Análisis de un Dataset

Notas del Dataset

Significado de cada atributo

1. **Marital Status:** The marital status of the student. (Categorical)
2. **Application Mode:** The method of application use by the student. (Categorical)
3. **Application Order:** The order in which the student applied. (Numerical)
4. **Course:** The course taken by the student. (Categorical)
5. **Daytime/evening attendance:** Whether the student attends classes during the day or in the evening. (Categorical)
6. **Previous qualification:** The qualification obtained by the student before enrolling in higher education. (Categorical)
7. **Nacionality:** The nationality of the student. (Categorical)
8. **Mother's qualification:** The qualification of the student's mother. (Categorical)
9. **Father's qualification:** The qualification of the student's father. (Categorical)
10. **Mother's occupation:** The occupation of the student's mother. (Categorical)
11. **Father's occupation:** The occupation of the student's father. (Categorical)
12. **Displaced:** Whether the student is a displaced person. (Categorical)
13. **Educational special needs:** Whether the student has any special educational needs. (Categorical)
14. **Debtor:** Whether the student is a debtor. (Categorical)
15. **Tuition fees up to date:** Whether the student's tuition fees are up to date. (Categorical)
16. **Gender:** The gender of the student. (Categorical)
17. **Scholarship holder:** Whether the student is a scholarship holder. (Categorical)
18. **Age at enrollment:** The age of the student at the time of enrollment. (Numerical)
19. **International:** Whether the student is an international student. (Categorical)
20. **Curricular units 1st sem (credited):** The number of curricular units credited by the student in the first semester. (Numerical)
21. **Curricular units 1st sem (enrolled):** The number of curricular units enrolled by the student in the first semester. (Numerical)
22. **Curricular units 1st sem (evaluations):** The number of curricular units evaluated by the student in the first semester. (Numerical)
23. **Curricular units 1st sem (approved):** The number of curricular units approved by the student in the first semester. (Numerical)
24. **Curricular units 1st sem (grade):**
25. **Curricular units 1st sem(without evaluations):**

Otros:

34. **GDP:** Producto Interno Bruto
35. **Target:** Estado del estudiante

NOTA: Los datos macroeconómicos, socioeconómicos y demográficos fueron capturados al momento de la inscripción del alumno. Los periodos académicos van desde el 2008/2009 hasta 2018/2019 tomando en cuenta estudiantes de diferentes áreas como agronomía, diseño, educación, enfermería, periodismo, tecnología, etc. El país de los alumnos es Portugal cuyo sistema de calificaciones se muestra a continuación:

1. 0-9.9: suspenso

2. 10-13.9: aprobado

3. 14-16.9: notable

4. 17-18.9: sobresaliente

5. 19-20: Matrícula de Honor

Codificación de cada atributo

Marital Status

1. Soltero
2. Casado
3. Viudo
4. Divorciado
5. Unión Libre
6. Separado Legalmente

Nacionalidad

1. Portugués
2. Alemán
3. Español
4. Italiano
5. Holandés
6. Inglés
7. Lituano
8. Angoleño
9. Cabo Verde
10. Guinea
11. Mozambiqueño
12. Santomean

13. Turco
14. Brasileño
15. Rumano
16. República de Moldova
17. Mexicano
18. Ucraniano
19. Ruso
20. Cubano
21. Colombiano

Método de Aplicación

1. 1st phase—general contingent
2. Ordinance No. 612/93
3. 1st phase—special contingent (Azores Island)
4. Holders of other higher courses
5. Ordinance No. 854-B/99
6. International student (bachelor)
7. 1st phase—special contingent (Madeira Island)
8. 2nd phase—general contingent
9. 3rd phase—general contingent
10. Ordinance No. 533-A/99, item b2) (Different Plan)
11. Ordinance No. 533-A/99, item b3 (Other Institution)
12. Over 23 years old
13. Transfer
14. Change in course
15. Technological specialization diploma holders
16. Change in institution/course
17. Short cycle diploma holders
18. Change in institution/course (International)

Curso

1. Biofuel Production Technologies
2. Animation and Multimedia Design
3. Social Service (evening attendance)
4. Agronomy
5. Communication Design
6. Veterinary Nursing
7. Informatics Engineering
8. Equiniculture
9. Management
10. Social Service

11. Tourism
12. Nursing
13. Oral Hygiene
14. Advertising and Marketing Management
15. Journalism and Communication
16. Basic Education
17. Management (evening attendance)

Grado de Estudios Previo

1. Secondary education
2. Higher education—bachelor’s degree
3. Higher education—degree
4. Higher education—master’s degree
5. Higher education—doctorate
6. Frequency of higher education
7. 12th year of schooling—not completed
8. 11th year of schooling—not completed
9. Other—11th year of schooling
10. 10th year of schooling
11. 10th year of schooling—not completed
12. Basic education 3rd cycle (9th/10th/11th year) or equivalent
13. Basic education 2nd cycle (6th/7th/8th year) or equivalent
14. Technological specialization course
15. Higher education—degree (1st cycle)
16. Professional higher technical course
17. Higher education—master’s degree (2nd cycle)

Grado de Estudios Padre y Madre

1. Secondary Education—12th Year of Schooling or Equivalent
2. Higher Education—bachelor’s degree
3. Higher Education—degree
4. Higher Education—master’s degree
5. Higher Education—doctorate
6. Frequency of Higher Education
7. 12th Year of Schooling—not completed
8. 11th Year of Schooling—not completed
9. 7th Year (Old)
10. Other—11th Year of Schooling
11. 2nd year complementary high school course
12. 10th Year of Schooling
13. General commerce course

14. Basic Education 3rd Cycle (9th/10th/11th Year) or Equivalent
15. Complementary High School Course
16. Technical-professional course
17. Complementary High School Course—not concluded
18. 7th year of schooling
19. 2nd cycle of the general high school course
20. 9th Year of Schooling—not completed
21. 8th year of schooling
22. General Course of Administration and Commerce
23. Supplementary Accounting and Administration
24. Unknown
25. Cannot read or write
26. Can read without having a 4th year of schooling
27. Basic education 1st cycle (4th/5th year) or equivalent
28. Basic Education 2nd Cycle (6th/7th/8th Year) or equivalent
29. Technological specialization course
30. Higher education—degree (1st cycle)
31. Specialized higher studies course
32. Professional higher technical course
33. Higher Education—master’s degree (2nd cycle)

Ocupación de Padre y Madre

1. Student
2. Representatives of the Legislative Power and Executive Bodies,

Directors, Directors and Executive Managers 3. Specialists in Intellectual and Scientific Activities 4. Intermediate Level Technicians and Professions 5. Administrative staff 6. Personal Services, Security and SafetyWorkers, and Sellers 7. Farmers and Skilled Workers in Agriculture, Fisheries, and Forestry 8. Skilled Workers in Industry, Construction, and Craftsmen 9. Installation and Machine Operators and Assembly Workers 10. Unskilled Workers 11. Armed Forces Professions 12. Other Situation; 13—(blank) 14. Armed Forces Officers 15. Armed Forces Sergeants 16. Other Armed Forces personnel 17. Directors of administrative and commercial services 18. Hotel, catering, trade, and other services directors 19. Specialists in the physical sciences, mathematics, engineering, and related techniques 20. Health professionals 21. Teachers 22. Specialists in finance, accounting, administrative organization, and public and commercial relations 23. Intermediate level science and engineering technicians and professions 24. Technicians and professionals of intermediate level of health 25. Intermediate level technicians from legal, social, sports, cultural, and similar services 26. Information and communication technology technicians 27. Office workers, secretaries in general, and data processing operators 28. Data, accounting, statistical, financial services, and registry-related operators 29. Other administrative support staff 30. Personal service workers 31. Sellers 32. Personal care workers and the like 33. Protection and security services personnel 34. Market-oriented farmers and skilled agricultural and animal production workers 35. Farmers, livestock keepers, fishermen, hunters and gatherers, and subsistence 36. Skilled construction workers and the like, except electricians 37. Skilled workers in metallurgy, metalworking, and similar 38. Skilled workers in electricity and electronics 39. Workers in food processing, woodworking, and clothing and other industries and crafts 40. Fixed plant and machine operators 41. Assembly workers 42. Vehicle drivers and mobile equipment operators 43. Unskilled workers in agriculture, animal production, and fisheries and forestry 44. Unskilled workers in extractive industry, construction, manufacturing, and transport 45. Meal preparation assistants 46. Street vendors (except food) and street service providers

Género

0. Femenino
1. Masculino

Asistencia Matutina/Vespertina

0. Vespertino
1. Matutino

Atributos con valores SÍ/NO

Valores:

0. NO
1. SÍ

Atributos:

1. Displaced
2. Educational Special Needs
3. Debtor
4. Tuitions fees up to date
5. Scholarship holder
6. International

Análisis

In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
```

Leer Dataset

```
In [ ]: df = pd.read_csv('Dataset_Students_Success\dataset.csv')
        df.head(5)
```

Out[ ]:

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Nacionality	Mother's qualificatior
0	1	8	5	2	1	1	1	13
1	1	6	1	11	1	1	1	1
2	1	1	5	5	1	1	1	22
3	1	8	2	15	1	1	1	23
4	2	12	1	3	0	1	1	22

5 rows × 35 columns

Número de Atributos e Instancias

```
In [ ]: inst, atr = df.shape

        print('El número de instancias es: ', inst)
        print('El número de atributos es: ', atr)
```

```
El número de instancias es:  4424
El número de atributos es:  35
```

Número de Atributos con Datos Faltantes

```
In [ ]: datos_faltantes = df.isnull().sum()
        col_dat_falt = datos_faltantes[datos_faltantes > 0]

        # print(datos_faltantes)
        if col_dat_falt.empty:
            print('No hay datos faltantes')
        else:
            print('Las columnas con datos faltantes son:  \n')
            print(col_dat_falt)
```

```
No hay datos faltantes
```

Mínimo, Máximo, Media y Desviación Estándar de cada atributo

```
In [ ]: df_numeric = df.iloc[:,:-1] # Crear un nuevo dataframe sin la
                                    # última columna que es categórica
        medidas = {
            "min": df_numeric.min(),
            "max": df_numeric.max(),
            "media": df_numeric.mean(),
            "Desviación Estándar": df_numeric.std()
        }
        medidas = pd.DataFrame(medidas)
```

```
print(len(medidas))
medidas
```

```
34
```

Out[ ]:

	min	max	media	Desviación Estándar
Marital status	1.00	6.000000	1.178571	0.605747
Application mode	1.00	18.000000	6.886980	5.298964
Application order	0.00	9.000000	1.727848	1.313793
Course	1.00	17.000000	9.899186	4.331792
Daytime/evening attendance	0.00	1.000000	0.890823	0.311897
Previous qualification	1.00	17.000000	2.531420	3.963707
Nacionality	1.00	21.000000	1.254521	1.748447
Mother's qualification	1.00	29.000000	12.322107	9.026251
Father's qualification	1.00	34.000000	16.455244	11.044800
Mother's occupation	1.00	32.000000	7.317812	3.997828
Father's occupation	1.00	46.000000	7.819168	4.856692
Displaced	0.00	1.000000	0.548373	0.497711
Educational special needs	0.00	1.000000	0.011528	0.106760
Debtor	0.00	1.000000	0.113698	0.317480
Tuition fees up to date	0.00	1.000000	0.880651	0.324235
Gender	0.00	1.000000	0.351718	0.477560
Scholarship holder	0.00	1.000000	0.248418	0.432144
Age at enrollment	17.00	70.000000	23.265145	7.587816
International	0.00	1.000000	0.024864	0.155729
Curricular units 1st sem (credited)	0.00	20.000000	0.709991	2.360507
Curricular units 1st sem (enrolled)	0.00	26.000000	6.270570	2.480178
Curricular units 1st sem (evaluations)	0.00	45.000000	8.299051	4.179106
Curricular units 1st sem (approved)	0.00	26.000000	4.706600	3.094238
Curricular units 1st sem (grade)	0.00	18.875000	10.640822	4.843663
Curricular units 1st sem (without evaluations)	0.00	12.000000	0.137658	0.690880
Curricular units 2nd sem (credited)	0.00	19.000000	0.541817	1.918546
Curricular units 2nd sem (enrolled)	0.00	23.000000	6.232143	2.195951
Curricular units 2nd sem (evaluations)	0.00	33.000000	8.063291	3.947951
Curricular units 2nd sem (approved)	0.00	20.000000	4.435805	3.014764
Curricular units 2nd sem (grade)	0.00	18.571429	10.230206	5.210808
Curricular units 2nd sem (without evaluations)	0.00	12.000000	0.150316	0.753774
Unemployment rate	7.60	16.200000	11.566139	2.663850
Inflation rate	-0.80	3.700000	1.228029	1.382711

	min	max	media	Desviación Estándar
GDP	-4.06	3.510000	0.001969	2.269935

Dataframe de Correlación de Pearson

In [ ]:

```
df_pearson = df_numeric.corr()

df_pearson
```

Out[ ]:

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Naci
Marital status	1.000000	0.224855	-0.125854	0.018925	-0.274939	0.120925	-0.0
Application mode	0.224855	1.000000	-0.246497	-0.085116	-0.268616	0.433028	-0.0
Application order	-0.125854	-0.246497	1.000000	0.118928	0.158657	-0.199029	-0.0
Course	0.018925	-0.085116	0.118928	1.000000	-0.070232	-0.158382	-0.0
Daytime/evening attendance	-0.274939	-0.268616	0.158657	-0.070232	1.000000	-0.103022	0.0
Previous qualification	0.120925	0.433028	-0.199029	-0.158382	-0.103022	1.000000	-0.0
Nacionality	-0.020722	-0.001360	-0.029385	-0.004761	0.024433	-0.038997	1.0
Mother's qualification	0.185522	0.092867	-0.061719	0.058909	-0.195346	0.018868	-0.0
Father's qualification	0.128326	0.072798	-0.049936	0.045659	-0.137769	0.013152	-0.0
Mother's occupation	0.069734	0.033489	-0.046591	0.029672	-0.037986	0.006190	0.0
Father's occupation	0.024351	0.001253	-0.029754	0.016489	0.000845	0.005381	0.0
Displaced	-0.234886	-0.263079	0.332362	0.006142	0.251767	-0.149356	-0.0
Educational special needs	-0.028343	-0.030868	0.025597	-0.001886	0.031017	-0.015015	-0.0
Debtor	0.034304	0.114348	-0.072151	-0.053149	0.006658	0.117447	0.0
Tuition fees up to date	-0.087158	-0.127339	0.055891	0.029099	0.038799	-0.095246	-0.0
Gender	-0.014738	0.147226	-0.089559	-0.111383	-0.012326	0.089952	-0.0
Scholarship holder	-0.053765	-0.152818	0.073709	0.051668	0.093912	-0.085668	-0.0
Age at enrollment	0.522717	0.450700	-0.271154	-0.036929	-0.462280	0.249821	-0.0
International	-0.027905	0.005050	-0.028801	-0.004662	0.027973	-0.033498	0.9
Curricular units 1st sem (credited)	0.061209	0.238269	-0.133354	-0.140546	-0.127466	0.159940	0.0
Curricular units 1st sem (enrolled)	0.052107	0.159547	-0.016808	0.112285	-0.043056	0.080860	-0.0
Curricular units 1st sem	0.058030	0.219154	-0.092156	0.025970	-0.045889	0.129364	0.0

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Naci
(evaluations)							
Curricular units 1st sem (approved)	-0.031027	-0.023713	0.035580	0.077038	0.016935	-0.005295	0.(
Curricular units 1st sem (grade)	-0.059811	-0.106213	0.058308	0.179482	0.063974	-0.034252	0.(
Curricular units 1st sem (without evaluations)	0.034711	0.040255	-0.031699	-0.060483	0.045630	0.018276	0.(
Curricular units 2nd sem (credited)	0.062831	0.228973	-0.125815	-0.120390	-0.111953	0.138463	-0.(
Curricular units 2nd sem (enrolled)	0.039026	0.127461	0.028878	0.185879	0.000371	0.056450	-0.(
Curricular units 2nd sem (evaluations)	0.022784	0.164992	-0.055089	0.049236	0.014610	0.101501	-0.(
Curricular units 2nd sem (approved)	-0.043739	-0.065203	0.071793	0.120000	0.034022	-0.037265	-0.(
Curricular units 2nd sem (grade)	-0.071506	-0.104424	0.055517	0.178997	0.050493	-0.038765	-0.(
Curricular units 2nd sem (without evaluations)	0.020426	0.042009	-0.015757	-0.013984	-0.004229	0.024186	-0.(
Unemployment rate	-0.020338	0.091567	-0.098419	-0.050116	0.061974	0.096914	-0.(
Inflation rate	0.008761	-0.019613	-0.011133	0.028775	-0.024043	-0.056388	-0.(
GDP	-0.027003	-0.014563	0.030201	-0.012518	0.022929	0.053968	0.(

34 rows × 34 columns

Calcular Valores Atípicos (outliers)

```
In [ ]: dicc_len_outliers = {}

        for column in df_numeric.columns:
            q1, q3 = df_numeric[column].quantile([0.25, 0.75]) # Calcular primer y tercer cuar
            iqr = q3 - q1 # Calcular Rango Intercuartil

            # Filtrar valores
            outliers = df_numeric[(df_numeric[column] > q3 + 1.5*iqr) | (df_numeric[column] <
            # Agregar al diccionario
            dicc_len_outliers[column] = len(outliers)

        print("OUTLIERS PARA CADA ATRIBUTO:", '\n')
        print(dicc_len_outliers)
```

```
OUTLIERS PARA CADA ATRIBUTO:

{'Marital status': 505, 'Application mode': 0, 'Application order': 541, 'Course': 0,
'Daytime/evening attendance': 483, 'Previous qualification': 707, 'Nacionality': 110,
"Mother's qualification": 0, "Father's qualification": 0, "Mother's occupation": 84,
"Father's occupation": 84, 'Displaced': 0, 'Educational special needs': 51, 'Debtor':
503, 'Tuition fees up to date': 528, 'Gender': 0, 'Scholarship holder': 1099, 'Age at
enrollment': 441, 'International': 110, 'Curricular units 1st sem (credited)': 577,
'Curricular units 1st sem (enrolled)': 424, 'Curricular units 1st sem (evaluations)':
158, 'Curricular units 1st sem (approved)': 180, 'Curricular units 1st sem (grade)':
726, 'Curricular units 1st sem (without evaluations)': 294, 'Curricular units 2nd sem
(credited)': 530, 'Curricular units 2nd sem (enrolled)': 369, 'Curricular units 2nd s
em (evaluations)': 109, 'Curricular units 2nd sem (approved)': 44, 'Curricular units
2nd sem (grade)': 877, 'Curricular units 2nd sem (without evaluations)': 282, 'Unempl
oyment rate': 0, 'Inflation rate': 0, 'GDP': 0}
```

Graficación con Matplotlib

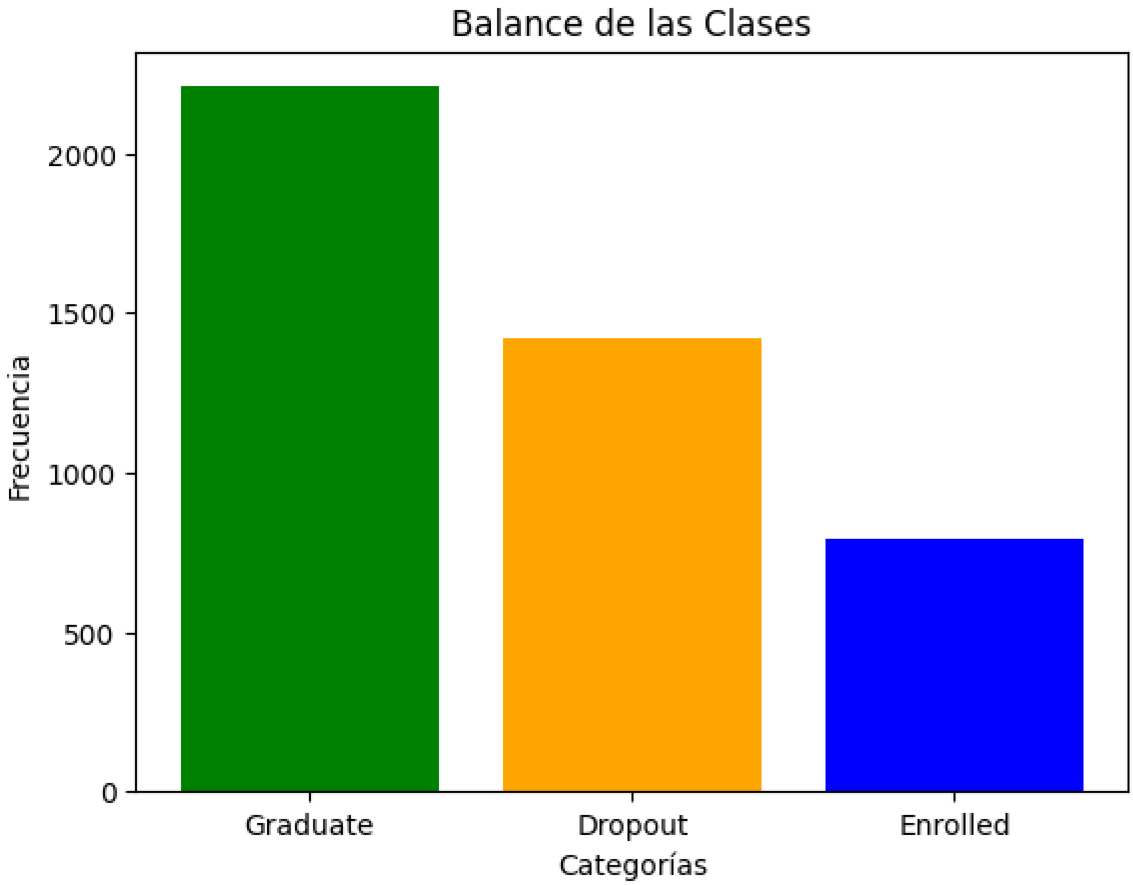
1. Gráfico de barras para visualizar el balance de las clases de acuerdo a la última columna 'Target'

In [ ]:

```
colores = {"Graduate": 'green', "Dropout": 'orange', "Enrolled": 'blue'}
# Contar cantidad de veces en las que aparece cada clase
frecuencias = df['Target'].value_counts()

# Crear gráfica
plt.bar(frecuencias.index, frecuencias.values, color=[colores.get(x) for x in frecuenc
plt.xlabel('Categorías')
plt.ylabel('Frecuencia')
plt.title('Balance de las Clases')
```

Out[ ]:

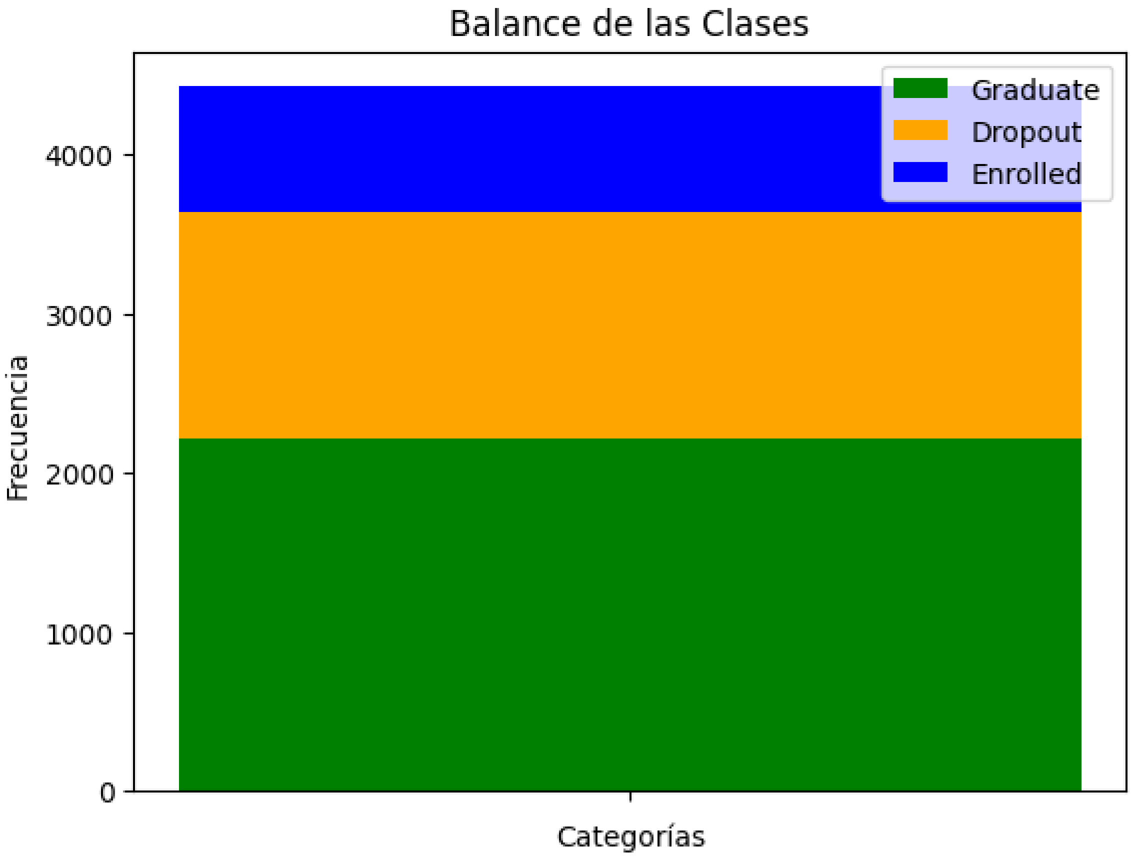
```
Text(0.5, 1.0, 'Balance de las Clases')
```



2. Gráfica de Barras Apiladas

```
In [ ]: plt.bar(frecuencias.index[0], frecuencias[0], color='g', label=frecuencias.index[0])
        plt.bar(frecuencias.index[0], frecuencias[1], color = 'orange', bottom=frecuencias[0],
        plt.bar(frecuencias.index[0], frecuencias[2], color = 'blue', bottom=frecuencias[0]+fr
        plt.xlabel('Categorías')
        plt.ylabel('Frecuencia')
        plt.title('Balance de las Clases')
        plt.legend()
        plt.gca().set_xticklabels([]) # Quitar etiquetas de debajo del gráfico
```

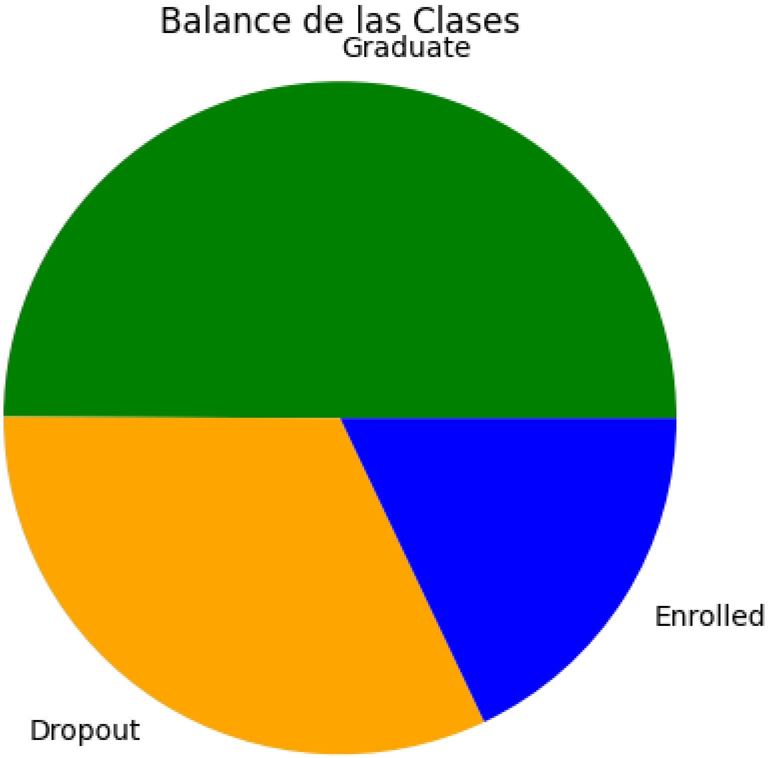
```
Out[ ]: [Text(0, 0, '')]
```



3. Gráfico de Pastel

In [ ]:
```
plt.pie(frecuencias.values, labels=frecuencias.index, colors=[colores.get(x) for x in
plt.title('Balance de las Clases')
plt.axis('equal')
```

Out[ ]:
```
(-1.0999999999211363,
 1.0999999999962446,
 -1.0999998347850697,
 1.099999992095447)
```



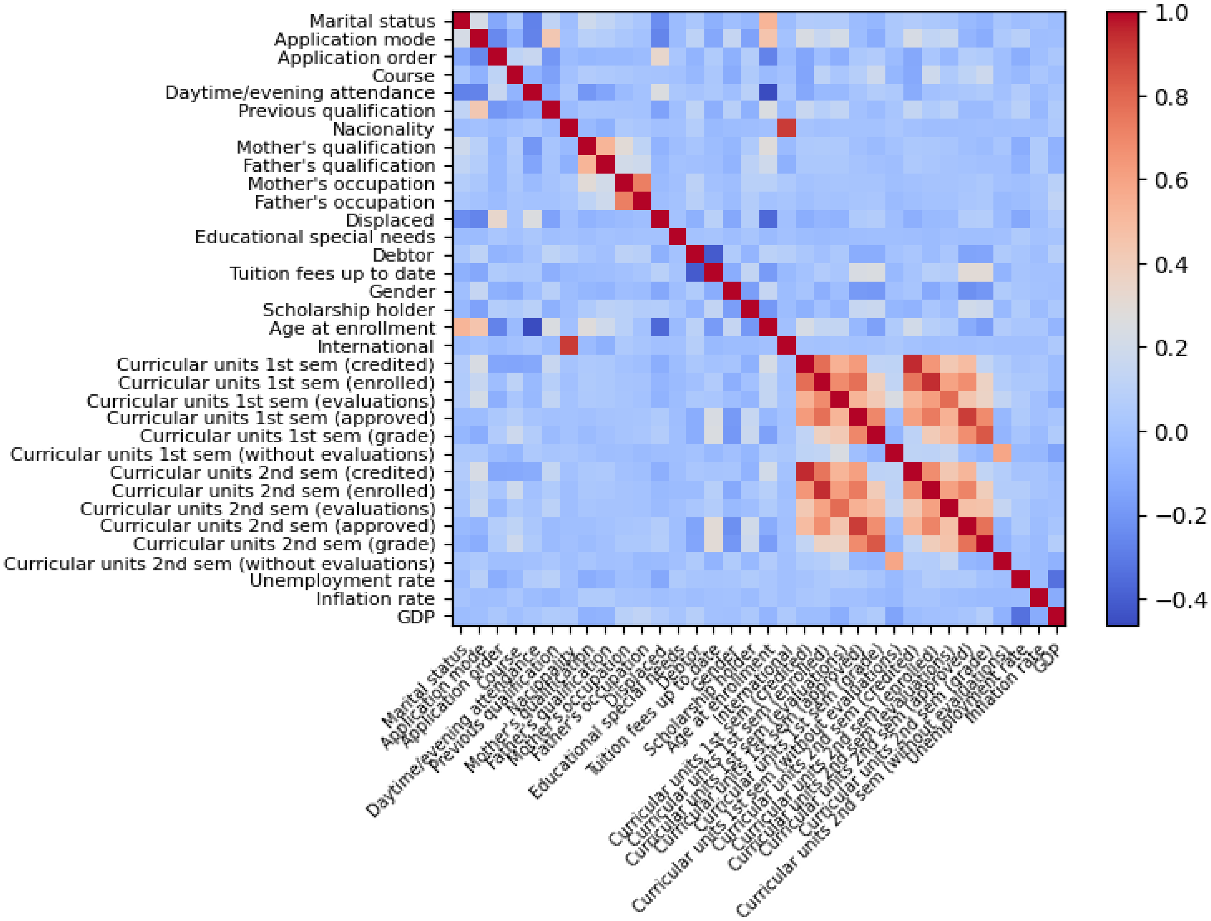
4. Mapa de calor (en base a la correlación de Pearson)

In [ ]:

```
figure = plt.figure()
ax = plt.subplot()
im = ax.imshow(df_pearson, cmap='coolwarm')

# Añadir etiquetas a los ejes y límites
ax.set_xticks(range(len(df_pearson)))
ax.set_yticks(range(len(df_pearson)))
ax.set_xticklabels(df_pearson.columns, fontsize=7)
ax.set_yticklabels(df_pearson.columns, fontsize=8)

# Rotar las etiquetas del eje X para que quepan en el gráfico
plt.setp(ax.get_xticklabels(), rotation=45, ha='right', rotation_mode='anchor')

# Añadir una barra de colores para mostrar la intensidad de la correlación
figure.colorbar(im)
```

Out[ ]:

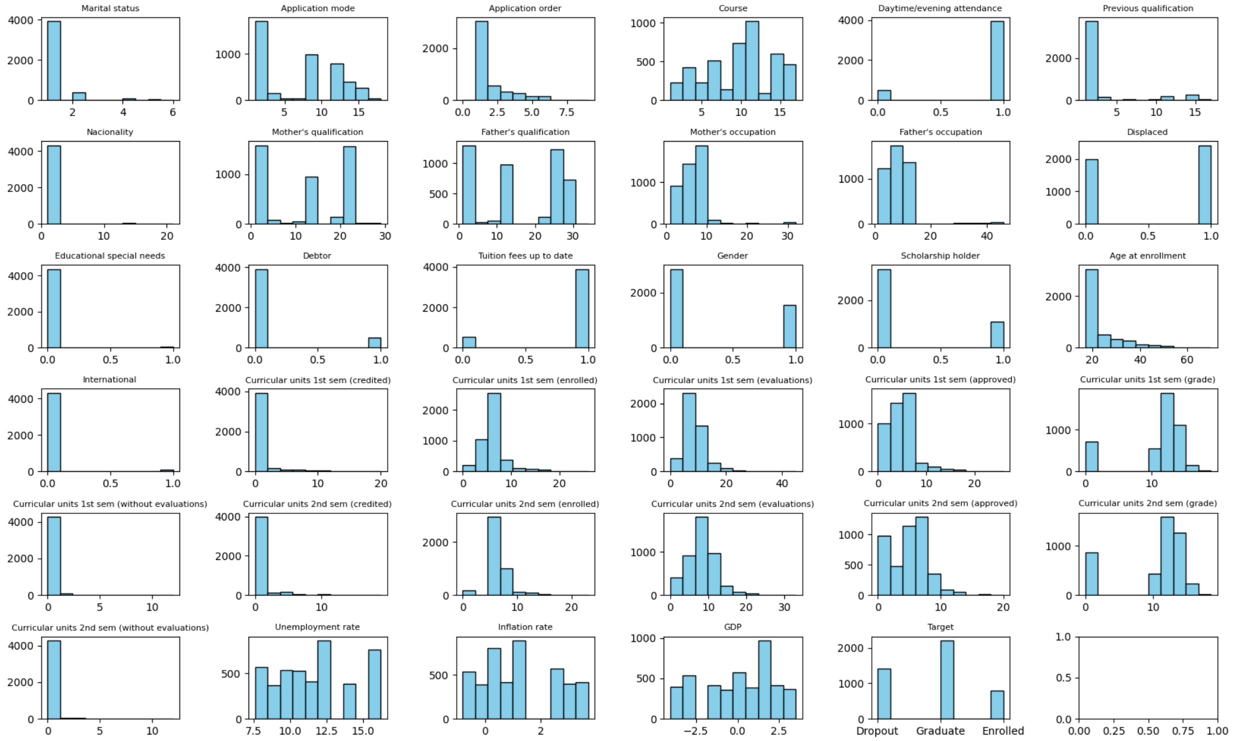
```
<matplotlib.colorbar.Colorbar at 0x2001fa5c700>
```



5. Histograma para visualizar la distribución de los datos

In [ ]:

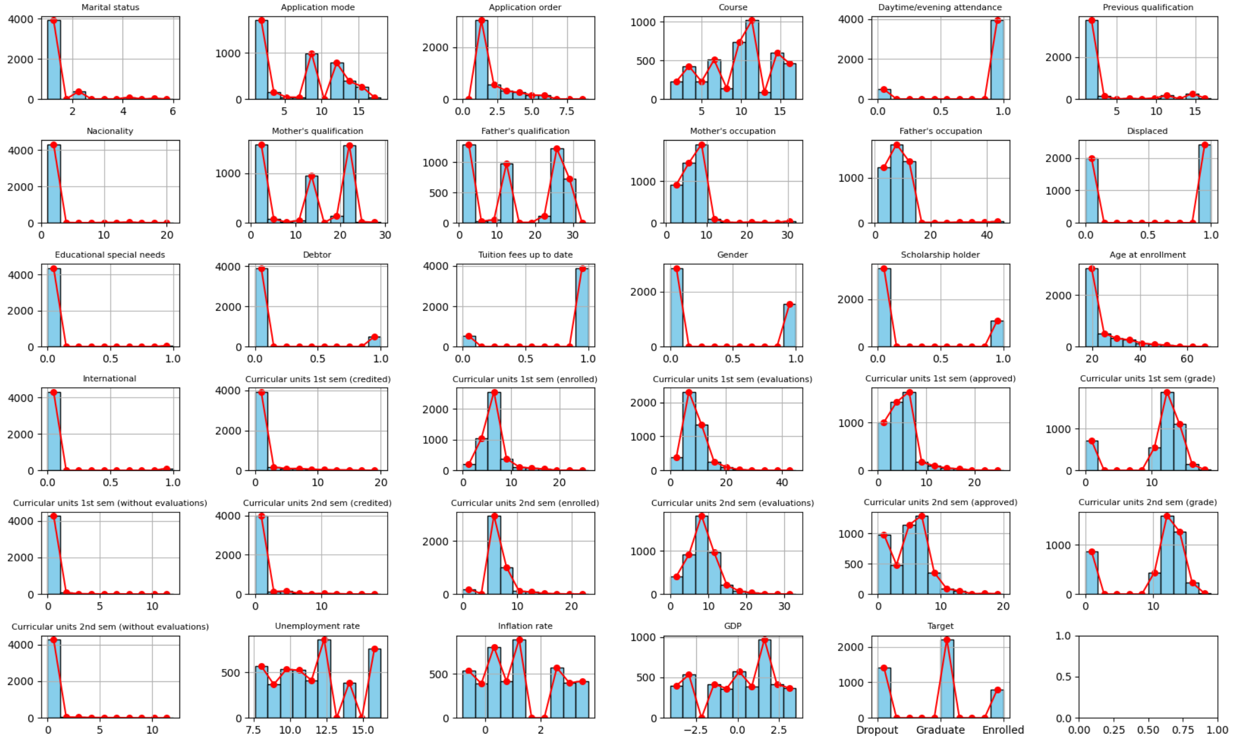
```
# Crear histograma
fig, axs = plt.subplots(nrows=6, ncols=6, figsize=(20, 12))
fig.subplots_adjust(hspace=0.5, wspace=0.5)

# Crear histogramas para cada columna
for i, col in enumerate(df.columns):
    ax = axs[i//6, i%6] # seleccionar el subplot correspondiente
    ax.hist(df[col], bins=10, color='skyblue', edgecolor='black')
    ax.set_title(col, fontsize=8)
```



6. Histograma con Líneas

In [ ]:

```
fig, axs = plt.subplots(nrows=6, ncols=6, figsize=(20, 12))
fig.subplots_adjust(hspace=0.5, wspace=0.5)

for i, col in enumerate(df.columns):
    ax = axs[i//6, i%6]
    n, bins, patches = ax.hist(df[col], bins=10, color='skyblue', edgecolor='black')
    x = bins[:-1] + (bins[1] - bins[0]) / 2  # calcular el centro de cada barra
    y = n  # usar la frecuencia como altura del punto
    ax.scatter(x, y, c='red', marker='o', s=30)  # agregar los puntos
    ax.plot(x, y, c='red')  # unir los puntos con una línea
    ax.set_title(col, fontsize=8)
    ax.grid()
```

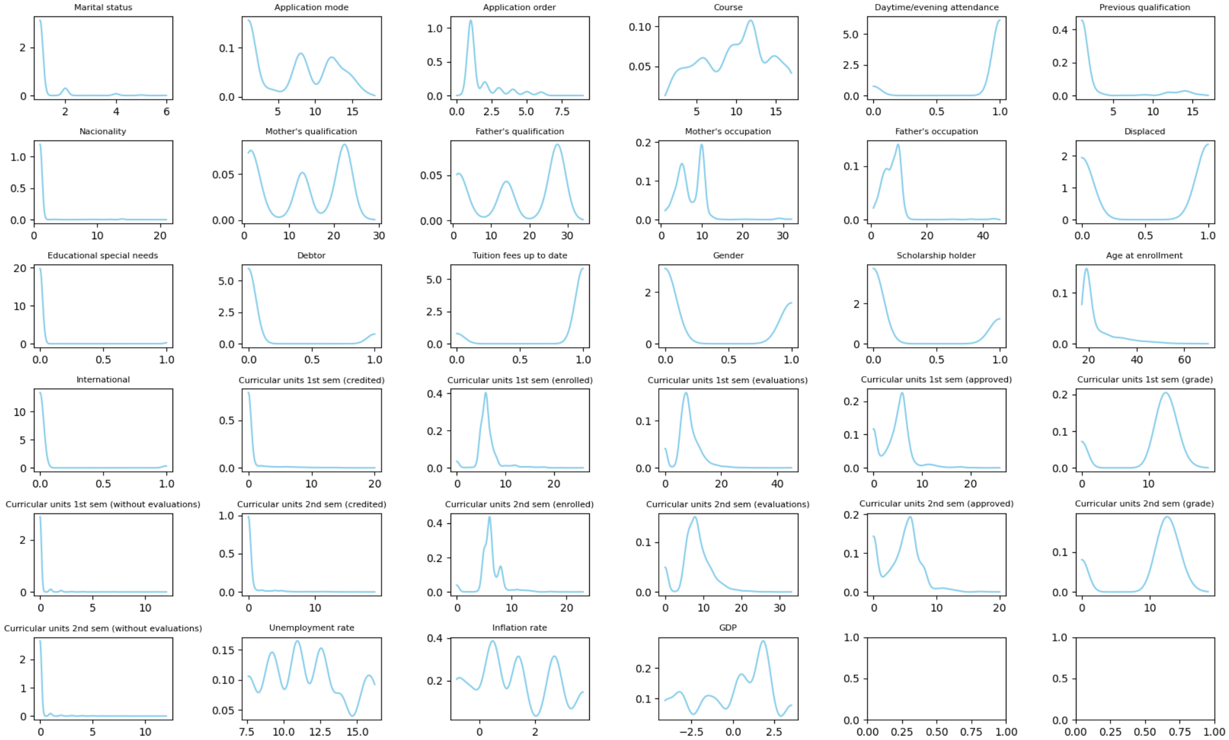


7. Gráfico de Densidad

In [ ]:

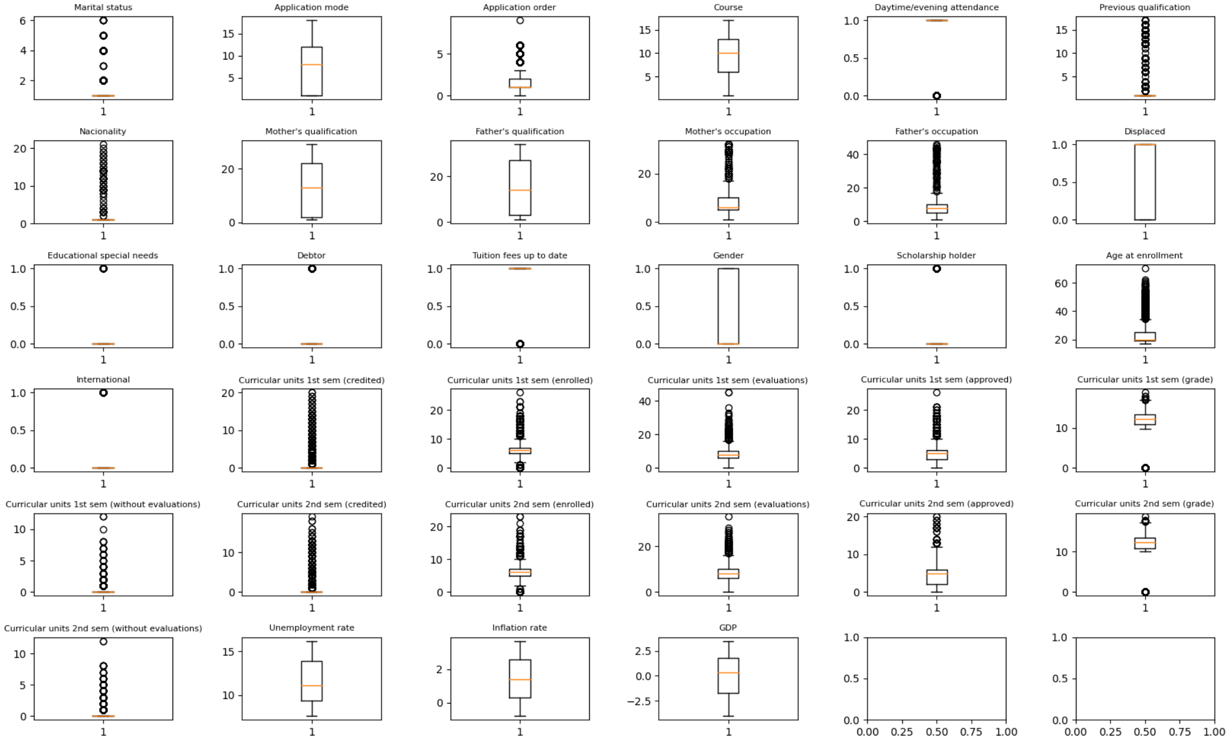
```
from scipy.stats import kde

# Crear gráficas de densidad para cada columna
fig, axs = plt.subplots(nrows=6, ncols=6, figsize=(20, 12))
fig.subplots_adjust(hspace=0.5, wspace=0.5)

for i, col in enumerate(df_numeric.columns):
    ax = axs[i//6, i%6] # seleccionar el subplot correspondiente
    density = kde.gaussian_kde(df_numeric[col])
    x = np.linspace(df_numeric[col].min(), df_numeric[col].max(), 1000)
    ax.plot(x, density(x), color='skyblue')
    ax.set_title(col, fontsize=8)
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_16716\4045357611.py:9: DeprecationWarning:
Please use `gaussian_kde` from the `scipy.stats` namespace, the `scipy.stats.kde` nam
espace is deprecated.
  density = kde.gaussian_kde(df_numeric[col])
```



8. Boxplot

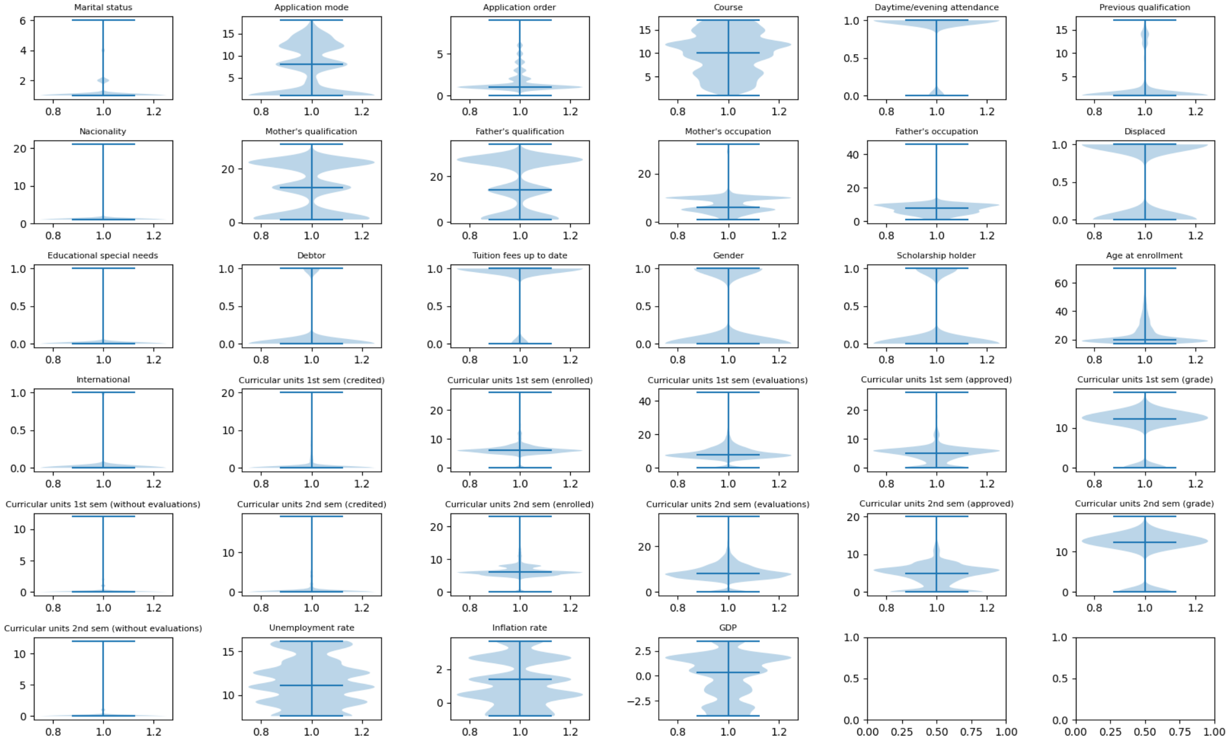
```
# Crear histograma
fig, axs = plt.subplots(nrows=6, ncols=6, figsize=(20, 12))
fig.subplots_adjust(hspace=0.5, wspace=0.5)

# Crear boxplot para cada columna
for i, col in enumerate(df_numeric.columns):
    ax = axs[i//6, i%6] # seleccionar el subplot correspondiente
    ax.boxplot(df_numeric[col])
    ax.set_title(col, fontsize=8)
```



9. Gráfico de Violín

In [ ]:

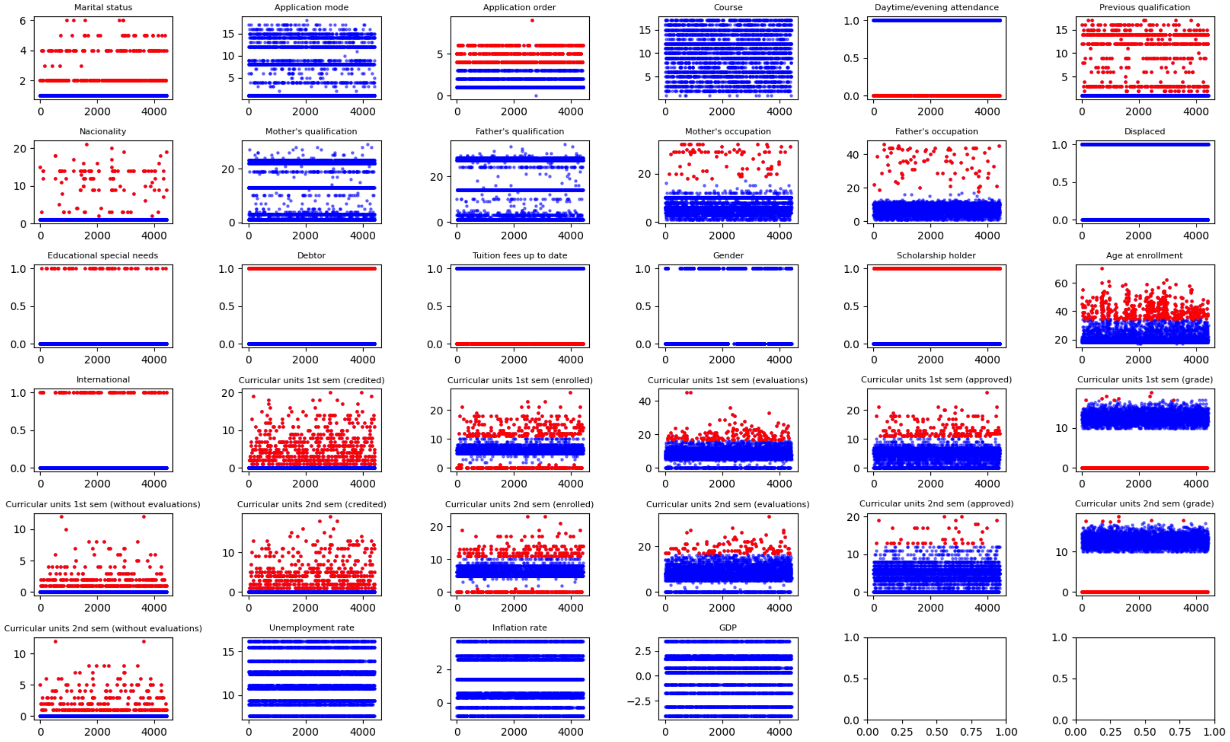
```
# Crear gráfico de violín
fig, axs = plt.subplots(nrows=6, ncols=6, figsize=(20, 12))
fig.subplots_adjust(hspace=0.5, wspace=0.5)

# Crear violín para cada columna
for i, col in enumerate(df_numeric.columns):
    ax = axs[i//6, i%6] # seleccionar el subplot correspondiente
    ax.violinplot(df_numeric[col], showmedians=True)
    ax.set_title(col, fontsize=8)
```



10. Gráfico de Dispersión

In [ ]:

```
# Crear histograma
fig, axs = plt.subplots(nrows=6, ncols=6, figsize=(20, 12))
fig.subplots_adjust(hspace=0.5, wspace=0.5)

# Crear gráfica de dispersión para cada columna
for i, col in enumerate(df_numeric.columns):
    ax = axs[i//6, i%6] # seleccionar el subplot correspondiente
    ax.scatter(range(len(df_numeric)), df_numeric[col], s=5, alpha=0.5, color='blue')
    # ax.scatter(range(len(dicc_outliers[col])), dicc_outliers[col], color='red', s=1)
    ax.set_title(col, fontsize=8)
    ax.set_xlabel('')
    ax.set_ylabel('')

    # Marcar los outliers con un color diferente
    q1 = df_numeric[col].quantile(0.25)
    q3 = df_numeric[col].quantile(0.75)
    iqr = q3 - q1
    outliers = df[(df[col] < q1 - 1.5*iqr) | (df[col] > q3 + 1.5*iqr)][col]
    ax.scatter(outliers.index, outliers, color='red', s=5)
```



Graficación con Plotly

1. Gráfico de Barras

```
In [ ]: import plotly.graph_objects as go

        colores = {"Graduate": 'green', "Dropout": 'orange', "Enrolled": 'blue'}
        frecuencias = df['Target'].value_counts()

        fig = go.Figure(go.Bar(x=frecuencias.index, y=frecuencias.values, marker_color=[colore
        fig.update_layout(title='Balance de las Clases', xaxis_title='Categorías', yaxis_title
```

2. Gráfica de Barras Apiladas

```
In [ ]: # Obtener las frecuencias de cada subcategoría
        graduados = df[df['Target'] == 'Graduate']['Target'].value_counts()
        dropouts = df[df['Target'] == 'Dropout']['Target'].value_counts()
        enrolled = df[df['Target'] == 'Enrolled']['Target'].value_counts()

        # Crear la figura con las barras apiladas
        fig = go.Figure()
        fig.add_trace(go.Bar(x=graduados.index, y=graduados.values, name='Graduate', marker_co
        fig.add_trace(go.Bar(x=graduados.index, y=dropouts.values, name='Dropout', marker_colo
        fig.add_trace(go.Bar(x=graduados.index, y=enrolled.values, name='Enrolled', marker_col
        fig.update_layout(title='Balance de las Clases', xaxis_title='Categorías', yaxis_title
```

3. Gráfica de Pastel

In [ ]:

```
# Crear gráfica de pastel
fig = go.Figure(data=[go.Pie(labels=frecuencias.index, values=frecuencias.values, mark

# Configurar diseño de la gráfica
fig.update_layout(title='Balance de las Clases')
```

4. Mapa de Calor

In [ ]:

```
import plotly.express as px

fig = px.imshow(df_pearson,
                x=df_pearson.columns,
                y=df_pearson.columns,
                color_continuous_scale='RdBu',
                zmin=-1,
                zmax=1)

fig.update_layout(width=800, height=800, title="Matriz de correlación")

fig.show()
```

5. Histograma

In [ ]:

```
from plotly.subplots import make_subplots

fig = make_subplots(rows=6, cols=6, subplot_titles=df_numeric.columns)

for i, col in enumerate(df_numeric.columns):
    ax = i//6 + 1
    ay = i%6 + 1
    fig.add_trace(go.Histogram(x=df_numeric[col], name=col), ax, ay)

fig.update_layout(title='Histogramas de cada columna', height=1000, width=2000, showle
```

6. Histograma con Densidad

In [ ]:

```
import plotly.figure_factory as ff
import numpy as np

hist_data = [df_numeric['Course']]
group_labels = ['Course'] # name of the dataset

fig = ff.create_distplot(hist_data, group_labels, colors=['rgb(0,125,125)'], show_rug=
fig.update_layout(title='Course', showlegend=False)
```

7. Gráfico de Densidad

In [ ]:

```python
import plotly.figure_factory as ff
import numpy as np

hist_data = [df_numeric['Curricular units 1st sem (approved)']]
group_labels = ['Curricular units 1st sem (approved)'] # name of the dataset

fig = ff.create_distplot(hist_data, group_labels, colors=['rgb(255,0,0)'], show_hist=F
fig.add_trace(go.Scatter(x=fig['data'][0]['x'], y=fig['data'][0]['y'], fill='tozeroy',
fig.update_layout(title='Curricular units 1st sem (approved)', showlegend=False)
```

8. Gráfico de Violín

In [ ]:

```python
import plotly.graph_objs as go

# Crear gráficas de densidad para cada columna
fig = make_subplots(rows=6, cols=6, subplot_titles=df_numeric.columns)

for i, col in enumerate(df_numeric.columns):
    ax = (i // 6) + 1
    ay = (i % 6) + 1
    fig.add_trace(go.Violin(y=df_numeric[col], name=col, box_visible=True), ax, ay)

fig.update_layout(title='Gráficos de densidad de cada columna', height=1000, width=200
```

9. Boxplot

In [ ]:

```python
import plotly.graph_objs as go

# Crear gráficas de densidad para cada columna
fig = make_subplots(rows=6, cols=6, subplot_titles=df_numeric.columns)

for i, col in enumerate(df_numeric.columns):
    ax = (i // 6) + 1
    ay = (i % 6) + 1
    fig.add_trace(go.Box(y=df_numeric[col], name=col), ax, ay)

fig.update_layout(title='Gráficos de densidad de cada columna', height=1000, width=200
```

10. Gráfico de Dispersión

In [ ]:

```python
# import plotly.graph_objects as go

# fig = make_subplots(rows=6, cols=6, subplot_titles=df_numeric.columns)

# for i, col in enumerate(df_numeric.columns):
#     ax = i//6 + 1
#     ay = i%6 + 1
#     fig.add_trace(go.Scatter(y=df_numeric[col], mode='markers', name=col), ax, ay)
#     fig.update_traces(histnorm='probability density', selector=dict(type='histogram'
# fig.update_layout(title='Densidad de cada columna', height=1000, width=2000, showleg
```