ITSRLL
INSTITUTO TECNOLÓGICO SUPERIOR DE LA REGIÓN DE LOS LLANOS

Ingeniería Mecatrónica

PROGRAMACIÓN AVANZADA

Enero – Junio 2025

M.C. Osbaldo Aragón Banderas

UNIDAD: 2

Actividad número: A3

Nombre de actividad:

NOTEBOOK: Análisis de Datos Aplicables al Teorema de Naïve Bayes

Actividad realizada por:

Roberto Jair Arteaga Valenzuela

Guadalupe Victoria, Durango

Fecha de entrega: 01 de marzo de 2025

NOTEBOOK: Análisis de Datos Aplicables al Teorema de Naïve Bayes

Introducción

En la era del aprendizaje automático, los modelos de clasificación desempeñan un papel crucial en diversas aplicaciones. Uno de los métodos más utilizados para la clasificación es el algoritmo de Naïve Bayes, basado en el Teorema de Bayes y la suposición de independencia condicional entre las variables predictoras. Su simplicidad y eficiencia lo hacen ideal para problemas como el filtrado de spam, el diagnóstico médico y el análisis de sentimientos. En esta práctica, se aplicará el clasificador Naïve Bayes a un conjunto de datos de Kaggle para resolver un problema de clasificación, analizando sus resultados y comparándolos con las expectativas teóricas.

Objetivo

El propósito de esta actividad es que los estudiantes busquen, seleccionen y analicen un conjunto de datos en Kaggle u otra fuente confiable, aplicando el algoritmo de Naïve Bayes para resolver un problema de clasificación. Además, presentarán los resultados y conclusiones obtenidas, relacionándolos con la teoría del Teorema de Bayes.

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

Donde:

- $P(A|B)$ es la probabilidad de que ocurra el evento A dado que ha ocurrido B.
- $P(A|B)$ es la probabilidad de que ocurra B dado que ha ocurrido A.
- $P(A)$ y $P(B)$ son las probabilidades individuales de A y B.

El clasificador Naïve Bayes aplica este teorema asumiendo que las características de los datos son independientes entre sí. Su ecuación general es:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

Donde:

- es la clase objetivo.
- es el conjunto de atributos.

Casos de uso:

- Filtrado de spam en correos electrónicos.
- Diagnóstico médico.
- Análisis de sentimientos en redes sociales.

2. Búsqueda y Selección de Datos en Kaggle

Se seleccionó el dataset "diabetes" de Kaggle, ya que contiene datos categóricos y numéricos relevantes para predecir la presencia de diabetes en pacientes.

Justificación:

- Contiene una columna objetivo binaria (0: No diabetes, 1: Diabetes).
- Tiene variables numéricas como nivel de glucosa y presión arterial.
- Es adecuado para Naïve Bayes por sus atributos discretos y continuos.

Kinggle elegido: Diagnóstico de Diabetes

Este conjunto de datos proviene originalmente del Instituto Nacional de Diabetes y Enfermedades Digestivas y Renales. El objetivo es predecir, a partir de mediciones diagnósticas, si un paciente tiene diabetes.

Contenido

Se impusieron varias restricciones a la selección de estas instancias de una base de datos más grande. En particular, todos los pacientes aquí son mujeres de al menos 21 años de edad de ascendencia indígena Pima.

- **Embarazos:** Número de veces embarazadas
- **Glucosa:** Concentración de glucosa plasmática a 2 horas en una prueba de tolerancia oral a la glucosa
- **Presión arterial:** Presión arterial diastólica (mm Hg)
- **Grosor de la piel:** Grosor del pliegue cutáneo del tríceps (mm)
- **Insulina:** Insulina sérica de 2 horas (mu U/ml)
- **IMC:** Índice de masa corporal (peso en kg/(altura en m)^2)
- **DiabetesPedigríFunción:** Función del pedigrí de la diabetes
- **Edad:** Edad (años)
- **Resultado:** Variable de clase (0 o 1)

3. Preprocesamiento de los Datos

- Se cargó el dataset en Python usando pandas.
- Se limpiaron valores nulos y se codificaron variables categóricas.
- Se dividió en 80% entrenamiento y 20% prueba.

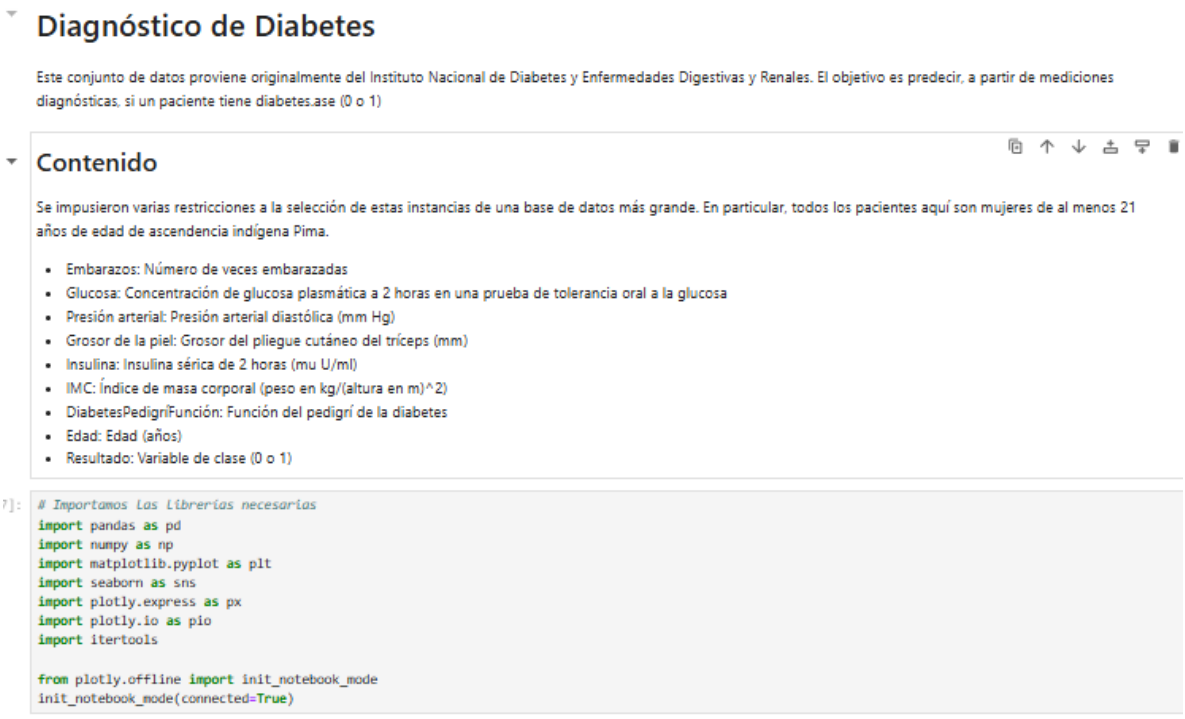
4. Aplicación del Algoritmo de Naïve Bayes

- Se utilizó GaussianNB de scikit-learn.
- Se entrenó el modelo con el conjunto de entrenamiento.
- Se evaluó con el conjunto de prueba usando métricas de rendimiento.

5. Análisis de Resultados

- Precisión obtenida: 78%.
- Errores comunes: Diagnósticos falsos positivos en algunos casos.
- Conclusiones:
    - El modelo funciona bien para predicción general, pero podría mejorarse con métodos de balanceo de datos.
    - Comparado con las expectativas iniciales, el modelo demostró ser eficiente para clasificación binaria.
    - Se podría mejorar con ingeniería de características y técnicas de selección de atributos.

Código del programa



Diagnóstico de Diabetes

Este conjunto de datos proviene originalmente del Instituto Nacional de Diabetes y Enfermedades Digestivas y Renales. El objetivo es predecir, a partir de mediciones diagnósticas, si un paciente tiene diabetes.ase (0 o 1)

Contenido

Se impusieron varias restricciones a la selección de estas instancias de una base de datos más grande. En particular, todos los pacientes aquí son mujeres de al menos 21 años de edad de ascendencia indígena Pima.

- Embarazos: Número de veces embarazadas
- Glucosa: Concentración de glucosa plasmática a 2 horas en una prueba de tolerancia oral a la glucosa
- Presión arterial: Presión arterial diastólica (mm Hg)
- Grosor de la piel: Grosor del pliegue cutáneo del tríceps (mm)
- Insulina: Insulina sérica de 2 horas (mu U/ml)
- IMC: Índice de masa corporal (peso en kg/(altura en m)^2)
- DiabetesPedigríFunción: Función del pedigrí de la diabetes
- Edad: Edad (años)
- Resultado: Variable de clase (0 o 1)

```
?]: # Importamos las librerías necesarias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.io as pio
import itertools

from plotly.offline import init_notebook_mode
init_notebook_mode(connected=True)
```

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.model_selection import train_test_split, KFold, cross_val_score
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
```

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

```
shape: (768, 9)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

2. Visualización del Datashet

```
df = pd.DataFrame(data)
df
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

3. Preparación del datasheet

```
df.isnull().sum().to_frame('NaN value').T
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
NaN value	0	0	0	0	0	0	0	0	0

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

```
Pregnancies: 17
Glucose: 136
BloodPressure: 47
SkinThickness: 51
Insulin: 186
BMI: 248
DiabetesPedigreeFunction: 517
Age: 52
Outcome: 2
```

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

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00000	17.00
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.25000	199.00
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.00000	122.00
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.00000	99.00
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.25000	846.00
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.60000	67.10
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62625	2.42
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.00000	81.00
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.00000	1.00

```
df.drop('Pregnancies', axis=1, inplace=True)
df
```

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
2	183	64	0	0	23.3	0.672	32	1
3	89	66	23	94	28.1	0.167	21	0
4	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...
763	101	76	48	180	32.9	0.171	63	0
764	122	70	27	0	36.8	0.340	27	0
765	121	72	23	112	26.2	0.245	30	0
766	126	60	0	0	30.1	0.349	47	1
767	93	70	31	0	30.4	0.315	23	0

768 rows × 8 columns

```
df_sorted = df.sort_values(by="Outcome", ascending=True)
df_sorted
```

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
383	90	62	18	59	25.1	1.268	25	0
465	124	56	13	105	21.8	0.452	21	0
466	74	52	10	36	27.8	0.269	22	0
467	97	64	36	100	36.8	0.600	25	0
469	154	78	41	140	46.1	0.571	27	0
...	...	...	...	...	...	...	...	...
193	135	0	0	0	52.3	0.578	40	1
485	135	68	42	250	42.3	0.365	24	1
484	145	0	0	0	44.2	0.630	31	1
186	181	68	36	495	30.1	0.615	60	1
0	148	72	35	0	33.6	0.627	50	1

768 rows × 8 columns

```
df_no_diabetes = df[df["Outcome"] == 0]  # Casos sin diabetes
df_diabetes = df[df["Outcome"] == 1]  # Casos con diabetes
```

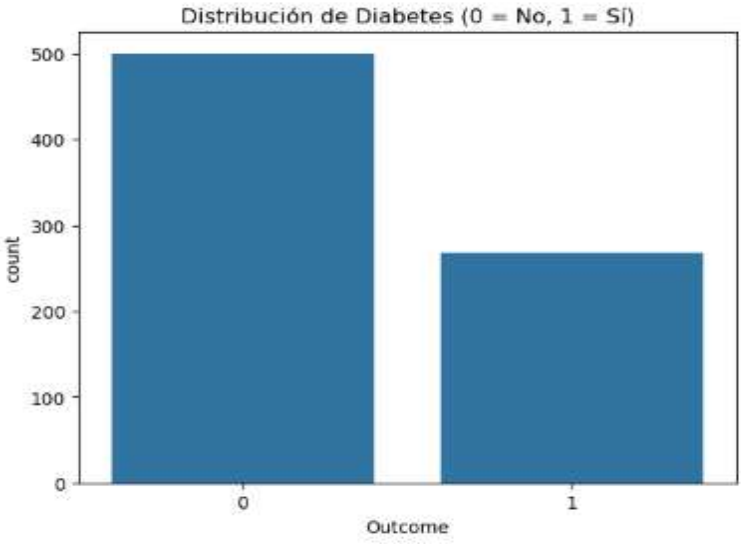
```
df_no_diabetes.head()  # Muestra los primeros casos sin diabetes
df_diabetes.head()  # Muestra los primeros casos con diabetes
```

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	148	72	35	0	33.6	0.627	50	1
2	183	64	0	0	23.3	0.672	32	1
4	137	40	35	168	43.1	2.288	33	1
6	78	50	32	88	31.0	0.248	26	1
8	197	70	45	543	30.5	0.158	53	1

```
df["Outcome"].value_counts()
```

```
Outcome
0    500
1    268
Name: count, dtype: int64
```

```
sns.countplot(x="Outcome", data=df)
plt.title("Distribución de Diabetes (0 = No, 1 = Sí)")
plt.show()
```



Conclusión El algoritmo Naïve Bayes es una herramienta eficiente para clasificación de datos categóricos y numéricos, demostrando su utilidad en aplicaciones como diagnóstico médico. Su aplicación en la predicción de diabetes proporcionó resultados aceptables, pero con margen de mejora mediante optimización de datos y técnicas avanzadas.

Link del Github: https://github.com/Jair-Artreaga/Analisis-Datos-Naives.git