ITSRLL
INSTITUTO TECNOLÓGICO SUPERIOR DE LA REGIÓN DE LOS LLANOS

Ingeniería Mecatrónica

PROGRAMACIÓN AVANZADA

Enero – Junio 2025

M.C. Osbaldo Aragón Banderas

UNIDAD:

1	**2**	3	4	5

Actividad número: 3

Nombre de actividad:

NOTEBOOK: Análisis de Datos Aplicables al Teorema de Naïve Bayes

Actividad realizada por:

Melissa Gómez Rentería.

Guadalupe Victoria, Durango

Fecha de entrega:

07	03	2025

¿QUÉ ES EL TEOREMA DE NAIVE BAYES?

El Teorema de Bayes es una regla matemática que nos permite calcular la probabilidad de que algo ocurra basándonos en información previa. En términos simples, nos ayuda a actualizar nuestras creencias cuando obtenemos nueva información.

Por ejemplo, si se escucha que va a llover mañana. Antes de revisar el clima, surge una idea general de qué tan probable es que llueva basándose en la época del año. Sin embargo, si luego logra apreciarse que el cielo está nublado y el viento está fuerte, esa información nueva puede hacer pensar que la lluvia es aún más probable. Eso es, en esencia, lo que hace el Teorema de Bayes, ajustar la probabilidad de un evento con base en nueva evidencia.

ECUACIÓN GENERAL DE NAÏVE BAYES

La fórmula matemática es la siguiente:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

- P(A | B): es la probabilidad posterior, es decir, qué tan probable es que ocurra A después de haber observado B.
- P(B | A) es la verosimilitud, es decir, qué tan probable es observar B si A es cierto.
- P(A) es la probabilidad previa, es decir, qué tan probable era A antes de considerar B.
- P(B) es la probabilidad total de que ocurra B, sin importar si A ocurre o no.

Ahora, el clasificador Naïve Bayes se basa en este teorema para hacer predicciones, pero con una suposición clave: asume que todas las características son independientes entre sí. Esto no siempre es cierto en la vida real, pero simplifica mucho los cálculos y hace que el modelo sea rápido y eficiente.

La ecuación general del Naïve Bayes se expresa así:

$$P(C|X_1, X_2, ..., X_n) = P(C) \cdot \prod_{i=1}^{n} P(X_i|C)$$

¿Qué significa cada parte de esta ecuación?

$P(C \mid X_1, X_2, ..., X_n)$ → La probabilidad de que un dato pertenezca a una categoría específica dado un conjunto de características observadas. Por ejemplo, si estamos clasificando correos, esta sería la probabilidad de que un correo sea spam basándonos en palabras clave y otros factores.

$P(C)$ → La probabilidad previa de la categoría. Si estamos clasificando enfermedades, esto sería qué tan común es esa enfermedad en la población en general.

$P(X_i \mid C)$ → La probabilidad de observar cada característica X_i si estamos en la categoría C. Por ejemplo, si una persona tiene fiebre, ¿qué tan probable es que tenga gripe?

Para clasificar un nuevo dato, el modelo calcula esta probabilidad para cada posible categoría y elige la que tenga el mayor valor.

CASOS DE USO REALES

Este modelo es muy eficiente en muchos problemas del mundo real porque es rápido, fácil de implementar y no requiere tantos datos como otros algoritmos más complejos.

1. Filtrado de spam en correos electrónicos

Servicios como Gmail usan Naïve Bayes para clasificar automáticamente los correos como spam o no spam. Analizan palabras clave, la dirección del remitente y otros factores para calcular la probabilidad de que un correo sea spam.

2. Diagnóstico médico

Se usa en hospitales para predecir la probabilidad de que un paciente tenga una enfermedad basándose en síntomas y antecedentes médicos. Por ejemplo, puede ayudar a diagnosticar diabetes analizando factores como nivel de azúcar en sangre, presión arterial y peso.

3. Clasificación de opiniones en redes sociales

Empresas de marketing analizan comentarios y reseñas de productos para detectar si son positivos, negativos o neutrales. Esto se usa en redes sociales como Twitter o en plataformas de reseñas como Amazon.

4. Sistemas de recomendación

Naïve Bayes ayuda a predecir qué productos o películas podrían interesarte basándose en tus preferencias pasadas. Por ejemplo, Netflix puede clasificar géneros de películas que podrían gustarte según las que has visto antes.

5. Detección de fraudes en transacciones bancarias

Los bancos utilizan este modelo para detectar actividades sospechosas en cuentas bancarias. Si una compra parece inusual con respecto al historial del usuario, el banco puede marcar la transacción como potencialmente fraudulenta.

ELECCIÓN DE DATASET (DIAGNÓSTICO DE SOP)

El dataset tiene una variable objetivo bien definida, que es la columna "PCOS (Y/N)":

- 1: significa que la paciente tiene SOP.
- 0: significa que no tiene SOP.

El Naïve Bayes es un modelo de clasificación que funciona muy bien cuando hay que decidir entre dos categorías basándose en varias características. En este caso, nos ayuda a predecir si una paciente tiene SOP o no, basándose en sus datos médicos.

El dataset tiene varias características médicas relevantes, para hacer un buen diagnóstico, se necesita información sobre varios factores de salud. Este dataset incluye características como:

- Edad de la paciente
- Índice de Masa Corporal (IMC)
- Niveles hormonales (testosterona)
- Frecuencia del ciclo menstrual
- Historial de fertilidad

Tener varias características hace que el modelo pueda encontrar patrones y mejorar la precisión del diagnóstico.

El modelo de Naïve Bayes se usa mucho en el campo de la medicina porque es fácil de entrenar y rápido pues no necesita muchos datos ni mucho poder de cómputo, además que maneja bien diferentes tipos de datos y puede trabajar con números (como el IMC) y con valores categóricos (como "Sí/No" en historial menstrual).

A diferencia de modelos más complejos, como redes neuronales, Naïve Bayes permite ver claramente qué factores influyen más en la predicción.

Algunos modelos avanzados, como redes neuronales, requieren mucha potencia de procesamiento y grandes volúmenes de datos para funcionar bien.

En cambio, Naïve Bayes puede entrenarse con menos datos y sigue siendo preciso.

Este dataset no es demasiado grande ni demasiado pequeño, lo que lo hace perfecto para este tipo de modelo.

LINK DE CONJUNTOS DE DATOS

https://www.kaggle.com/datasets/samikshadalvi/pcos-diagnosis-dataset

PREPROCESAMIENTO DE DATOS

Para comenzar a trabajar con los datos, es necesario subir el dataset con el conjunto de datos a trabajar, luego de ello, estos mismo tienen que pasar por un análisis, para lograr conocer si es necesario una limpieza de datos para el dataset, esto puede llegar a ocurrir si se encuentran datos nulos o de tipo que no sea numérico.

```
data=pd.read_csv('pcos_dataset.csv')
print(f'data.shape: {data.shape}')
data.head()
```

```
data.shape: (1000, 6)
```

	Age	BMI	Menstrual_Irregularity	Testosterone_Level(ng/dL)	Antral_Follicle_Count	PCOS_Diagnosis
0	24	35	1	25	20	0
1	37	26	0	57	25	0
2	32	24	0	93	28	0
3	28	29	0	63	26	0
4	25	22	1	60	8	0

Figura 1: Carga del dataset utilizado.

```
[ ]  for col in df:
       print(f"{col}: {df[col].nunique()}")
```

```
Age: 28
BMI: 18
Menstrual_Irregularity: 2
Testosterone_Level(ng/dL): 81
Antral_Follicle_Count: 25
PCOS_Diagnosis: 2
```

```
df.describe(include=[np.number]).T
```

	count	mean	std	min	25%	50%	75%	max
Age	1000.0	31.771	8.463462	18.0	24.0	32.0	39.00	45.0
BMI	1000.0	26.452	4.960272	18.0	22.0	26.0	31.00	35.0
Menstrual_Irregularity	1000.0	0.520	0.499850	0.0	0.0	1.0	1.00	1.0
Testosterone_Level(ng/dL)	1000.0	60.216	23.175600	20.0	42.0	60.0	80.00	100.0
Antral_Follicle_Count	1000.0	17.469	7.069301	5.0	12.0	18.0	23.25	29.0
PCOS_Diagnosis	1000.0	0.199	0.399448	0.0	0.0	0.0	0.00	1.0

Figura 2: Limpieza de datos.

ANÁLISIS EXPLORATORIO DE DATOS (EDA)

Se utilizan herramientas de graficado para visualizar la distribución de los datos de una forma más suave que un histograma, ya que en lugar de agrupar los datos en bins. KDE emplea una función de suavizado (Kernel) para generar una curva continua.

```
#Revisar el BMI (basado en diagnostico)
import warnings
warnings.filterwarnings('ignore')
font={'fontsize':16, 'fontstyle':'italic', 'backgroundcolor': 'black', 'color':'orange'}
%matplotlib inline
plt.style.use('seaborn-v0_8')
sns.kdeplot(df.loc[df['PCOS_Diagnosis']==0, 'BMI'], label='No diagnosticado', shade=True)
sns.kdeplot(df.loc[df['PCOS_Diagnosis']==1, 'BMI'], label='Diagnosticado', shade=True)
plt.title('SOP diagnosticado basado en el BMI', fontdict=font, pad=15)
plt.xticks(np.arange(5,40,5), rotation=90)
plt.xlim([5,40])
plt.legend()
plt.show()
```

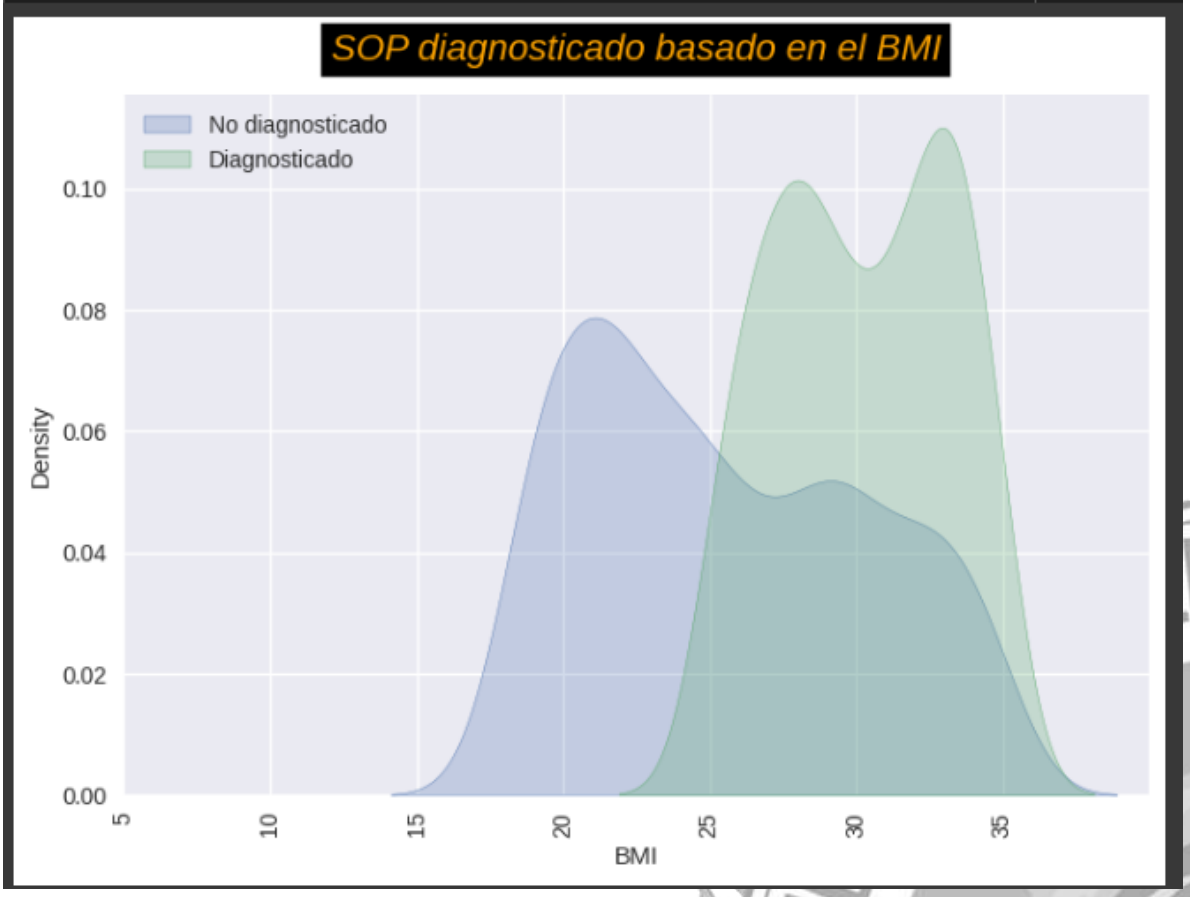

Figura 3: Primer análisis de datos.

```
%matplotlib inline
plt.style.use('seaborn-v0_8')
sns.kdeplot(df.loc[df['PCOS_Diagnosis']==0, 'Age'], label='No diagnosticado', shade=True)
sns.kdeplot(df.loc[df['PCOS_Diagnosis']==1, 'Age'], label='Diagnosticado', shade=True)
plt.title('KDE en edad (Basado en diagnosticos)', fontdict=font, pad=15)
plt.xticks(np.arange(0,60,5))
plt.xlim([10,60])
plt.legend()
plt.show()
```

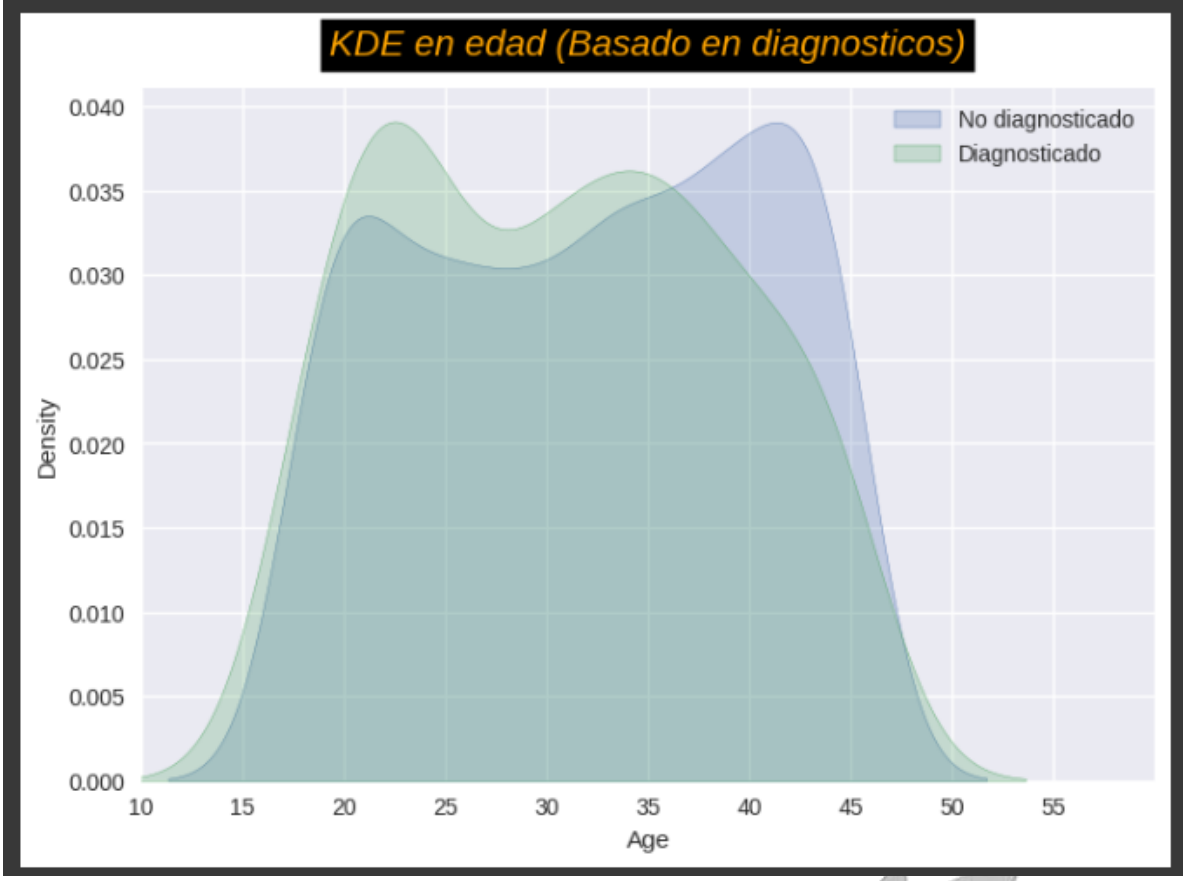

Figura 4: KDE de datos.

ANÁLISIS DE DATOS

Utilizando diversos recursos de graficado, se realizan análisis de los datos con los que se cuentan, siendo los más importantes los que se muestran, dentro del ejercicio se aplicaron análisis de datos univariables, bivariables y multivariables.

```
%matplotlib inline
fig, axes = plt.subplots(1,2, figsize=(10,4))
sns.countplot(data=df, x='PCOS_Diagnosis', ax=axes[0])
for container in axes[0].containers:
  axes[0].bar_label(container)
slices = df.PCOS_Diagnosis.value_counts().values
activities = ['No Diagnosticado (0)', 'Diagnosticado (1)']
axes[1].pie(slices, labels=activities, colors=['blue', 'orange'], shadow=True, explode=[0,0.05], autopct='%1.1f%%')
plt.suptitle('Conteo de diagnósticos', y=1.09, **font)
plt.show()
```

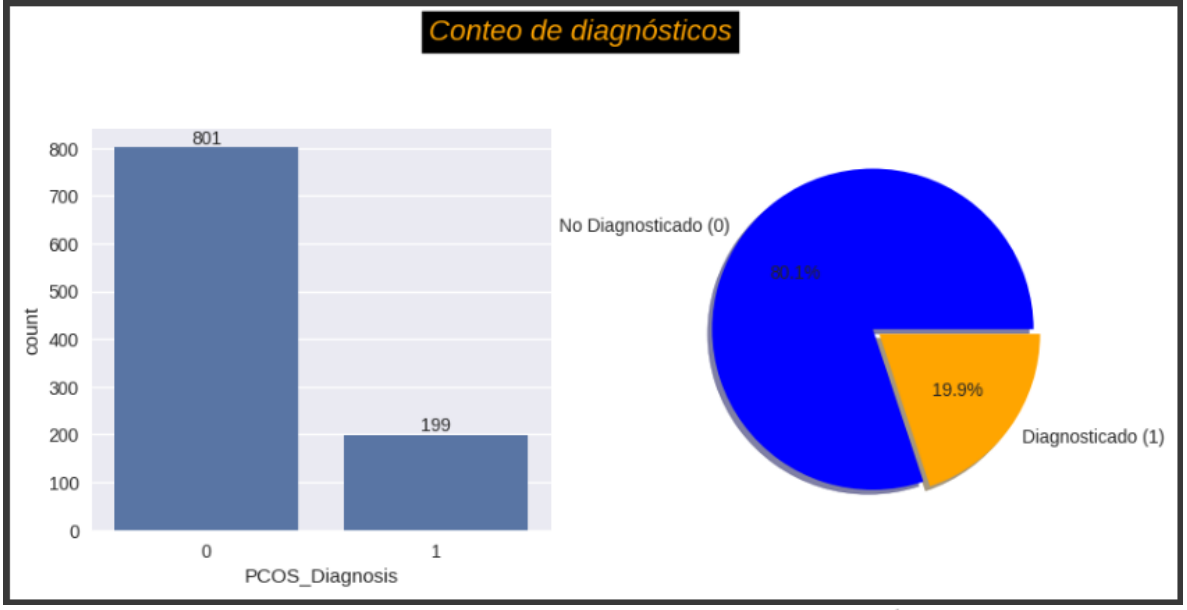

Figura 5: Análisis univariable de diagnósticos.

```
%matplotlib inline
sns.heatmap(df.corr(), cmap='Reds', annot=True)
plt.suptitle('Conteo de diagnósticos', y=1.09, x=0.35, **font)
plt.show()
```

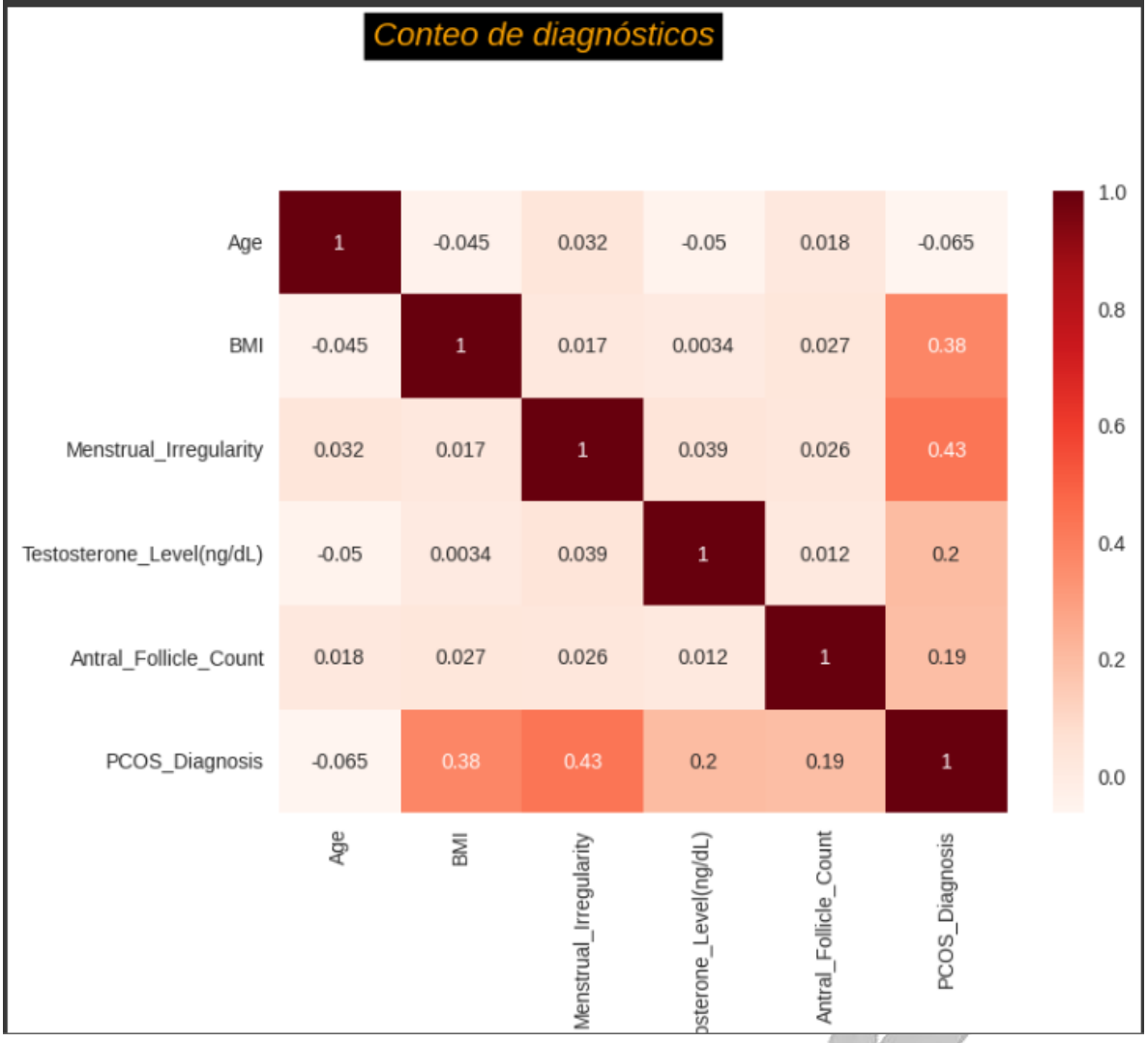

Figura 6: Análisis bivariable de datos.

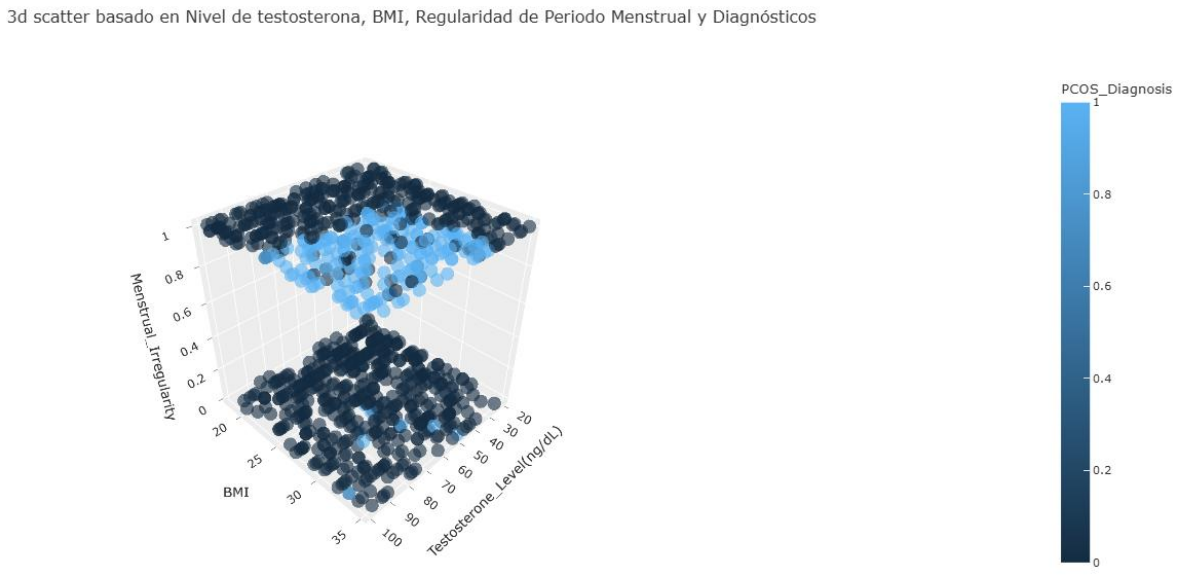

Figura 7: Análisis multivariable de datos.

ENTRENAMIENTO Y PRUEBAS DEL MODELO

Al poner a prueba diversas divisiones de conjuntos de entrenamiento, se llegó a la conclusión, que los valores óptimos para el modelo son un 84% de los datos para entrenamiento, mientras que el 16% de los datos restantes se toman para pruebas.

```
[ ] #Definimos características y etiquetas
    X = df[['BMI', 'Testosterone_Level(ng/dL)', 'Antral_Follicle_Count','Age']]
    y = df['PCOS_Diagnosis']

    #Dividimos los datos en conjuntos de entrenamiento (84%) y prueba (16%)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.16, random_state = 42)
```

Entrenamiento del modelo de Naive Bayes

```
#Creamos el modelo
modelo = GaussianNB()

#Entrenamos el modelo con los datos
modelo.fit(X_train, y_train)
```

```
GaussianNB
GaussianNB()
```

Figura 8: División de datos para entrenamiento y pruebas del modelo.

PREDICCIONES Y EVALUACIÓN DEL MODELO

Una vez que se llevó a cabo la división de datos, se realiza el testeo por parte del modelo, en el cual se desarrollan las predicciones en el conjunto de pruebas, obteniendo la exactitud del modelo, en el cual se obtuvo un valor de **0.843,** dicho valor corresponde a un 80% de precisión en sus predicciones, por lo que puede considerarse que el modelo es BUENO.

```
[ ]  #Realizamos predicciones en el conjunto de pruebas
     y_pred = modelo.predict(X_test)

     #Calculamos la exactitud
     accuracy = accuracy_score(y_test, y_pred)
     print(f'Exactitud del modelo: {accuracy}')

     #Matriz de confuion
     conf_matrix = confusion_matrix(y_test, y_pred)
     print('Matriz de confusion: \n', conf_matrix)

     #Reporte de clasificacion
     print('Reporte de clasificacion: \n', classification_report(y_test, y_pred))
```

```
Exactitud del modelo: 0.84375
Matriz de confusion:
 [[122   8]
 [ 17  13]]
Reporte de clasificacion:
              precision    recall  f1-score   support

           0       0.88      0.94      0.91       130
           1       0.62      0.43      0.51        30

    accuracy                           0.84       160
   macro avg       0.75      0.69      0.71       160
weighted avg       0.83      0.84      0.83       160
```

- LA EXACTITUD DEL MODELO ESTÁ SOBRE 80%, ES UN **BUEN** MODELO.

Figura 9: Predicción, evaluación y cálculo de métricas del modelo.

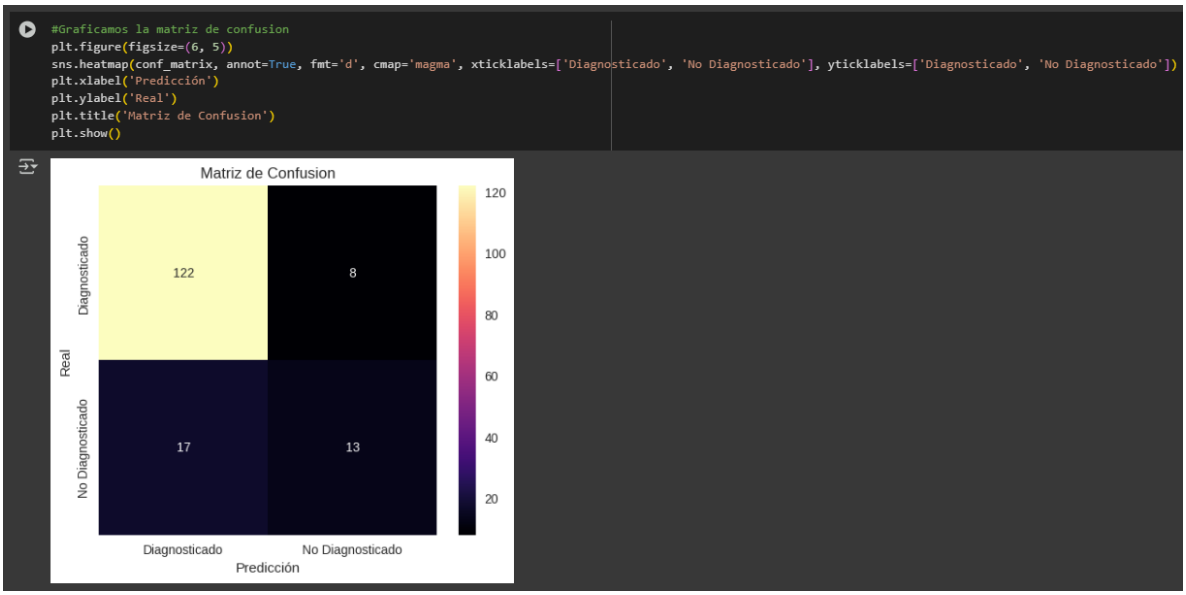

Figura 10: Matriz de confusión del modelo.

Una vez interpretados los datos que el modelo arroja, podemos llegar a la conclusión de que el modelo para el diagnóstico de Síndrome de Ovario Poliquístico tiene como resultados:

- Verdaderos Positivos (TP): **122** (Casos **correctamente diagnosticados** como positivos)
- Falsos Negativos (FN): **8** (Casos diagnosticados como negativos, pero que realmente eran positivos)
- Falsos Positivos (FP): **17** (Casos diagnosticados como positivos, pero que realmente eran negativos)
- Verdaderos Negativos (TN): **13** (Casos **correctamente diagnosticados** como negativos)