestyle='-', label='Mediana')
plt.title('Distribución de reportes')
plt.xlabel('reportes por estudiantes')
plt.ylabel('Número de estudiantes')
plt.legend()
plt.show()
```