procesamiento1

June 9, 2025

1 Análisis de la tabla clasificaciones

1.1 1. Importación de datos

En primer lugar, es necesario importar los datos de las calificaciones y de los cursos con la finalidad de verificar que los códigos de curso mencionados en el primer dataset existan realmente.

[2]:
```
# Instalar la librería "polars" en el entorno de Python
%pip install polars
```

```
Collecting polars
  Downloading polars-1.30.0-cp39-abi3-win_amd64.whl.metadata (15 kB)
Downloading polars-1.30.0-cp39-abi3-win_amd64.whl (36.4 MB)
   ---------------------------------------- 0.0/36.4 MB ? eta -:--:--
   -------- ------------------------------- 7.3/36.4 MB 34.9 MB/s eta 0:00:01
   ----------------- ---------------------- 16.5/36.4 MB 40.0 MB/s eta 0:00:01
   ------------------------------ -------- 28.6/36.4 MB 45.3 MB/s eta 0:00:01
   ---------------------------------------  36.2/36.4 MB 46.0 MB/s eta 0:00:01
   ---------------------------------------- 36.4/36.4 MB 39.8 MB/s eta 0:00:00
Installing collected packages: polars
Successfully installed polars-1.30.0
Note: you may need to restart the kernel to use updated packages.
```

[10]:
```
# Importar la librería polaris
import polars as pl

# Leer CSVs y asegurar que los códigos sean tratados como texto
df_notas = pl.read_csv(
    "../Data/DB_Notas.csv",
    schema_overrides={
        "cod_asignatura": pl.Utf8,
        "cod_alumno": pl.Utf8,
        "cod_plan": pl.Utf8
    },
    separator=";",
    infer_schema_length=1000
)
df_cursos = pl.read_csv(
    "../Data/DB_Cursos.csv",
```

```
    schema_overrides={"Codigo": pl.Utf8},
    separator=";"
)
```

[11]:
```
# Mostrar las primeras filas de los DataFrames
print("Notas DataFrame:")
print(df_notas.head())
print("\nCursos DataFrame:")
print(df_cursos.head())
```

```
Notas DataFrame:
shape: (5, 7)

 cod_semestre  cod_facultad  cod_escuela  cod_plan  cod_asignatur
cod_alumno  val_calific_
 ---           ---           ---          ---       a              ---
 final
 i64           i64           i64          str       ---            str
 ---
                                                    str
 i64


 20191         20            1            2009      201003         8200187
 16
 20191         20            1            2009      201101         6200206
 15
 20191         20            1            2009      201101         7200092
 6
 20191         20            1            2009      201101         110749
 14
 20191         20            1            2009      201101         4200074
 12



Cursos DataFrame:
shape: (5, 6)

 Codigo  Nombre                    Ciclo  Tipo         Plan  Escuela
 ---     ---                       ---    ---          ---   ---
 str     str                       i64    str          i64   i64

 201001  ALGORÍTMICA I             1      Obligatorio  2009  1
 201003  CALCULO I                 1      Obligatorio  2009  1
 201004  MATEMÁTICA BÁSICA I       1      Obligatorio  2009  1
 201007  COMPUTACIÓN E INFORMÁTICA  1      Obligatorio  2009  1
```

```
201008   TEORIA DE SISTEMAS          1       Obligatorio   2009   1
```

1.2 2. Cruce de tablas

En segundo lugar, se procede a cruzar las tablas para verificar los cursos realmente existentes en los planes vigentes de las escuelas de ingeniería de sistemas y software.

[12]:
```
# Hacer la unión (join) para añadir el nombre del curso
df_completo = df_notas.join(
    df_cursos,
    left_on="cod_asignatura",
    right_on="Codigo",
    how="left"
)

# Renombrar la columna del nombre del curso
df_completo = df_completo.with_columns(
    pl.col("Nombre").alias("nombre_Curso")
)
```

[13]:
```
# Mostrar las primeras filas del DataFrame completo
print("DataFrame completo:")
print(df_completo.head())
```

```
DataFrame completo:
shape: (5, 13)


 cod_semestr   cod_faculta   cod_escuel   cod_plan   …   Tipo         Plan
Escuela   nombre_Cur
 e             d             a            ---            ---          ---
---       so
 ---           ---           ---          str            str          i64
i64       ---
 i64           i64           i64
 str


 20191         20            1            2009       …   Obligatori   2009   1
 CALCULO I
                                                        o

 20191         20            1            2009       …   Obligatori   2009   1
 GERENCIA
                                                        o
 INFORMÁTIC

 A
```

```
20191         20            1             2009       …  Obligatori   2009   1
GERENCIA                                                o
INFORMÁTIC
A
20191         20            1             2009       …  Obligatori   2009   1
GERENCIA                                                o
INFORMÁTIC
A
20191         20            1             2009       …  Obligatori   2009   1
GERENCIA                                                o
INFORMÁTIC
A
```

[19]:
```python
# Eliminar las filas con valores nulos en la columna 'nombre_Curso'
df_completo = df_completo.filter(pl.col("nombre_Curso").is_not_null())
```

[ ]:
```python
# Eliminar los espacios en blanco al final de algunas celdas de la columna␣
↪'cod_plan'
df_completo = df_completo.with_columns(
    pl.col("cod_plan")
      .str.replace(r"\s+$", "", literal=False)
)
```

[29]:
```python
# Enlistar el nombre de las columnas del DataFrame completo
print("\nColumnas del DataFrame completo:")
for idx, col in enumerate(df_completo.columns):
    print(f"{idx+1}. {col}")
```

```
Columnas del DataFrame completo:
1. cod_semestre
2. cod_facultad
3. cod_escuela
4. cod_plan
5. cod_asignatura
6. cod_alumno
7. val_califc_final
8. Nombre
9. Ciclo
10. Tipo
```

11. Plan
12. Escuela
13. nombre_Curso

1.3 3. Transformación de datos

En tercer lugar, se categoriza las notas con la finalidad de adecuar el dataset para que sea compatible con el algoritmo J48.

[30]:
```
# Crear un DataFrame con las columnas de interés
df_interes = df_completo.select([
    "cod_escuela",
    "cod_plan",
    "cod_asignatura",
    "nombre_Curso",
    "val_calific_final"
])
```

[31]:
```
# Guardar el DataFrame con las columnas de interés en un nuevo CSV
df_interes.write_csv("../Data/DP_notas_verificadas.csv")

print(" ¡Listo! Archivo guardado como 'DP_notas_verificadas.csv'")
```

```
 ¡Listo! Archivo guardado como 'DP_notas_verificadas.csv'
```

[35]:
```
# Agregar una columna con la categorización de las notas
df_interes = df_interes.with_columns(
    pl.when( pl.col("val_calific_final") >= 17 ).then( pl.lit("Alta") )
      .when( pl.col("val_calific_final") >= 14 ).then( pl.lit("Aceptable") )
      .when( pl.col("val_calific_final") >= 11 ).then( pl.lit("Baja") )
      .otherwise( pl.lit("Reprobado") )
      .alias("categoria_nota")
)
```

[36]:
```
# Mostrar las primeras filas del DataFrame con la categorización de las notas
print("\nDataFrame con categorización de notas:")
print(df_interes.head())
```

```
DataFrame con categorización de notas:
shape: (5, 6)


 cod_escuela  cod_plan  cod_asignatura  nombre_Curso  val_calific_final
categoria_nota
 ---          ---       ---             ---           ---
---
 i64          str       str             str           i64
str
```

```
 1            2009       201003          CALCULO I      16
Aceptable
 1            2009       201101          GERENCIA       15
Aceptable
                                         INFORMÁTICA

 1            2009       201101          GERENCIA       6
Reprobado
                                         INFORMÁTICA

 1            2009       201101          GERENCIA       14
Aceptable
                                         INFORMÁTICA

 1            2009       201101          GERENCIA       12
Baja
                                         INFORMÁTICA
```

[38]:
```python
# Crear un DataFrame con las columnas de interés
df_interes2 = df_interes.select([
    "cod_escuela",
    "cod_plan",
    "cod_asignatura",
    "categoria_nota"
])
```

[39]:
```python
# Mostrar las primeras filas del DataFrame con la categorización de las notas
print("\nDataFrame con categorización de notas:")
print(df_interes2.head())
```

```
DataFrame con categorización de notas:
shape: (5, 4)

 cod_escuela  cod_plan  cod_asignatura  categoria_nota
 ---          ---       ---             ---
 i64          str       str             str

 1            2009      201003          Aceptable
 1            2009      201101          Aceptable
 1            2009      201101          Reprobado
 1            2009      201101          Aceptable
 1            2009      201101          Baja
```

[41]:
```python
# Filtrar las filas donde 'cod_plan' contiene solo números
df_interes2 = df_interes2.filter(pl.col("cod_plan").str.contains(r"^\d+$"))

# Cambiar el tipo de dato de la columna 'cod_plan' a numérico
df_interes2 = df_interes2.with_columns(
    pl.col("cod_plan").cast(pl.Int64)
)

# Mostrar los tipos de datos del DataFrame final
print("\nTipos de datos del DataFrame final:")
print(df_interes2.dtypes)
```

```
Tipos de datos del DataFrame final:
[Int64, Int64, String, String]
```

[42]:
```python
# Guardar el DataFrame con las columnas de interés en un nuevo CSV
df_interes2.write_csv("../Data/DP_notas_categorizadas.csv")

print("  ¡Listo! Archivo guardado como 'DP_notas_categorizadas.csv'")
```

```
  ¡Listo! Archivo guardado como 'DP_notas_categorizadas.csv'
```