Procesamiento_dataset

April 22, 2025

1 Análisis del dataset de docentes

[16]:
```
# Importar las librerías para trabajar la limpieza de datos
import pandas as pd
```

1.1 Conversión de los datos a tipos adecuados

[17]:
```
# Guardar dataset en un dataframe
df_docentes = pd.read_csv('Padron_docentes.csv', sep=';', encoding='UTF-8')

# Mostrar el dataframe
print('TIPOS DE DATOS')
print('==============')
df_docentes.info()
```

```
TIPOS DE DATOS
==============
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3256 entries, 0 to 3255
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ITEM                  3256 non-null   int64
 1   CODIGO DOCENTE        3256 non-null   object
 2   APELLIDO PATERNO      3255 non-null   object
 3   APELLIDO MATERNO      3250 non-null   object
 4   NOMBRES               3256 non-null   object
 5   CATEGORIA             3256 non-null   object
 6   TIPO DOC IDENTIDAD    3256 non-null   object
 7   N DOC IDENTIDAD       3256 non-null   object
 8   CORREO INSTITUCIONAL  3256 non-null   object
 9   FACULTAD              3256 non-null   object
dtypes: int64(1), object(9)
memory usage: 254.5+ KB
```

[18]:
```
# Convertir los tipos de datos
df_docentes['CODIGO DOCENTE'] = df_docentes['CODIGO DOCENTE'].astype(str)
df_docentes['APELLIDO PATERNO'] = df_docentes['APELLIDO PATERNO'].astype(str)
```

```
df_docentes['APELLIDO MATERNO'] = df_docentes['APELLIDO MATERNO'].astype(str)
df_docentes['NOMBRES'] = df_docentes['NOMBRES'].astype(str)
df_docentes['CATEGORIA'] = df_docentes['CATEGORIA'].astype(str)
df_docentes['TIPO DOC IDENTIDAD'] = df_docentes['TIPO DOC IDENTIDAD'].
 ↪astype(str)
df_docentes['N DOC IDENTIDAD'] = df_docentes['N DOC IDENTIDAD'].astype(str)
df_docentes['CORREO INSTITUCIONAL'] = df_docentes['CORREO INSTITUCIONAL'].
 ↪astype(str)
df_docentes['FACULTAD'] = df_docentes['FACULTAD'].astype(str)

# Mostrar los tipos de datos
print('TIPOS DE DATOS')
print('==============')
for col in df_docentes.columns:
    print(f'{col}: {df_docentes[col].apply(type).unique()}')
```

```
TIPOS DE DATOS
==============
ITEM: [<class 'int'>]
CODIGO DOCENTE: [<class 'str'>]
APELLIDO PATERNO: [<class 'str'>]
APELLIDO MATERNO: [<class 'str'>]
NOMBRES: [<class 'str'>]
CATEGORIA: [<class 'str'>]
TIPO DOC IDENTIDAD: [<class 'str'>]
N DOC IDENTIDAD: [<class 'str'>]
CORREO INSTITUCIONAL: [<class 'str'>]
FACULTAD: [<class 'str'>]
```

[19]:
```
# Mostrar el dataframe
df_docentes.head(5)
```

[19]:

	ITEM	CODIGO DOCENTE	APELLIDO PATERNO	APELLIDO MATERNO	NOMBRES	CATEGORIA	TIPO DOC IDENTIDAD	N DOC IDENTIDAD	CORREO INSTITUCIONAL	FACULTAD
0	1	0A1018	AGUERO	DEL CARPIO	LIZARDO ELIAS	Asociado T. Completo	DNI	7971397		
1	2	0A7296	AGUIRRE	CASTRO	CARMEN JUDITH	Auxiliar TP. 10hrs.	DNI	23917134		
2	3	0A0034	ANDRADES	SOSA	JOSE IGNACIO	Asociado T. Completo	DNI	25450694		
3	4	004898	ARIAS	MERCADO	LUIS ALBERTO	Asociado T. Completo	DNI	6056404		
4	5	005452	ASPILCUETA	ASPILCUETA	SIMON RICARDO	Asociado T. Completo	DNI	10720986		

```
0      laguerod@unmsm.edu.pe  Ciencias administrativas
1    caguirreca@unmsm.edu.pe  Ciencias administrativas
2     jandrades@unmsm.edu.pe  Ciencias administrativas
3       lariasm@unmsm.edu.pe  Ciencias administrativas
4  raspilcuetaa@unmsm.edu.pe  Ciencias administrativas
```

1.2 Corrección de errores tipográficos o inconsistencias

[20]:
```
# Convertir a formato capitalizable los valores de las columnas "APELLIDO␣
 ↪PATERNO", "APELLIDO MATERNO"
# y "NOMBRES"
df_docentes['APELLIDO PATERNO'] = df_docentes['APELLIDO PATERNO'].str.title()
df_docentes['APELLIDO MATERNO'] = df_docentes['APELLIDO MATERNO'].str.title()
df_docentes['NOMBRES'] = df_docentes['NOMBRES'].str.title()

# Mostrar el dataframe
df_docentes.head(5)
```

[20]:
```
   ITEM CODIGO DOCENTE APELLIDO PATERNO APELLIDO MATERNO        NOMBRES  \
0     1         0A1018           Aguero       Del Carpio  Lizardo Elias
1     2         0A7296          Aguirre           Castro  Carmen Judith
2     3         0A0034         Andrades             Sosa   Jose Ignacio
3     4         004898            Arias          Mercado   Luis Alberto
4     5         005452       Aspilcueta       Aspilcueta  Simon Ricardo

               CATEGORIA TIPO DOC IDENTIDAD N DOC IDENTIDAD  \
0  Asociado T. Completo                 DNI         7971397
1   Auxiliar TP. 10hrs.                 DNI        23917134
2  Asociado T. Completo                 DNI        25450694
3  Asociado T. Completo                 DNI         6056404
4  Asociado T. Completo                 DNI        10720986

        CORREO INSTITUCIONAL                  FACULTAD
0      laguerod@unmsm.edu.pe  Ciencias administrativas
1    caguirreca@unmsm.edu.pe  Ciencias administrativas
2     jandrades@unmsm.edu.pe  Ciencias administrativas
3       lariasm@unmsm.edu.pe  Ciencias administrativas
4  raspilcuetaa@unmsm.edu.pe  Ciencias administrativas
```

1.3 Eliminación de valores nulos o duplicados

[21]:
```
# Contabilizar los campos vacíos dentro de cada columna del dataset
print(f"El dataset tiene {df_docentes.isnull().sum().sum()} campos vacíos.")
```

```
El dataset tiene 0 campos vacíos.
```

[22]:
```
# Identificar si hay celdas vacías en el dataframe
print('CAMPOS VACÍOS')
print('============')
print(df_docentes.isnull().sum())
```

```
CAMPOS VACÍOS
============
ITEM                    0
CODIGO DOCENTE          0
APELLIDO PATERNO        0
APELLIDO MATERNO        0
NOMBRES                 0
CATEGORIA               0
TIPO DOC IDENTIDAD      0
N DOC IDENTIDAD         0
CORREO INSTITUCIONAL    0
FACULTAD                0
dtype: int64
```

1.4 Normalización de los datos

[23]:
```
# Mostrar valores únicos de la columna "CATEGORIA" en una lista
print('VALORES ÚNICOS')
print('============')
print(df_docentes['CATEGORIA'].unique())
```

```
VALORES ÚNICOS
============
['Asociado T. Completo ' 'Auxiliar TP. 10hrs. ' 'Auxiliar T. Completo '
 'Principal T. Completo' 'Principal TP. 20hrs. ' 'Auxiliar TP. 20hrs. '
 'Asociado D . E xclusiva. ' 'Asociado TP. 20hrs. '
 'Principal D . E xclusiva' 'Auxiliar TP. 04hrs. ' 'Principal TP. 16hrs. '
 'Auxiliar D. Exclusiva ' 'E Auxiliar TP. 20hrs. '
 'Principal T. Completo ' 'Principal ' 'Principal TP. 10hrs. '
 'Asociado TP. 10hrs. ' 'Auxiliar TP. 15hrs. ' 'Auxiliar TP. 20hrs.'
 'Auxiliar D. Exclusiva' 'Asociado TP. 20hrs.' 'Auxiliar T. Completo'
 'Asociado T. Completo' 'Asociado D . E xclusiva.' 'Principal TP. 20hrs.'
 'Auxiliar TP. 10hrs.' 'Asociado TP. 06hrs. ' 'Asociado TP. 08hrs. '
 'Auxiliar TP. 06hrs. ' 'Principal TP. 08hrs. ' 'Principal TP. 06hrs. '
 'Asociado TP. 16hrs. ' 'Auxiliar TP. 08hrs. ' 'Asociado TP. 11hrs. '
 'Asociado TP. 10hrs.' 'Auxiliar TP. 08hrs.' 'Auxiliar TP. 20hrs.  '
 'Asociado D . E xclusiva.  ' 'Asociado T. Completo  '
 'Principal TP. 20hrs.  ' 'Asociado TP. 20hrs.  '
 'Principal T. Completo  ' 'Principal D . E xclusiva '
 'Auxiliar T. Completo  ' 'Asociado TP. 15hrs.  ' 'Auxiliar TP. 13hrs.  '
 'Auxiliar D. Exclusiva  ' 'Principal  ' ' Auxiliar T. Completo '
 ' Asociado T. Completo ' ' Principal T. Completo '
 ' Asociado D . E xclusiva. ' ' Auxiliar TP. 20hrs. '
```

```
 ' Asociado TP. 20hrs. ' ' Principal TP. 20hrs. ' ' Asociado TP. 10hrs. '
 ' Auxiliar TP. 10hrs. ' ' Auxiliar D. Exclusiva '
 ' Principal D . E xclusiva' ' Auxiliar TP. 04hrs. '
 'Asociado TP. 10hrs.  ' 'Auxiliar TP. 16hrs. ' 'Asociado TP. 15hrs. '
 'Asociado TP. 18hrs. ' 'SAsociado TP. 20hrs. ' 'Auxiliar TP. 14hrs. '
 'Auxiliar ']
```

[24]:
```
# Quitar el espacio en blanco al inicio y al final de los valores de la columna␣
 ↪"CATEGORIA"
df_docentes['CATEGORIA'] = df_docentes['CATEGORIA'].str.strip()

# Mostrar los valores únicos de la columna "CATEGORIA" en una lista
print('VALORES ÚNICOS')
print('============')
print(df_docentes['CATEGORIA'].unique())
```

```
VALORES ÚNICOS
============
['Asociado T. Completo' 'Auxiliar TP. 10hrs.' 'Auxiliar T. Completo'
 'Principal T. Completo' 'Principal TP. 20hrs.' 'Auxiliar TP. 20hrs.'
 'Asociado D . E xclusiva.' 'Asociado TP. 20hrs.'
 'Principal D . E xclusiva' 'Auxiliar TP. 04hrs.' 'Principal TP. 16hrs.'
 'Auxiliar D. Exclusiva' 'E Auxiliar TP. 20hrs.' 'Principal'
 'Principal TP. 10hrs.' 'Asociado TP. 10hrs.' 'Auxiliar TP. 15hrs.'
 'Asociado TP. 06hrs.' 'Asociado TP. 08hrs.' 'Auxiliar TP. 06hrs.'
 'Principal TP. 08hrs.' 'Principal TP. 06hrs.' 'Asociado TP. 16hrs.'
 'Auxiliar TP. 08hrs.' 'Asociado TP. 11hrs.' 'Asociado TP. 15hrs.'
 'Auxiliar TP. 13hrs.' 'Auxiliar TP. 16hrs.' 'Asociado TP. 18hrs.'
 'SAsociado TP. 20hrs.' 'Auxiliar TP. 14hrs.' 'Auxiliar']
```

[25]:
```
# Separar la columna "CATEGORIA" en dos columnas: "TIPO TRABAJADOR" y␣
 ↪"MODALIDAD", tomando
# como separador al primer espacio en blanco y eliminando el espacio en blanco␣
 ↪al inicio y al
# final de los valores de las nuevas columnas
df_docentes[['TIPO TRABAJADOR', 'MODALIDAD']] = df_docentes['CATEGORIA'].str.
 ↪split(' ', n=1, expand=True)
df_docentes['TIPO TRABAJADOR'] = df_docentes['TIPO TRABAJADOR'].str.strip()
df_docentes['MODALIDAD'] = df_docentes['MODALIDAD'].str.strip()

# Mostrar los valores únicos de la columna "TIPO TRABAJADOR" en una lista
print('VALORES ÚNICOS DE TIPO TRABAJADOR')
print('============')
print(df_docentes['TIPO TRABAJADOR'].unique())

# Mostrar los valores únicos de la columna "MODALIDAD" en una lista
print('VALORES ÚNICOS DE MODALIDAD')
print('============')
```

```
print(df_docentes['MODALIDAD'].unique())
```

```
VALORES ÚNICOS DE TIPO TRABAJADOR
============
['Asociado' 'Auxiliar' 'Principal' 'E' 'SAsociado']
VALORES ÚNICOS DE MODALIDAD
============
['T. Completo' 'TP. 10hrs.' 'TP. 20hrs.' 'D . E xclusiva.'
 'D . E xclusiva' 'TP. 04hrs.' 'TP. 16hrs.' 'D. Exclusiva'
 'Auxiliar TP. 20hrs.' None 'TP. 15hrs.' 'TP. 06hrs.' 'TP. 08hrs.'
 'TP. 11hrs.' 'TP. 13hrs.' 'TP. 18hrs.' 'TP. 14hrs.']
```

[26]:
```
# Modificar los valores de la columna "TIPO TRABAJADOR" que sean iguales a "E"␣
 ↪por "Auxiliar"
df_docentes.loc[df_docentes['TIPO TRABAJADOR'] == 'E', 'TIPO TRABAJADOR'] =␣
 ↪'Auxiliar'

# Modificar los valores de la columna "MODALIDAD" que sean iguales a "Auxiliar␣
 ↪TP. 20hrs."
# por "TP. 20hrs."
df_docentes.loc[df_docentes['MODALIDAD'] == 'Auxiliar TP. 20hrs.', 'MODALIDAD']␣
 ↪= 'TP. 20hrs.'

# Modificar los valores de la columna "TIPO TRABAJADOR" que sean iguales a "E"␣
 ↪por "Auxiliar"
df_docentes.loc[df_docentes['TIPO TRABAJADOR'] == 'SAsociado', 'TIPO␣
 ↪TRABAJADOR'] = 'Asociado'

# Mostrar los valores únicos de la columna "TIPO TRABAJADOR" en una lista
print('VALORES ÚNICOS DE TIPO TRABAJADOR')
print('============')
print(df_docentes['TIPO TRABAJADOR'].unique())

# Mostrar los valores únicos de la columna "MODALIDAD" en una lista
print('\nVALORES ÚNICOS DE MODALIDAD')
print('============')
print(df_docentes['MODALIDAD'].unique())
```

```
VALORES ÚNICOS DE TIPO TRABAJADOR
============
['Asociado' 'Auxiliar' 'Principal']

VALORES ÚNICOS DE MODALIDAD
============
['T. Completo' 'TP. 10hrs.' 'TP. 20hrs.' 'D . E xclusiva.'
 'D . E xclusiva' 'TP. 04hrs.' 'TP. 16hrs.' 'D. Exclusiva' None
 'TP. 15hrs.' 'TP. 06hrs.' 'TP. 08hrs.' 'TP. 11hrs.' 'TP. 13hrs.'
 'TP. 18hrs.' 'TP. 14hrs.']
```

[27]:
```
# Mostrar las filas que tenga valor None en la columna "MODALIDAD"
print('FILAS CON VALOR NONE EN MODALIDAD')
print('================================')
df_docentes[df_docentes['MODALIDAD'].isnull()]
```

```
FILAS CON VALOR NONE EN MODALIDAD
================================
```

[27]:

	ITEM	CODIGO DOCENTE	APELLIDO PATERNO	APELLIDO MATERNO	NOMBRES	CATEGORIA	TIPO DOC IDENTIDAD	N DOC IDENTIDAD	CORREO INSTITUCIONAL	FACULTAD	TIPO TRABAJADOR
385	386	057142	Ramon	Ruffner De Vega	Jeri Gloria	Principal	DNI	6245729	jramonr@unmsm.edu.pe	Ciencias contables	Principal
1503	1504	010146	Cabrera	Carranza	Carlos Francisco	Principal	DNI	17402784	ccabrerac@unmsm.edu.pe	Ingeniería geológica, minera, metalúrgica y ge…	Principal
2480	2481	08548E	Niño	Montero	Jose Segundo	Principal	DNI	25830033	jninom@unmsm.edu.pe	Medicina	Principal
2963	2964	0A06353	Braul	Moreno	Edgardo Andre	Auxiliar	DNI	70935119	edgardoandrebraul9@gmail.com	Psicología	Auxiliar
2979	2980	0A06354	Escalante	Abanto	Casimiro	Auxiliar	DNI	10583025	cescalantea@unmsm.edu.pe	Psicología	Auxiliar
2992	2993	0A06355	Jochatoma	Roque	Carlos Alberto	Auxiliar	DNI	45245702	carlos_arqueologo@yahoo.com	Psicología	Auxiliar
2996	2997	0A06356	Lozada	Miranda	María Anseli	Auxiliar	DNI	6274169	anseli35@hotmail.com	Psicología	Auxiliar
3043	3044	0A06357	Vargas	Giles	Julia Hortencia	Auxiliar	DNI	7608450	juliavargasgiles1@gmail.com	Psicología	Auxiliar

```
     MODALIDAD
385       None
1503      None
2480      None
2963      None
2979      None
2992      None
2996      None
3043      None
```

[28]:
```
# Modificar el valor None de la columna "MODALIDAD" por "No especificado"
df_docentes['MODALIDAD'].fillna('No especificado', inplace=True)

# Mostrar las filas actualizadas que tengan el valor "No especificado" en la columna "MODALIDAD"
print('FILAS CON VALOR "NO ESPECIFICADO" EN MODALIDAD')
print('==============================================')
df_docentes[df_docentes['MODALIDAD'] == 'No especificado']
```

```
FILAS CON VALOR "NO ESPECIFICADO" EN MODALIDAD
==============================================

C:\Users\carolina\AppData\Local\Temp\ipykernel_17312\2244375154.py:2:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  df_docentes['MODALIDAD'].fillna('No especificado', inplace=True)
```

[28]:
```
      ITEM CODIGO DOCENTE APELLIDO PATERNO APELLIDO MATERNO  \
385    386         057142            Ramon  Ruffner De Vega
1503  1504         010146          Cabrera         Carranza
2480  2481         08548E             Niño          Montero
2963  2964        0A06353            Braul           Moreno
2979  2980        0A06354        Escalante           Abanto
2992  2993        0A06355        Jochatoma            Roque
2996  2997        0A06356           Lozada          Miranda
3043  3044        0A06357           Vargas            Giles

                NOMBRES  CATEGORIA TIPO DOC IDENTIDAD N DOC IDENTIDAD  \
385         Jeri Gloria  Principal                DNI         6245729
```

```
1503   Carlos Francisco  Principal                DNI        17402784
2480       Jose Segundo  Principal                DNI        25830033
2963      Edgardo Andre   Auxiliar                DNI        70935119
2979           Casimiro   Auxiliar                DNI        10583025
2992     Carlos Alberto   Auxiliar                DNI        45245702
2996       María Anseli   Auxiliar                DNI         6274169
3043    Julia Hortencia   Auxiliar                DNI         7608450

              CORREO INSTITUCIONAL  \
385           jramonr@unmsm.edu.pe
1503        ccabrerac@unmsm.edu.pe
2480           jninom@unmsm.edu.pe
2963  edgardoandrebraul9@gmail.com
2979      cescalantea@unmsm.edu.pe
2992   carlos_arqueologo@yahoo.com
2996          anseli35@hotmail.com
3043   juliavargasgiles1@gmail.com

                                                FACULTAD TIPO TRABAJADOR  \
385                                  Ciencias contables       Principal
1503  Ingeniería geológica, minera, metalúrgica y ge…       Principal
2480                                         Medicina       Principal
2963                                      Psicología        Auxiliar
2979                                      Psicología        Auxiliar
2992                                      Psicología        Auxiliar
2996                                      Psicología        Auxiliar
3043                                      Psicología        Auxiliar

            MODALIDAD
385   No especificado
1503  No especificado
2480  No especificado
2963  No especificado
2979  No especificado
2992  No especificado
2996  No especificado
3043  No especificado
```

[29]:
```python
# Eliminar la columna "CATEGORIA" del dataframe
df_docentes.drop(columns=['CATEGORIA'], inplace=True)

# Mostrar el dataframe
print('DATAFRAME ACTUALIZADO')
print('====================')
df_docentes.head(5)
```

```
DATAFRAME ACTUALIZADO
====================
```

[29]:

	ITEM	CODIGO DOCENTE	APELLIDO PATERNO	APELLIDO MATERNO	NOMBRES	\
0	1	0A1018	Aguero	Del Carpio	Lizardo Elias	
1	2	0A7296	Aguirre	Castro	Carmen Judith	
2	3	0A0034	Andrades	Sosa	Jose Ignacio	
3	4	004898	Arias	Mercado	Luis Alberto	
4	5	005452	Aspilcueta	Aspilcueta	Simon Ricardo	

	TIPO DOC IDENTIDAD	N DOC IDENTIDAD	CORREO INSTITUCIONAL	\
0	DNI	7971397	laguerod@unmsm.edu.pe	
1	DNI	23917134	caguirreca@unmsm.edu.pe	
2	DNI	25450694	jandrades@unmsm.edu.pe	
3	DNI	6056404	lariasm@unmsm.edu.pe	
4	DNI	10720986	raspilcuetaa@unmsm.edu.pe	

	FACULTAD	TIPO TRABAJADOR	MODALIDAD
0	Ciencias administrativas	Asociado	T. Completo
1	Ciencias administrativas	Auxiliar	TP. 10hrs.
2	Ciencias administrativas	Asociado	T. Completo
3	Ciencias administrativas	Asociado	T. Completo
4	Ciencias administrativas	Asociado	T. Completo

1.5 Análisis de valores atípicos

[34]:
```
# Mostrar tipos de documentos de identidad
print('TIPOS DE DOCUMENTOS DE IDENTIDAD')
print('================================')
df_docentes['TIPO DOC IDENTIDAD'].unique()
```

```
TIPOS DE DOCUMENTOS DE IDENTIDAD
================================
```

[34]:
```
array(['DNI', 'CE', 'PASS'], dtype=object)
```

- Los DNI (Documento Nacional de Identidad) tienen 8 dígitos
- Los CE (Carnet de Extranjería) tienen 11 dígitos, aunque los documentos emitidos antes de 2000 pueden tener un número distinto
- Los PASS (Pasaporte) tienen 9 dígitos

[37]:
```
# Validar registros erróneos en función del tipo de documento
if df_docentes['TIPO DOC IDENTIDAD'].str.contains('DNI').any():
    registros_erroneos = (
        (df_docentes['N DOC IDENTIDAD'].str.len() != 8) |
        (df_docentes['N DOC IDENTIDAD'].str.contains(r'\D'))
    )
elif df_docentes['TIPO DOC IDENTIDAD'].str.contains('CE').any():
    registros_erroneos = (
        (df_docentes['N DOC IDENTIDAD'].str.len() != 11) |
        (df_docentes['N DOC IDENTIDAD'].str.contains(r'\D'))
```

```
    )
else:
    registros_erroneos = (
        (df_docentes['N DOC IDENTIDAD'].str.len() != 9) |
        (df_docentes['N DOC IDENTIDAD'].str.contains(r'\D'))
    )

# Calcular el porcentaje de registros erróneos
total_registros = len(df_docentes)
porcentaje_erroneos = (registros_erroneos.sum() / total_registros) * 100

print(f"El porcentaje de registros erróneos en la columna 'N DOC IDENTIDAD' es:␣
 ↪{porcentaje_erroneos:.2f}%")
```

```
El porcentaje de registros erróneos en la columna 'N DOC IDENTIDAD' es: 58.38%
```

[43]:
```
# Mostrar los registros erróneos donde el "TIPO DOC IDENTIDEDAD" sea igual a DNI
errores_dni = df_docentes[registros_erroneos & (df_docentes['TIPO DOC␣
 ↪IDENTIDAD'] == 'DNI')]
n_errores_dni = errores_dni.shape[0]

print('REGISTROS ERRONEOS POR DNI - Total de errores: ', n_errores_dni)
print('============================')

errores_dni
```

```
REGISTROS ERRONEOS POR DNI - Total de errores:  1886
============================
```

[43]:

	ITEM	CODIGO DOCENTE	APELLIDO PATERNO	APELLIDO MATERNO	NOMBRES	\
0	1	0A1018	Aguero	Del Carpio	Lizardo Elias	
3	4	004898	Arias	Mercado	Luis Alberto	
5	6	0A7297	Bacigalupo	Pozo	Juan Alberto	
6	7	090018	Barreda	Guerra	Juan Manuel	
7	8	007218	Bautista	Flores	Elena Isabel	
…	…	…	…	…	…	
3241	3242	09868A	Sanchez	Perea	Nofre	
3242	3243	08792E	Sandoval	Chaupe	Nieves Nancy	
3245	3246	087513	Santillan	Altamirano	Gilberto	
3252	3253	0A0244	Vasquez	Cachay	Maria Elith	
3254	3255	074691	Villanueva	Chavez	Cesar Augusto	

	TIPO DOC IDENTIDAD	N DOC IDENTIDAD	CORREO INSTITUCIONAL	\
0	DNI	7971397	laguerod@unmsm.edu.pe	
3	DNI	6056404	lariasm@unmsm.edu.pe	
5	DNI	7623179	jbacigalupop@unmsm.edu.pe	
6	DNI	8236561	jbarredag1@unmsm.edu.pe	
7	DNI	7239532	ebautistaf@unmsm.edu.pe	

```
…                 …              …                          …
3241                DNI         6772234     nsanchezp@unmsm.edu.pe
3242                DNI         7011047    nsandovalc@unmsm.edu.pe
3245                DNI         7182194   gsantillana@unmsm.edu.pe
3252                DNI         9945245     mvasquezc@unmsm.edu.pe
3254                DNI         8142455  cvillanuevac@unmsm.edu.pe

                      FACULTAD TIPO TRABAJADOR        MODALIDAD
0     Ciencias administrativas        Asociado      T. Completo
3     Ciencias administrativas        Asociado      T. Completo
5     Ciencias administrativas        Auxiliar      T. Completo
6     Ciencias administrativas       Principal      T. Completo
7     Ciencias administrativas        Asociado      T. Completo
…                         …              …              …
3241              Veterinaria        Asociado      T. Completo
3242              Veterinaria       Principal   D . E xclusiva
3245              Veterinaria        Asociado      T. Completo
3252              Veterinaria       Principal   D . E xclusiva
3254              Veterinaria        Asociado  D . E xclusiva.

[1886 rows x 11 columns]
```

[44]:
```
# Mostrar los registros erróneos donde el "TIPO DOC IDENTITEDAD" sea igual a CE
errores_ce = df_docentes[registros_erroneos & (df_docentes['TIPO DOC␣
 ↪IDENTIDAD'] == 'CE')]
n_errores_ce = errores_ce.shape[0]

print('REGISTROS ERRONEOS POR CE - Total de errores: ', n_errores_ce)
print('============================')

errores_ce
```

```
REGISTROS ERRONEOS POR CE - Total de errores:  14
============================
```

[44]:
```
      ITEM CODIGO DOCENTE APELLIDO PATERNO APELLIDO MATERNO  \
170    171         0A7538          Britzke              Nan
996    997         0A4237           Borios              Nan
1027  1028         0A5285        De Assis          Clímaco
1044  1045         0A9193         Garduño            Bello
1646  1647         0A6581          Flores        Gutierrez
1767  1768         095753       Lam O Lin              Nan
1857  1858         0A4251       Duponchel                .
1941  1942         0A4115           Oisel              Nan
2971  2972         0A7672            Cruz         Manrique
3114  3115         0A5075          Loroño         Gonzalez
3122  3123         0A6215          Nagles            Vidal
3128  3129         0A6214             Paz            Rojas
```

```
3137  3138         0A4096            Reyes            Yanes
3246  3247         0A4070           Santos            Rueda

                     NOMBRES TIPO DOC IDENTIDAD N DOC IDENTIDAD  \
170                  Ricardo                 CE         1314410
996          Stephanie Carine                CE       000424148
1027                  Danilo                 CE       000134948
1044                  Bianca                 CE       004769587
1646             Jose Ovidio                 CE         2318980
1767   Zhing Fong O Jing Feng                CE          294538
1857        David Jean Robert                CE          845461
1941  Guillaume Yannick Serge                CE         1444366
2971              Yeni Rocio                 CE         5352397
3114          Marcos Antonio                 CE         3595107
3122           Edgar Orlando                 CE          731661
3128               Jose Luis                 CE         5653225
3137       Andreina Alexandra                CE         2832213
3246         Francisco Javier                CE         1102671

              CORREO INSTITUCIONAL                              FACULTAD  \
170           rbritzke@unmsm.edu.pe                   Ciencias biológicas
996            sborios@unmsm.edu.pe                     Ciencias sociales
1027           dassisc@unmsm.edu.pe                     Ciencias sociales
1044         bgardunob@unmsm.edu.pe                     Ciencias sociales
1646          jfloresg@unmsm.edu.pe                  Ingeniería industrial
1767              zlam@unmsm.edu.pe  Ingeniería de Sistemas e Informática
1857        dduponchel@unmsm.edu.pe             Letras y ciencias humanas
1941            goisel@unmsm.edu.pe             Letras y ciencias humanas
2971           ycruzm@unmsm.edu.pe                              Psicología
3114          mloronog@unmsm.edu.pe                                 Química
3122          enaglesv@unmsm.edu.pe                                 Química
3128             jpazr@unmsm.edu.pe                                 Química
3137           areyesy@unmsm.edu.pe                                 Química
3246  francisco.santos@unmsm.edu.pe                             Veterinaria

     TIPO TRABAJADOR       MODALIDAD
170         Auxiliar     T. Completo
996         Auxiliar     T. Completo
1027        Auxiliar     T. Completo
1044        Auxiliar     T. Completo
1646        Auxiliar     T. Completo
1767        Asociado      TP. 20hrs.
1857       Principal  D . E xclusiva
1941        Auxiliar     T. Completo
2971        Auxiliar     T. Completo
3114       Principal     T. Completo
3122       Principal     T. Completo
```

```
3128       Principal  D . E xclusiva
3137        Auxiliar    D. Exclusiva
3246        Auxiliar     T. Completo
```

[45]:
```
# Mostrar los registros erróneos donde el "TIPO DOC IDENTIDEDAD" sea igual a␣
 ↪PASS
errores_pass = df_docentes[registros_erroneos & (df_docentes['TIPO DOC␣
 ↪IDENTIDAD'] == 'PASS')]
n_errores_pass = errores_pass.shape[0]

print('REGISTROS ERRONEOS POR PASS - Total de errores: ', n_errores_pass)
print('============================')

errores_pass
```

```
REGISTROS ERRONEOS POR PASS - Total de errores:  1
============================
```

[45]:
```
      ITEM CODIGO DOCENTE APELLIDO PATERNO APELLIDO MATERNO        NOMBRES  \
1116  1117         0A7604           Romero           Varela  Douglas Yohel

     TIPO DOC IDENTIDAD N DOC IDENTIDAD   CORREO INSTITUCIONAL  \
1116               PASS         E943169  dromerov@unmsm.edu.pe

               FACULTAD TIPO TRABAJADOR   MODALIDAD
1116  Ciencias sociales        Auxiliar  TP. 10hrs.
```

Dado que los errores por N DOC IDENTIDAD superan el 50% y que esta no es la variable target, no se eliminarán estos registros.

[46]:
```
# Mostrar tipos de datos del dataframe final
print('TIPOS DE DATOS')
print('==============')
df_docentes.info()
```

```
TIPOS DE DATOS
==============
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3256 entries, 0 to 3255
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ITEM                  3256 non-null   int64
 1   CODIGO DOCENTE        3256 non-null   object
 2   APELLIDO PATERNO      3256 non-null   object
 3   APELLIDO MATERNO      3256 non-null   object
 4   NOMBRES               3256 non-null   object
 5   TIPO DOC IDENTIDAD    3256 non-null   object
 6   N DOC IDENTIDAD       3256 non-null   object
```

```
 7   CORREO INSTITUCIONAL  3256 non-null   object
 8   FACULTAD              3256 non-null   object
 9   TIPO TRABAJADOR       3256 non-null   object
 10  MODALIDAD             3256 non-null   object
dtypes: int64(1), object(10)
memory usage: 279.9+ KB
```

[47]:
```
# Convertir los tipos de datos
df_docentes['CODIGO DOCENTE'] = df_docentes['CODIGO DOCENTE'].astype(str)
df_docentes['APELLIDO PATERNO'] = df_docentes['APELLIDO PATERNO'].astype(str)
df_docentes['APELLIDO MATERNO'] = df_docentes['APELLIDO MATERNO'].astype(str)
df_docentes['NOMBRES'] = df_docentes['NOMBRES'].astype(str)
df_docentes['TIPO DOC IDENTIDAD'] = df_docentes['TIPO DOC IDENTIDAD'].
 ↪astype(str)
df_docentes['N DOC IDENTIDAD'] = df_docentes['N DOC IDENTIDAD'].astype(str)
df_docentes['CORREO INSTITUCIONAL'] = df_docentes['CORREO INSTITUCIONAL'].
 ↪astype(str)
df_docentes['FACULTAD'] = df_docentes['FACULTAD'].astype(str)
df_docentes['TIPO TRABAJADOR'] = df_docentes['TIPO TRABAJADOR'].astype(str)
df_docentes['MODALIDAD'] = df_docentes['MODALIDAD'].astype(str)

# Mostrar los tipos de datos
print('TIPOS DE DATOS')
print('==============')
for col in df_docentes.columns:
    print(f'{col}: {df_docentes[col].apply(type).unique()}')
```

```
TIPOS DE DATOS
==============
ITEM: [<class 'int'>]
CODIGO DOCENTE: [<class 'str'>]
APELLIDO PATERNO: [<class 'str'>]
APELLIDO MATERNO: [<class 'str'>]
NOMBRES: [<class 'str'>]
TIPO DOC IDENTIDAD: [<class 'str'>]
N DOC IDENTIDAD: [<class 'str'>]
CORREO INSTITUCIONAL: [<class 'str'>]
FACULTAD: [<class 'str'>]
TIPO TRABAJADOR: [<class 'str'>]
MODALIDAD: [<class 'str'>]
```

1.6 Balanceo del dataset

[49]:
```
# Exportar el dataframe a un archivo CSV
df_docentes.to_csv('Padron_docentes_limpio_y_normalizado.csv', sep=';',␣
 ↪index=False, encoding='UTF-8')
```

[51]:
```
from imblearn.under_sampling import RandomUnderSampler
from sklearn.utils import shuffle, resample

# Cargar el nuevo dataset en un dataframe
df_docentes_procesado = pd.read_csv('Padron_docentes_limpio_y_normalizado.csv',␣
 ↪sep=';', encoding='UTF-8')

df_docentes_procesado.head(5)
```

[51]:

	ITEM	CODIGO DOCENTE	APELLIDO PATERNO	APELLIDO MATERNO	NOMBRES
0	1	0A1018	Aguero	Del Carpio	Lizardo Elias
1	2	0A7296	Aguirre	Castro	Carmen Judith
2	3	0A0034	Andrades	Sosa	Jose Ignacio
3	4	004898	Arias	Mercado	Luis Alberto
4	5	005452	Aspilcueta	Aspilcueta	Simon Ricardo

	TIPO DOC IDENTIDAD	N DOC IDENTIDAD	CORREO INSTITUCIONAL
0	DNI	7971397	laguerod@unmsm.edu.pe
1	DNI	23917134	caguirreca@unmsm.edu.pe
2	DNI	25450694	jandrades@unmsm.edu.pe
3	DNI	6056404	lariasm@unmsm.edu.pe
4	DNI	10720986	raspilcuetaa@unmsm.edu.pe

	FACULTAD	TIPO TRABAJADOR	MODALIDAD
0	Ciencias administrativas	Asociado	T. Completo
1	Ciencias administrativas	Auxiliar	TP. 10hrs.
2	Ciencias administrativas	Asociado	T. Completo
3	Ciencias administrativas	Asociado	T. Completo
4	Ciencias administrativas	Asociado	T. Completo

Variable target: TIPO TRABAJADOR

[53]:
```
# Verificar el  balance de clases
print(df_docentes_procesado['TIPO TRABAJADOR'].value_counts())
```

```
TIPO TRABAJADOR
Auxiliar     1235
Asociado     1225
Principal     796
Name: count, dtype: int64
```

[54]:
```
# Separamos las características (X) de la variable objetivo (y)
X = df_docentes_procesado.drop('TIPO TRABAJADOR', axis=1)
Y = df_docentes_procesado['TIPO TRABAJADOR']
```

[55]:
```
# Configurar RandomUnderSampler para reducir las clases mayoritarias al tamaño␣
 ↪de la minoritaria
rus = RandomUnderSampler(random_state=42)
```

```
X_res, y_res = rus.fit_resample(X, Y)
```

[56]:
```
# Reconstruir el DataFrame balanceado
df_balanceado = pd.concat([X_res, y_res], axis=1)
```

[57]:
```
# Verificar el nuevo balance de clases
print(df_balanceado['TIPO TRABAJADOR'].value_counts())
```

```
TIPO TRABAJADOR
Asociado     796
Auxiliar     796
Principal    796
Name: count, dtype: int64
```

[59]:
```
# Exportar el dataframe a un archivo CSV
df_balanceado.to_csv('Padron_docentes_balanceado.csv', sep=';', index=False,␣
↪encoding='UTF-8')
```