# MODELO PREDICTIVO DE ALTERACIONES LUMBARES A PARTIR DE DATOS BIOMECÁNICOS DE LA COLUMNA VERTEBRAL Y LA PELVIS.

David Prados Medina.

Certificado de Big Data & IA: Proyecto final.

# Índice

**Introducción** ..... **3**

**Comprensión del negocio** ..... **4**

Caso de uso ..... 4

Objetivos del negocio ..... 4

Beneficios esperados ..... 4

**Entendimiento de los datos (EDA)** ..... **5**

Preparación ..... 5

Cargar y conocer los datos ..... 5

Nulos y duplicados ..... 7

Estadísticas descriptivas y distribuciones ..... 8

Variables de alineación lumbopélvica ..... 8

Variables morfológicas y estructurales ..... 9

Variables de orientación global de la columna ..... 9

Detección de outliers (IQR) ..... 10

Variables categóricas: Cardinalidad y niveles raros ..... 10

Correlaciones numéricas (Pearson) ..... 11

Relaciones entre variables (Scatter plots, boxplots y crosstab) ..... 12

**Preparación de los datos (Ingeniería de características)** ..... **14**

Separación inicial ..... 14

Encoding de la variable objetivo ..... 15

Tratamiento de outliers o valores extremos ..... 16

Generación de nuevas características relevantes ..... 17

Eliminación de columnas redundantes ..... 17

**Modelado** ..... **18**

Random Forest ..... 18

Regresión logística ..... 18

**Evaluación** ..... **19**

Random Forest ..... 19

Regresión logística ..... 20

**Conclusión** ..... **21**

# Introducción

El dolor lumbar es una de las afecciones musculoesqueléticas más comunes en la población, con múltiples causas relacionadas con la columna vertebral, la musculatura o la pelvis. Su detección temprana y la identificación de factores de riesgo son fundamentales para prevenir lesiones y diseñar tratamientos más efectivos en fisioterapia.

Debido a mi etapa anterior como fisioterapeuta, he decidido realizar este proyecto con el objetivo de desarrollar un modelo predictivo que clasifique a los pacientes como normales o con alteraciones lumbares, a través de medidas biomecánicas y morfológicas de la columna vertebral y la pelvis. Para ello, he utilizado el dataset "Lower Back Pain Symptoms Dataset", compuesto por 310 observaciones y 12 variables numéricas relacionadas con la biomecánica y la morfología del paciente, como la inclinación pélvica, el grado de espondilolistesis o el ángulo de la lordosis lumbar, además de la variable objetivo-binaria ("Normal" / "Abnormal").

El proyecto combina conocimientos clínicos y técnicas de Machine Learning, pasando por las fases de comprensión del negocio, entendimiento de los datos (EDA), preparación de los datos (ingeniería de características), modelado y evaluación de las métricas resultantes.

He decidido realizar el entrenamiento de dos modelos, Random Forest y regresión logística, con el objetivo de comparar su desempeño y seleccionar la mejor alternativa la predicción de alteraciones lumbares.

# Comprensión del negocio

## Caso de uso

El caso de uso de este proyecto se sitúa en el ámbito de la fisioterapia y la prevención de patologías lumbares. A partir de parámetros biomecánicos de la columna vertebral y la pelvis, el modelo predictivo buscará clasificar a un paciente como “Normal” o “Abnormal”, en función de si presenta alteraciones asociadas a patología lumbar.

El modelo predictivo se plantea como una herramienta de apoyo de decisiones clínicas, especialmente útil en fases de valoración inicial.

## Objetivos del negocio

Los principales objetivos son:

- Anticipar posibles alteraciones lumbares a partir de datos biomecánicos objetivos.
- Apoyar al profesional sanitario en la identificación de pacientes con riesgo de patología lumbar.
- Optimizar el tiempo de evaluación clínica, reduciendo la dependencia de valoraciones menos objetivas.
- Mejorar la prevención de dolor lumbar.

## Beneficios esperados

La implementación de un modelo predictivo como el propuesto puede aportar los siguientes beneficios:

- Mejora de la detección precoz de alteraciones biomecánicas relacionadas con el dolor lumbar.
- Reducción del riesgo de falsos negativos, evitando clasificar como “Normal” a pacientes con posibles patologías a futuro.
- Apoyo a la toma de decisiones clínicas, especialmente para fisioterapeutas con menos experiencia.
- Estandarizar los procesos de valoración, disminuyendo la variabilidad y la subjetividad entre profesionales.
- Potencial reducción de costes sanitarios, gracias a intervenciones tempranas y preventivas.

# Entendimiento de los datos (EDA)

La fase de entendimiento de los datos (EDA: Exploratory Data Analysis) es una parte clave para entrenar el modelo, ya que conocer la estructura, los nulos, los valores atípicos y las relaciones de mis datos es muy importante para la realización del modelo.

## Preparación

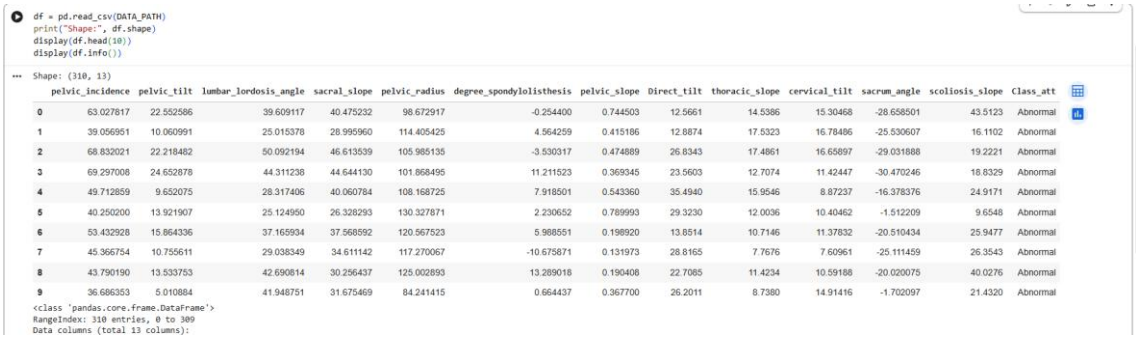
Importo las librerías y configuro las opciones básicas para gráficos en mi notebook. En este caso usaré la librería matplotlib y la librería seaborn para la representación gráfica.

```
import os, sys, platform
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

print("Python:", sys.version.split()[0], "| Plataforma:", platform.platform())
print("Pandas:", pd.__version__)

# Estilo de gráficos (simple y consistente)
plt.rcParams['figure.figsize'] = (8, 5)
plt.rcParams['axes.grid'] = True

# Ruta del dataset
DATA_PATH = "/content/Dataset_spine.csv"
```

```
Python: 3.12.12 | Plataforma: Linux-6.6.105+-x86_64-with-glibc2.35
Pandas: 2.2.2
```

## Cargar y conocer los datos

Abro el dataset y veo su forma, tipos y una muestra de 10 filas.



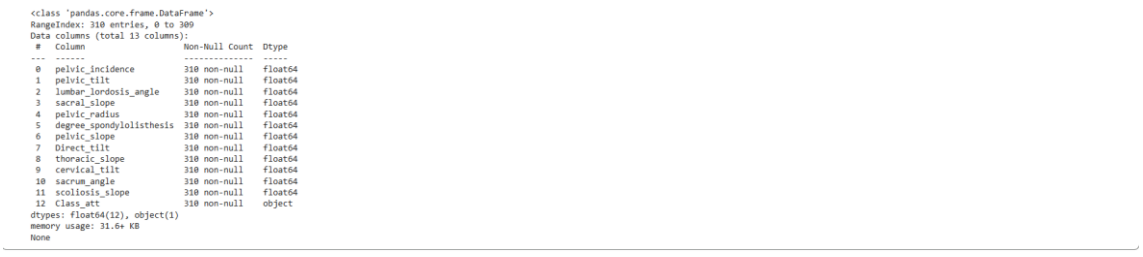
```
df = pd.read_csv(DATA_PATH)
print("Shape:", df.shape)
display(df.head(10))
display(df.info())
```

```
Shape: (310, 13)
```

| | pelvic_incidence | pelvic_tilt | lumbar_lordosis_angle | sacral_slope | pelvic_radius | degree_spondylolisthesis | pelvic_slope | Direct_tilt | thoracic_slope | cervical_tilt | sacrum_angle | scoliosis_slope | Class_att |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63.027817 | 22.552586 | 39.609117 | 40.475232 | 98.672917 | -0.254400 | 0.744503 | 12.5661 | 14.5386 | 15.30468 | -28.658501 | 43.5123 | Abnormal |
| 1 | 39.056951 | 10.060991 | 25.015378 | 28.995960 | 114.405425 | 4.564259 | 0.415186 | 12.8874 | 17.5323 | 16.78486 | -25.530607 | 16.1102 | Abnormal |
| 2 | 68.832021 | 22.218482 | 50.092194 | 46.613539 | 105.985135 | -3.530317 | 0.474889 | 26.8343 | 17.4861 | 16.65897 | -29.031888 | 19.2221 | Abnormal |
| 3 | 69.297008 | 24.652878 | 44.311238 | 44.644130 | 101.868495 | 11.211523 | 0.369345 | 23.5603 | 12.7074 | 11.42447 | -30.470246 | 18.8329 | Abnormal |
| 4 | 49.712859 | 9.652075 | 28.317406 | 40.060784 | 108.168725 | 7.918501 | 0.543360 | 35.4940 | 15.9546 | 8.87237 | -16.378376 | 24.9171 | Abnormal |
| 5 | 40.250200 | 13.921907 | 25.124950 | 26.328293 | 130.327871 | 2.230652 | 0.789993 | 29.3230 | 12.0036 | 10.40462 | -1.512209 | 9.6548 | Abnormal |
| 6 | 53.432928 | 15.864336 | 37.165934 | 37.568592 | 120.567523 | 5.988551 | 0.198920 | 13.8514 | 10.7146 | 11.37832 | -20.510434 | 25.9477 | Abnormal |
| 7 | 45.366754 | 10.755611 | 29.038349 | 34.611142 | 117.270067 | -10.675871 | 0.131973 | 28.8165 | 7.7676 | 7.60961 | -25.111459 | 26.3543 | Abnormal |
| 8 | 43.790190 | 13.533753 | 42.690814 | 30.256437 | 125.002893 | 13.289018 | 0.190408 | 22.7085 | 11.4234 | 10.59188 | -20.020075 | 40.0276 | Abnormal |
| 9 | 36.686353 | 5.010884 | 41.948751 | 31.675469 | 84.241415 | 0.664437 | 0.367700 | 26.2011 | 8.7380 | 14.91416 | -1.702097 | 21.4320 | Abnormal |

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 310 entries, 0 to 309
Data columns (total 13 columns):
```

El dataset contiene 310 filas y 13 columnas.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 310 entries, 0 to 309
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   pelvic_incidence          310 non-null    float64
 1   pelvic_tilt               310 non-null    float64
 2   lumbar_lordosis_angle     310 non-null    float64
 3   sacral_slope              310 non-null    float64
 4   pelvic_radius             310 non-null    float64
 5   degree_spondylolisthesis  310 non-null    float64
 6   pelvic_slope              310 non-null    float64
 7   Direct_tilt               310 non-null    float64
 8   thoracic_slope            310 non-null    float64
 9   cervical_tilt             310 non-null    float64
 10  sacrum_angle              310 non-null    float64
 11  scoliosis_slope           310 non-null    float64
 12  Class_att                 310 non-null    object
dtypes: float64(12), object(1)
memory usage: 31.6+ KB
None
```

Las columnas corresponden a:

- **Pelvic incidence (Incidencia pélvica):** Parámetro anatómico fijo que describe la relación entre la pelvis y la columna.
- **Pelvic tilt (Inclinación pélvica):** Mide la rotación de la pelvis en el plano sagital.
- **Lumbar lordosis angle (Ángulo de lordosis lumbar):** Cuantifica la curvatura fisiológica de la columna lumbar.
- **Sacral slope (Pendiente sacra):** Ángulo entre la base del sacro y la horizontal.
- **Pelvic radius (radio pélvico):** Distancia que representa la morfología pélvica.
- **Degree spondylolisthesis (Grado de espondilolistesis):** Indica el desplazamiento de una vértebra sobre la otra.
- **Pelvic slope (Pendiente pélvica):** Describe la orientación de la pelvis en relación con el eje corporal.
- **Direct tilt (Inclinación directa):** Mide la orientación del tronco respecto al eje vertical.
- **Thoracic slope (Pendiente torácica):** Describe la orientación de la columna torácica.
- **Cervical tilt (Inclinación cervical):** Mide la orientación de la columna cervical.
- **Sacrum angle (Ángulo sacro):** Representa la orientación del sacro dentro del complejo lumbopélvico.
- **Scoliosis slope (Pendiente escoliótica):** Indica la orientación lateral de la columna.
- **Class att (Clase):** Se divide en dos posibles parámetros:
    - *Normal:* Parámetros dentro de rangos fisiológicos.
    - *Abnormal:* Alteraciones biomecánicas asociadas a patología lumbar.

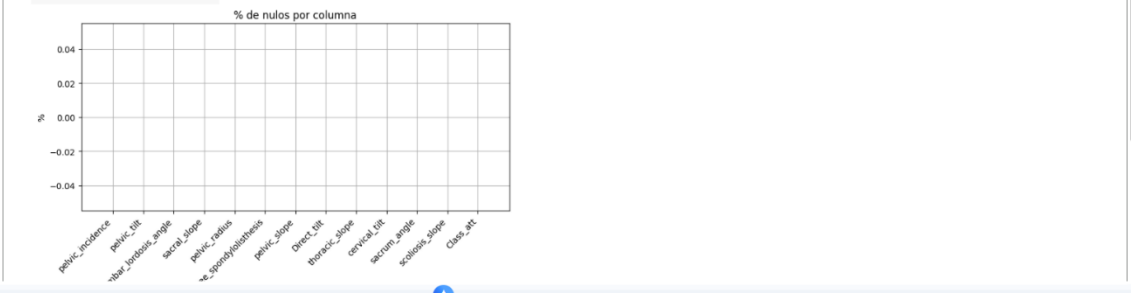
## Nulos y duplicados

Busco valores vacíos y filas repetidas. Si encuentro columnas con muchos nulos o con valores duplicados, las eliminaré en fases posteriores.

```
nulls_count = df.isna().sum()
nulls_pct = (df.isna().mean()*100).round(2)
profile_nulls = pd.DataFrame({"nulls": nulls_count, "%_nulls": nulls_pct}).sort_values("%_nulls", ascending=False)
print("Duplicados:", df.duplicated().sum())
display(profile_nulls)

# Visualización básica del % de nulos por columna (matplotlib, 1 gráfico)
fig, ax = plt.subplots()
ax.bar(profile_nulls.index.astype(str), profile_nulls['%_nulls'])
ax.set_title('% de nulos por columna')
ax.set_ylabel('%')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
Duplicados: 0
```

| | nulls | %_nulls |
|---|---|---|
| pelvic_incidence | 0 | 0.0 |
| pelvic_tilt | 0 | 0.0 |
| lumbar_lordosis_angle | 0 | 0.0 |
| sacral_slope | 0 | 0.0 |
| pelvic_radius | 0 | 0.0 |
| degree_spondylolisthesis | 0 | 0.0 |
| pelvic_slope | 0 | 0.0 |
| Direct_tilt | 0 | 0.0 |
| thoracic_slope | 0 | 0.0 |
| cervical_tilt | 0 | 0.0 |
| sacrum_angle | 0 | 0.0 |
| scoliosis_slope | 0 | 0.0 |
| Class_att | 0 | 0.0 |




Este dataset no tiene ni elementos duplicados ni datos nulos.

## Estadísticas descriptivas y distribuciones

Esta etapa se realiza con la finalidad de entender la forma de las variables numéricas y ver posibles valores extremos.

```
df_conv = df.copy()
desc_num = df_conv.select_dtypes(include=['int64','float64']).describe().T
display(desc_num)

num_cols = df_conv.select_dtypes(include=['int64','float64']).columns.tolist()
if len(num_cols) > 0:
    fig, ax = plt.subplots()
    col0 = num_cols[0]
    ax.hist(df_conv[col0].dropna(), bins=20)
    ax.set_title(f'Distribución de {col0}')
    plt.show()
```

1 to 12 of 12 entries Filter

| index | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| pelvic_incidence | 310.0 | 60.49665292951613 | 17.236520321708856 | 26.14792141 | 46.43029421 | 58.691038135 | 72.87769551 | 129.8340406 |
| pelvic_tilt | 310.0 | 17.542821967970966 | 10.008330258206636 | -6.554948347 | 10.66706906 | 16.35768863 | 22.12039474 | 49.4318636 |
| lumbar_lordosis_angle | 310.0 | 51.93092960345161 | 18.554063962761177 | 14.0 | 37.0 | 49.56239828 | 62.99999999 | 125.7423855 |
| sacral_slope | 310.0 | 42.953830961419356 | 13.423102164839566 | 13.3669307 | 33.3471220125 | 42.404912075 | 52.695888355 | 121.4296656 |
| pelvic_radius | 310.0 | 117.92065502380645 | 13.317377044904567 | 70.08257486 | 110.7091963 | 118.2681783 | 125.467674425 | 163.0710405 |
| degree_spondylolisthesis | 310.0 | 26.296694437867735 | 37.559026554872374 | -11.05817866 | 1.60372667375 | 11.767933769999999 | 41.287351962500004 | 418.5430821 |
| pelvic_slope | 310.0 | 0.4729792534451613 | 0.28578674082869 | 0.003220264 | 0.2243670405 | 0.475988571 | 0.7048461815 | 0.998826684 |
| Direct_tilt | 310.0 | 21.321526129032257 | 8.639423372673225 | 7.027 | 13.054400000000001 | 21.90715 | 28.954075000000003 | 36.7439 |
| thoracic_slope | 310.0 | 13.06451129032258 | 3.3997128499770826 | 7.0378 | 10.4178 | 12.93845 | 15.889525 | 19.324 |
| cervical_tilt | 310.0 | 11.933316741935485 | 2.8932653038026737 | 7.0306 | 9.54114 | 11.953835 | 14.37181 | 16.82108 |
| sacrum_angle | 310.0 | -14.053139487096773 | 12.225582016950817 | -35.287375 | -24.28952225 | -14.6228555 | -3.49709425 | 6.972071 |
| scoliosis_slope | 310.0 | 25.64598064516129 | 10.450558176100033 | 7.0079 | 17.189075000000003 | 24.93195 | 33.979600000000005 | 44.3412 |

Show 25 per page

## Variables de alineación lumbopélvica

**Incidencia pélvica:**

Presenta una media de 60,50 con una desviación estándar de 17,24 lo que indica alta variabilidad anatómica entre individuos. El amplio rango de valores sugiere la coexistencia de formas pélvicas normales y patológicas.

**Inclinación pélvica:**

La media es de 17,24 con valores que van desde -6,55 hasta 49,43 reflejando importantes diferencias posturales en los pacientes. Esta dispersión puede estar asociada a mecanismos compensatorios ante alteraciones lumbares.

**Ángulo de lordosis lumbar**:

Presenta una media de 51,93 y una desviación estándar de 18,55 lo que indica una variabilidad considerable en la curvatura lumbar. Los valores extremos refuerzan la hipótesis de que esta variable puede ser importante para clasificar entre pacientes normales y pacientes con alteraciones.

**Pendiente sacra:**

Con una media de 42,95, esta variable también muestra un rango amplio, coherente con su relación con la lordosis lumbar y la incidencia pélvica.

## Variables morfológicas y estructurales

**Radio pélvico:**

Presenta una media de 117,92 y una desviación relativamente baja (13,32). Esto indica que es una variable más estable, posiblemente menos sensible a alteraciones funcionales.

**Grado de espondilolistesis:**

Esta variable destaca por su alta dispersión (37,56) y la existencia de valores extremos muy elevados, alcanzando un máximo de 418,54. Este comportamiento es significativo y sugiere que puede tener un alto peso predictivo.

## Variables de orientación global de la columna

**Ángulo sacro:**

Presenta una media negativa (-14,05) lo cual es esperable debido a su definición. Su rango sugiere diferentes orientaciones sacras entre los pacientes analizados.

**Resto de variables:**

Las variables restantes (pendiente pélvica, inclinación pélvica, pendiente torácica, inclinación cervical y pendiente escoliótica) están relacionadas con la orientación global de la columna, por lo que las he analizado de forma conjunta. Todas ellas presentan una menor dispersión de los valores, pero con suficiente variabilidad para reflejar diferencia entre la alineación global del raquis.

## Detección de outliers (IQR)

Uso el rango intercuartílico (IQR). Los valores fuera de [Q1-1.5*IQR, Q3+1.5*IQR] se marcan como atípicos potenciales.

```
num = df_conv.select_dtypes(include=['int64','float64'])
if num.shape[1] > 0:
    q1, q3 = num.quantile(0.25), num.quantile(0.75)
    iqr = q3 - q1
    low, high = q1 - 1.5*iqr, q3 + 1.5*iqr
    out_iqr = ~num.apply(lambda s: s.between(low[s.name], high[s.name]))
    out_counts = out_iqr.sum()
    display(pd.DataFrame({'outliers_IQR': out_counts}))
```

| | outliers_IQR |
|---|---|
| pelvic_incidence | 3 |
| pelvic_tilt | 13 |
| lumbar_lordosis_angle | 1 |
| sacral_slope | 1 |
| pelvic_radius | 11 |
| degree_spondylolisthesis | 10 |
| pelvic_slope | 0 |
| Direct_tilt | 0 |
| thoracic_slope | 0 |
| cervical_tilt | 0 |
| sacrum_angle | 0 |
| scoliosis_slope | 0 |

Este resultado sugiere que:

Las variables con outliers contienen información relevante para discriminar entre condición lumbar normal y anormal. Entre estas variables destacan la inclinación pélvica (13 outliers), el radio pélvico (11 outliers) y el grado de espondilolistesis (10 outliers). Además, estas variables están más relacionadas con la biomecánica lumbar y pélvica.

Por otro lado, el resto de las variables que no presentan outliers representan datos más estables, con menos variabilidad extrema y con un posible menor impacto en el proceso de creación del modelo.

## Variables categóricas: Cardinalidad y niveles raros

En esta etapa veo la distribución de categorías y busco detectar niveles pocos frecuentes (<5%).

```
cat_cols = df_conv.select_dtypes(include=['category','object']).columns.tolist()
summary = {}
for c in cat_cols:
    vc = df_conv[c].value_counts(dropna=False, normalize=True).mul(100).round(2)
    rare = vc[vc < 5.0]
    summary[c] = {"cardinality": int(df_conv[c].nunique(dropna=False)), "rare_levels_%": rare.to_dict()}
display(pd.DataFrame(summary).T)
```

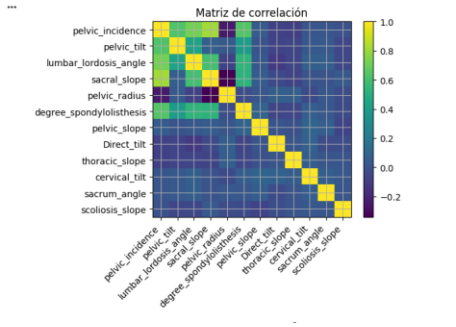
| | cardinality | rare_levels_% |
|---|---|---|
| Class_att | 2 | {} |

En este dataset, la única variable categórica es mi futura variable objetivo (Class att o clase). Esta variable presenta dos niveles (Normal y Abnormal), sin desbalance relevante, por lo que no se identifican niveles raros.

## Correlaciones numéricas (Pearson)

Identifico las relaciones fuertes y posibles redundancias.

```
corr = df_conv.select_dtypes(include=['int64','float64']).corr(numeric_only=True)
if corr.shape[0] > 0:
    fig, ax = plt.subplots(figsize=(6,5))
    im = ax.imshow(corr.values, aspect='auto')
    ax.set_xticks(range(len(corr.columns)))
    ax.set_xticklabels(corr.columns, rotation=45, ha='right')
    ax.set_yticks(range(len(corr.columns)))
    ax.set_yticklabels(corr.columns)
    ax.set_title('Matriz de correlación')
    plt.colorbar(im, ax=ax)
    plt.tight_layout(); plt.show()

    corr_abs = corr.abs()
    upper = corr_abs.where(np.triu(np.ones(corr_abs.shape), k=1).astype(bool))
    pairs = upper.stack().sort_values(ascending=False)
    display(pairs.head(10))
```




| | | 0 |
|---|---|---|
| pelvic_incidence | sacral_slope | 0.814960 |
| | lumbar_lordosis_angle | 0.717282 |
| | degree_spondylolisthesis | 0.638743 |
| | pelvic_tilt | 0.629199 |
| lumbar_lordosis_angle | sacral_slope | 0.598387 |
| | degree_spondylolisthesis | 0.533667 |
| sacral_slope | degree_spondylolisthesis | 0.523557 |
| pelvic_tilt | lumbar_lordosis_angle | 0.432764 |
| | degree_spondylolisthesis | 0.397862 |
| sacral_slope | pelvic_radius | 0.342128 |

dtype: float64

En este análisis destaca principalmente la alta correlación entre la incidencia pélvica y el ángulo sacro (0,815) o la correlación entre la incidencia pélvica y el ángulo de lordosis lumbar (0,717). Estas relaciones muestran coherencia anatómica y biomecánica de la pelvis y la columna lumbar. Esta relación puede generar redundancia en la posterior creación del modelo.

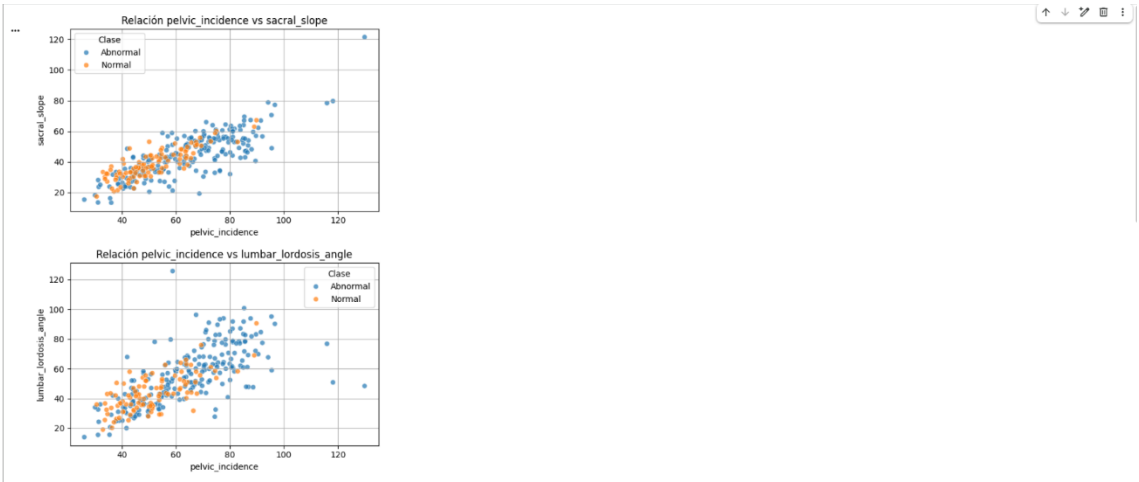
Además, las variables asociadas a la desalineación lumbar (grado de espondilolistesis) presenta una asociación moderada con la incidencia pélvica y la lordosis, indicando que puede ser relevante para la predicción de alteraciones lumbares.

## Relaciones entre variables (Scatter plots, boxplots y crosstab)

Realizo una combinación de scatter plots, boxplots y análisis de correlaciones para mostrar de forma visual las relaciones entre las variables.

```python
# Scatter plots de variables numéricas más correlacionadas
scatter_pairs = [
    ('pelvic_incidence', 'sacral_slope'),
    ('pelvic_incidence', 'lumbar_lordosis_angle'),
    ('degree_spondylolisthesis', 'lumbar_lordosis_angle')
]

for x_col, y_col in scatter_pairs:
    plt.figure(figsize=(6,4))
    sns.scatterplot(x=df_conv[x_col], y=df_conv[y_col], hue=df_conv['Class_att'], alpha=0.7)
    plt.title(f'Relación {x_col} vs {y_col}')
    plt.xlabel(x_col)
    plt.ylabel(y_col)
    plt.legend(title='Clase')
    plt.tight_layout()
    plt.show()

# Boxplots de variables numéricas según la clase
boxplot_cols = [
    'pelvic_incidence',
    'degree_spondylolisthesis',
    'lumbar_lordosis_angle',
    'pelvic_tilt'
]
```

```python
for col in boxplot_cols:
    plt.figure(figsize=(6,4))
    sns.boxplot(x='Class_att', y=col, data=df_conv)
    plt.title(f'Distribución de {col} según clase')
    plt.xlabel('Clase')
    plt.ylabel(col)
    plt.tight_layout()
    plt.show()

# Crosstab
cat_cols = df_conv.select_dtypes(include=['category','object']).columns.tolist()
for c in cat_cols:
    if c != 'Class_att':
        ct = pd.crosstab(df_conv['Class_att'], df_conv[c], normalize='index').round(3)
        display(ct)
```




Analizo las relaciones entre diferentes variables numéricas y la variable objetivo (Class att o clase).

Los scatter plots muestran que las variables incidencia pélvica y ángulo sacro presentan tendencias claras según la clase.

Los boxplots reflejan diferencias entre la distribución de variables clave como el grado de espondilolistesis o el ángulo de la lordosis lumbar entre pacientes normales y pacientes con alteraciones lumbares, reforzando su relevancia para el modelo predictivo.

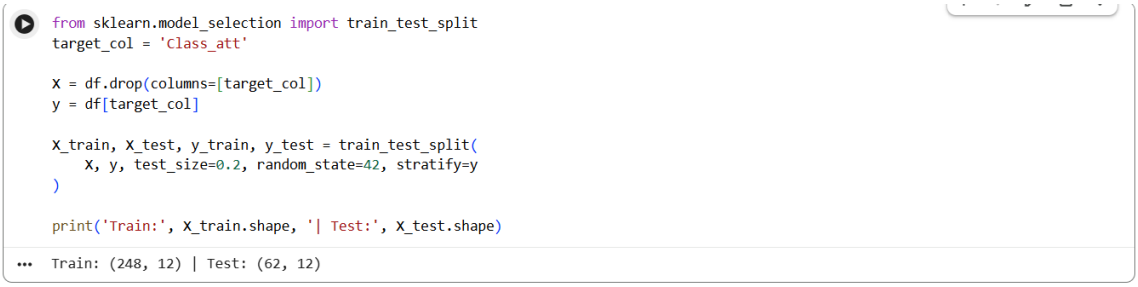
# Preparación de los datos (Ingeniería de características)

La fase de ingeniería de características, o preparación de los datos, es vital para obtener modelos predictivos precisos. En esta etapa se busca optimizar la calidad de los datos a través de la selección de variables relevantes, la eliminación de valores prescindibles y la normalización o estandarización de las variables que corresponda.

## Separación inicial

Defino train/test desde el principio, además del target_col (en mi caso Class att o clase).

Por otro lado, importo la librería sklearn, la librería más usada en Python para machine learning. Esto me permite añadir la función de train_test_split, que me permite dividir mi dataset en entrenamiento (train) y prueba (test).



```
from sklearn.model_selection import train_test_split
target_col = 'Class_att'

X = df.drop(columns=[target_col])
y = df[target_col]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print('Train:', X_train.shape, '| Test:', X_test.shape)
```

```
Train: (248, 12) | Test: (62, 12)
```

## Encoding de la variable objetivo

Convierto la columna Class att a 0/1 para facilitar el entrenamiento del modelo.

Para ello empiezo revisando los valores únicos y los tipos de datos.

```
print(df['Class_att'].unique())
print(df['Class_att'].dtype)
```

```
['Abnormal' 'Normal']
object
```

Convierto a string y elimino los espacios.

```
y_train = y_train.astype(str).str.strip()
y_test  = y_test.astype(str).str.strip()
```

Mapeo “Normal” a 0 y “Abnormal” a 1.

```
y_train = y_train.map({'Normal': 0, 'Abnormal': 1})
y_test  = y_test.map({'Normal': 0, 'Abnormal': 1})
```

Compruebo que el cambio se haya realizado correctamente a través de un feedback, imprimiendo los primeros valores y la distribución en train y en test.

```
# Primeros valores
print(y_train.head())
print(y_test.head())

# Distribución de clases
print("Distribución train:")
print(y_train.value_counts())
print("Distribución test:")
print(y_test.value_counts())
```

```
81     1
203    1
224    0
122    1
70     1
Name: Class_att, dtype: int64
226    0
21     1
209    1
99     1
50     1
Name: Class_att, dtype: int64
Distribución train:
Class_att
1    168
0     80
Name: count, dtype: int64
Distribución test:
Class_att
1    42
0    20
Name: count, dtype: int64
```

La codificación a 0 y 1 se ha realizado correctamente.

## Tratamiento de outliers o valores extremos

Genero banderas de valores extremos para valorar su efecto, genero FE_train y FE_test para no modificar los originales. Además, establezco los umbrales de las banderas de outliers basados en el EDA realizado anteriormente.

```
FE_train = X_train.copy()
FE_test  = X_test.copy()

FE_train['high_pelvic_radius'] = (FE_train['pelvic_radius'] > 140).astype(int)
FE_test['high_pelvic_radius']  = (FE_test['pelvic_radius'] > 140).astype(int)

FE_train['high_pelvic_tilt'] = (FE_train['pelvic_tilt'] > 30).astype(int)
FE_test['high_pelvic_tilt']  = (FE_test['pelvic_tilt'] > 30).astype(int)

FE_train['high_degree_spondy'] = (FE_train['degree_spondylolisthesis'] > 50).astype(int)
FE_test['high_degree_spondy']  = (FE_test['degree_spondylolisthesis'] > 50).astype(int)
```

Realizo una comprobación posterior para ver que los cambios se han realizado correctamente.

```
print("Primeras filas de FE_train:")
display(FE_train.head())

print("\nPrimeras filas de FE_test:")
display(FE_test.head())

print("\nDistribución de banderas de outliers en train:")
print(FE_train[['high_pelvic_radius', 'high_pelvic_tilt', 'high_degree_spondy']].sum())
```

```
Primeras filas de FE_train:
```

| | pelvic_incidence | pelvic_tilt | lumbar_lordosis_angle | sacral_slope | pelvic_radius | degree_spondylolisthesis | pelvic_slope | Direct_tilt | thoracic_slope | cervical_tilt | sacrum_angle | scoliosis_slope | high_pelvic_radius | high_pelvic_tilt | high_degree_spondy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **81** | 74.005541 | 21.122402 | 57.379502 | 52.883139 | 120.205983 | 74.555166 | 0.408965 | 10.5895 | 12.5946 | 15.87462 | -10.624659 | 31.9699 | 0 | 0 | 1 |
| **203** | 73.635962 | 9.711318 | 63.000000 | 63.924644 | 98.727930 | 26.975787 | 0.198909 | 30.8752 | 9.4553 | 11.84325 | -34.729173 | 26.8740 | 0 | 0 | 0 |
| **224** | 89.834676 | 22.639217 | 90.563461 | 67.195460 | 100.501192 | 3.040973 | 0.379933 | 9.4868 | 17.7556 | 10.98189 | -16.891891 | 28.0900 | 0 | 0 | 0 |
| **122** | 80.074914 | 48.069531 | 52.403439 | 32.005383 | 110.709912 | 67.727316 | 0.099941 | 20.2822 | 10.3082 | 15.89258 | -14.156070 | 39.9730 | 0 | 1 | 1 |
| **70** | 72.560702 | 17.385191 | 52.000000 | 55.175511 | 119.193724 | 32.108537 | 0.267292 | 26.9716 | 18.3111 | 13.67428 | -19.605044 | 22.7590 | 0 | 0 | 0 |

```
Primeras filas de FE_test:
```

| | pelvic_incidence | pelvic_tilt | lumbar_lordosis_angle | sacral_slope | pelvic_radius | degree_spondylolisthesis | pelvic_slope | Direct_tilt | thoracic_slope | cervical_tilt | sacrum_angle | scoliosis_slope | high_pelvic_radius | high_pelvic_tilt | high_degree_spondy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **226** | 63.969522 | 16.060945 | 63.123736 | 47.898577 | 142.380125 | 6.298971 | 0.134954 | 31.7659 | 10.9570 | 13.32169 | -9.939014 | 31.3141 | 1 | 0 | 0 |
| **21** | 54.919443 | 21.062332 | 42.200000 | 33.857110 | 125.212716 | 2.432561 | 0.175245 | 23.0791 | 14.2195 | 14.14196 | 3.780394 | 24.9278 | 0 | 0 | 0 |
| **209** | 48.259920 | 16.417462 | 36.329137 | 31.842457 | 94.882336 | 28.343799 | 0.388445 | 16.1775 | 15.0636 | 13.79474 | -8.044644 | 21.6135 | 0 | 0 | 0 |
| **99** | 58.521623 | 13.922286 | 41.467855 | 44.599337 | 115.514798 | 30.387984 | 0.401085 | 34.6931 | 10.3564 | 10.64403 | -26.051990 | 10.4338 | 0 | 0 | 0 |
| **50** | 55.285852 | 20.440118 | 34.000000 | 34.845733 | 115.877017 | 3.558372 | 0.680655 | 16.7110 | 15.9714 | 14.37627 | 4.779509 | 43.2610 | 0 | 0 | 0 |

```
Distribución de banderas de outliers en train:
high_pelvic_radius    11
high_pelvic_tilt      30
high_degree_spondy    54
dtype: int64
```

## Generación de nuevas características relevantes

Añado la columna “pelvic_to_sacral_ratio” dónde combino la incidencia pélvica entre el ángulo sacro.

```
FE_train['pelvic_to_sacral_ratio'] = FE_train['pelvic_incidence'] / FE_train['sacral_slope']
FE_test['pelvic_to_sacral_ratio']  = FE_test['pelvic_incidence'] / FE_test['sacral_slope']

print("Primeras filas de FE_train:")
display(FE_train.head())

print("\nPrimeras filas de FE_test:")
display(FE_test.head())
```

```
...s de FE_train:
```

| incidence | pelvic_tilt | lumbar_lordosis_angle | sacral_slope | pelvic_radius | degree_spondylolisthesis | pelvic_slope | Direct_tilt | thoracic_slope | cervical_tilt | sacrum_angle | scoliosis_slope | high_pelvic_radius | high_pelvic_tilt | high_degree_spondy | pelvic_to_sacral_ratio |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 74.005541 | 21.122402 | 57.379502 | 52.883139 | 120.205963 | 74.555166 | 0.406965 | 10.5895 | 12.5946 | 15.87462 | -10.624659 | 31.9699 | 0 | 0 | 1 | 1.399417 |
| 73.635962 | 9.711318 | 63.000000 | 63.924644 | 98.727930 | 26.975787 | 0.198909 | 30.8752 | 9.4553 | 11.84325 | -34.729173 | 26.6740 | 0 | 0 | 0 | 1.151918 |
| 89.834676 | 22.639217 | 90.563461 | 67.195460 | 100.501192 | 3.040973 | 0.379933 | 9.4868 | 17.7556 | 10.98189 | -16.891891 | 28.0900 | 0 | 0 | 0 | 1.336916 |
| 80.074914 | 48.069531 | 52.403439 | 32.005383 | 110.709912 | 67.727316 | 0.099941 | 20.2822 | 10.3082 | 15.89258 | -14.156070 | 39.9730 | 0 | 1 | 1 | 2.501920 |
| 72.560702 | 17.385191 | 52.000000 | 55.175511 | 119.193724 | 32.108537 | 0.267292 | 26.9716 | 18.3111 | 13.67428 | -19.605044 | 22.7590 | 0 | 0 | 0 | 1.315089 |

```
...s de FE_test:
```

| incidence | pelvic_tilt | lumbar_lordosis_angle | sacral_slope | pelvic_radius | degree_spondylolisthesis | pelvic_slope | Direct_tilt | thoracic_slope | cervical_tilt | sacrum_angle | scoliosis_slope | high_pelvic_radius | high_pelvic_tilt | high_degree_spondy | pelvic_to_sacral_ratio |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 63.959522 | 16.060945 | 63.123736 | 47.898577 | 142.360125 | 6.298971 | 0.134954 | 31.7659 | 10.9570 | 13.32169 | -9.939014 | 31.3141 | 1 | 0 | 0 | 1.335312 |
| 54.919443 | 21.062332 | 42.200000 | 33.857110 | 125.212716 | 2.432561 | 0.175245 | 23.0791 | 14.2195 | 14.14196 | 3.780394 | 24.9278 | 0 | 0 | 0 | 1.622095 |
| 48.259920 | 16.417462 | 36.329137 | 31.842457 | 94.882336 | 28.343799 | 0.388445 | 16.1775 | 15.0636 | 13.79474 | -8.044644 | 21.6135 | 0 | 0 | 0 | 1.515584 |
| 58.521623 | 13.922286 | 41.467855 | 44.599337 | 115.514798 | 30.387984 | 0.401085 | 34.6931 | 10.3564 | 10.64403 | -26.051990 | 10.4338 | 0 | 0 | 0 | 1.312164 |
| 55.285852 | 20.440118 | 34.000000 | 34.845733 | 115.877017 | 3.558372 | 0.680655 | 16.7110 | 15.9714 | 14.37627 | 4.779509 | 43.2610 | 0 | 0 | 0 | 1.586589 |

## Eliminación de columnas redundantes

Tras valorar los datos a través de la fase EDA, se mostró una gran correlación entre la incidencia pélvica y el ángulo de lordosis sacra. Además, en la etapa anterior he combinado los dos datos a través de la nueva columna “pelvic_to_sacral_ratio” por lo que voy a eliminar la columna sacral slope o ángulo sacro para evitar ruido innecesario al modelo y acelerar su entrenamiento, sin perder información relevante.

```
FE_train = FE_train.drop(columns=['sacral_slope'])
FE_test  = FE_test.drop(columns=['sacral_slope'])

print("Columnas restantes en FE_train:")
print(FE_train.columns)
```

```
... Columnas restantes en FE_train:
Index(['pelvic_incidence', 'pelvic_tilt', 'lumbar_lordosis_angle',
       'pelvic_radius', 'degree_spondylolisthesis', 'pelvic_slope',
       'Direct_tilt', 'thoracic_slope', 'cervical_tilt', 'sacrum_angle',
       'scoliosis_slope', 'high_pelvic_radius', 'high_pelvic_tilt',
       'high_degree_spondy', 'pelvic_to_sacral_ratio'],
      dtype='object')
```

# Modelado

En esta fase voy a realizar el entrenamiento de mi modelo predictivo, he escogido realizar un modelo de Random Forest, ya que es un modelo robusto que interpreta bien las variables numéricas y categóricas.

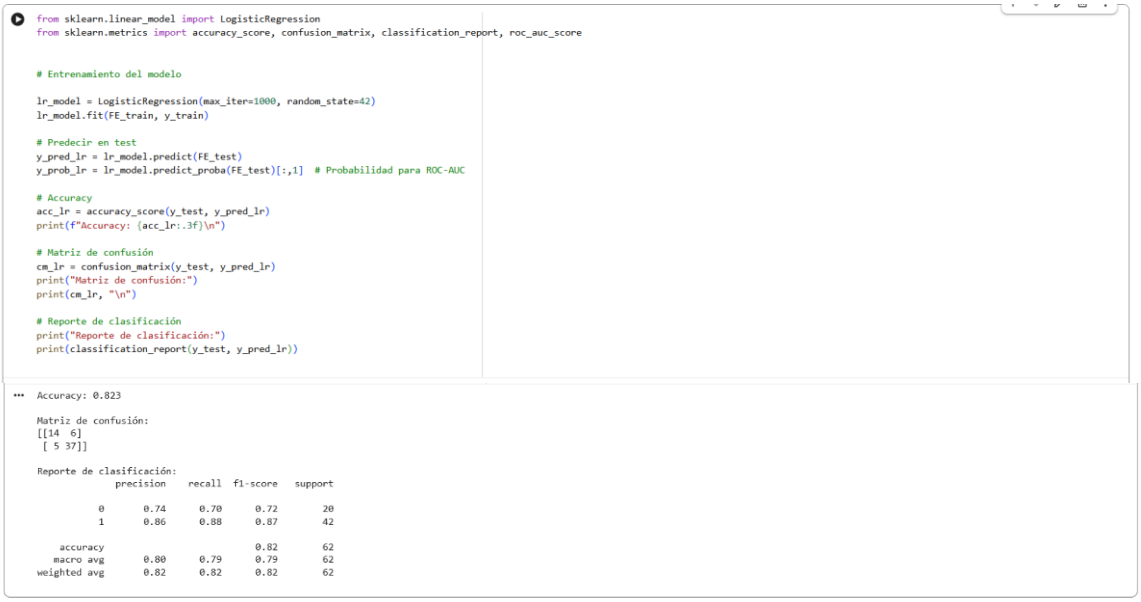
Más tarde, realizaré un segundo modelo para comparar, en este caso seleccionaré un modelo de Regresión logística y así podré visualizar el accuracy y las métricas de los dos modelos, como accuracy, precision, recall y F1-score.

## Random Forest

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Creación del modelo
rf_model = RandomForestClassifier(
    n_estimators=200,  # número de árboles
    random_state=42,
    max_depth=None,
    min_samples_split=2
)

# Entrenar
rf_model.fit(FE_train, y_train)

# Predecir en test
y_pred = rf_model.predict(FE_test)

# Evaluación
acc = accuracy_score(y_test, y_pred)
cm  = confusion_matrix(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Accuracy: {acc:.3f}")
print("Matriz de confusión:")
print(cm)
print("Reporte de clasificación:")
print(report)
```

```
Accuracy: 0.758
Matriz de confusión:
[[13  7]
 [ 8 34]]
Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.62      0.65      0.63        20
           1       0.83      0.81      0.82        42

    accuracy                           0.76        62
   macro avg       0.72      0.73      0.73        62
weighted avg       0.76      0.76      0.76        62
```

## Regresión logística



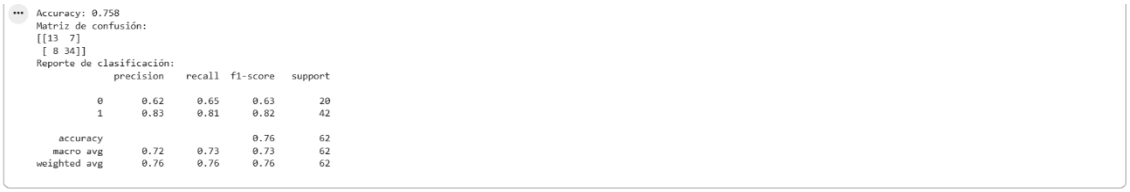
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_auc_score


# Entrenamiento del modelo

lr_model = LogisticRegression(max_iter=1000, random_state=42)
lr_model.fit(FE_train, y_train)

# Predecir en test
y_pred_lr = lr_model.predict(FE_test)
y_prob_lr = lr_model.predict_proba(FE_test)[:,1]  # Probabilidad para ROC-AUC

# Accuracy
acc_lr = accuracy_score(y_test, y_pred_lr)
print(f"Accuracy: {acc_lr:.3f}\n")

# Matriz de confusión
cm_lr = confusion_matrix(y_test, y_pred_lr)
print("Matriz de confusión:")
print(cm_lr, "\n")

# Reporte de clasificación
print("Reporte de clasificación:")
print(classification_report(y_test, y_pred_lr))
```

```
Accuracy: 0.823

Matriz de confusión:
[[14  6]
 [ 5 37]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.74      0.70      0.72        20
           1       0.86      0.88      0.87        42

    accuracy                           0.82        62
   macro avg       0.80      0.79      0.79        62
weighted avg       0.82      0.82      0.82        62
```

# Evaluación

En esta última etapa analizo las métricas de validación correspondientes a los algoritmos.

## Random Forest



```
Accuracy: 0.758
Matriz de confusión:
[[13  7]
 [ 8 34]]
Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.62      0.65      0.63        20
           1       0.83      0.81      0.82        42

    accuracy                           0.76        62
   macro avg       0.72      0.73      0.73        62
weighted avg       0.76      0.76      0.76        62
```

**Accuracy**

El accuracy obtenido ha sido 0,758 , lo que significa que el modelo acierta aproximadamente el 76% de los casos del conjunto de test. Para un dataset pequeño (310 filas), es un rendimiento razonable para un primer modelo.

**Matriz de confusión**

Según la matriz de confusión, el modelo predice mejor la clase “Abnormal” (34 correctas) frente a la clase “Normal” (13 correctas) probablemente porque es la clase mayoritaria en el dataset.

**Métricas por clase**

- **Precision:** La mayoría de las predicciones de la clase “Abnormal” son correctas (0,83), mientras que la clase “Normal” falla más de lo deseado (0,62).
- **Recall:** El modelo detecta con mayor eficacia la clase “Abnormal”(0,81) frente a la clase “Normal” (0,65)
- **F1-Score:** Buen equilibrio para la clase Abnormal (0.82) y mayores dificultades para la clase Normal (0,63).

## Regresión logística

```
Accuracy: 0.823

Matriz de confusión:
[[14  6]
 [ 5 37]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.74      0.70      0.72        20
           1       0.86      0.88      0.87        42

    accuracy                           0.82        62
   macro avg       0.80      0.79      0.79        62
weighted avg       0.82      0.82      0.82        62
```

**Accuracy**

El accuracy obtenido ha sido 0,823 , lo que significa que el modelo acierta aproximadamente el 82%, ligeramente mejor que Random Forest. Para un dataset pequeño, esto indica que la regresión logística captura bien el patrón entre las variables y la clase.

**Matriz de confusión**

Según la matriz de confusión, el modelo predice mejor la clase “Abnormal” (37correctas) frente a la clase “Normal” (14 correctas), en este caso el modelo mejora en la predicción de las dos clases.

**Métricas por clase**

- **Precision:** La mayoría de las predicciones de la clase “Abnormal” son correctas (0,86),además la clase “Normal” mejora frente al modelo anterior (0,74).
- **Recall:** El modelo detecta con mayor eficacia la clase “Abnormal”(0,88) frente a la clase “Normal” (0,70), igual que en Random Forest.
- **F1-Score:** Excelente equilibrio para la clase Abnormal (0.87) y mejoría para la clase Normal (0,63).

# Conclusión

El proyecto evidencia que es posible predecir la presencia de alteraciones lumbares que provoquen a su vez dolor lumbar, a través de medidas biomecánicas. La regresión logística, por su simplicidad y capacidad de interpretación, parece ser más adecuada para aplicaciones clínicas donde la predicción es la clave para el éxito del modelo. Sin embargo, Random Forest sigue siendo útil para interpretar interacciones más complejas y evaluar la relevancia de las variables.

Como posible mejora, un dataset más amplio con mayor número de observaciones, especialmente de la clase “Normal”, podría mejorar la capacidad de predicción y permitir desarrollar un modelo predictivo de mayor calidad.

Este enfoque combina conocimiento clínico y técnicas de Machine Learning, proporcionando una herramienta potencial para la identificación temprana de pacientes con riesgo de dolor lumbar y apoyando la toma de decisiones del fisioterapeuta.