# Objetivos

Desarrollo Taller 2 AI: PCA

Juan Sebastián Bravo Santacruz

### Utilizar el método de Análisis de Componentes Principales (PCA) y analizar su funcionalidad.

# Conjunto de datos data\_3D

Utilizando el conjunto de datos data 3D (datos del punto 2 del Taller 1):

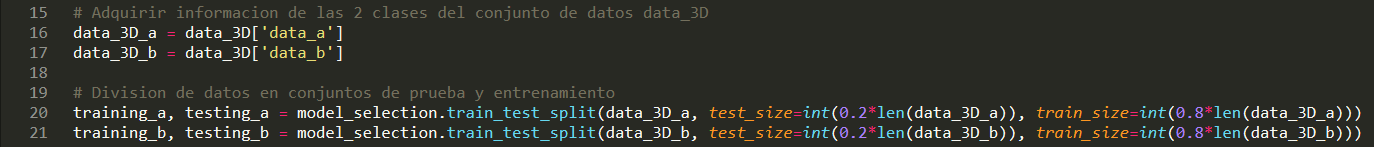
## Determine la transformación de a utilizando PCA (sobre el conjunto de entrenamiento):

Inicialmente se realizó la importación de los módulos numpy, sklearn y matplotlib. Se importaron también los datos del archivo data.npy y se separaron los conjuntos data\_2D y data\_3D en 2 distintas variables:

Text

Description automatically generated

La división de datos se realizó adquiriendo inicialmente los datos de cada una de las clases para el conjunto data\_3D, seguidamente se utilizó la función train\_test\_split() del módulo sklearn como se muestra en la imagen:



En este, para el parámetro test\_size, se encontró el 20% del número total de datos de cada clase y para el parámetro train\_size se encontró el 80% del número total de datos de cada clase.

Para el proceso de reducción de reducción de dimensión, inicialmente se quitó el valor medio a los conjuntos de entrenamiento tanto para la clase a como para la clase b de los datos:



Una vez obtenidos los conjuntos de entrenamiento centrados en el origen, se calculó la matriz de covarianza de estos y posteriormente se encontraron los valores y vectores propios de las matrices:

Text

Description automatically generated

En este caso, al tener 3 atributos, se espera tener una matriz de covarianza de 3x3.

A screenshot of a computer

Description automatically generated with medium confidence

Figura 1. Matriz de covarianza clase a.

A screenshot of a computer

Description automatically generated with medium confidence

Figura 2. Matriz de covarianza clase b.

Por otra parte, se tienen 3 vectores propios con 3 componentes cada uno ya que se trata de un conjunto de datos de 3 dimensiones y 3 valores propios, uno por cada vector.

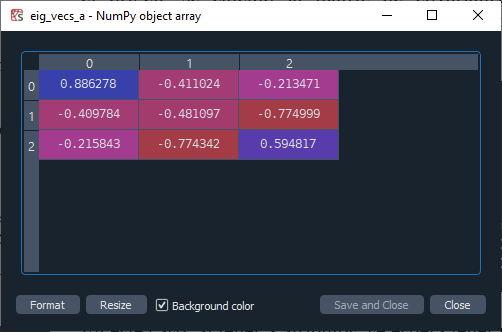


Figura 3. Vectores propios clase a.

A screenshot of a computer

Description automatically generated

Figura 4. Vectores propios clase b.

A screenshot of a computer

Description automatically generated

Figura 5. Valores propios clase a.

A screenshot of a computer

Description automatically generated with medium confidence

Figura 6. Valores propios clase b.

Se conservaron los 2 valores propios más grandes y sus respectivos valores propios para crear los nuevos conjuntos de entrenamiento siguiendo las siguientes ecuaciones:

A picture containing text

Description automatically generated

Resultando así 2 nuevos conjuntos de entrenamiento para cada una de las clases, con igual número de elementos, pero con habiendo reducido los atributos de 3 a 2.

## Implemente un clasificador Bayesiano Gaussiano sobre los datos transformados, y:

1. Estime la función de verosimilitud de cada clase:

Inicialmente para la implantación de la función de verosimilitud presentada en la siguiente ecuación:

Se encontró el número de atributos y las probabilidades a priori de cada una de las clases, en este caso, asumiendo que estaban distribuidas homogéneamente:

Text

Description automatically generated

Posteriormente, se encontraron los valores centrales y las matrices de covarianza para cada una de las clases.

Text

Description automatically generated

Al tener ahora una reducción a 2 atributos se tiene un vector con 2 valores de media, uno por cada atributo.

A screenshot of a computer

Description automatically generated with medium confidence

Figura 7. Vector medias para clase a.

A screenshot of a computer

Description automatically generated with medium confidence

Figura 8. Vector medias para clase a.

Del valor de las medias, se puede observar cómo este es 0, debido a que el conjunto de datos de entrenamiento nuevo, se encuentra ubicado ahora en el origen.

Adicionalmente, se tienen matrices de covarianza de 2x2, debido al número de atributos:

A screenshot of a computer

Description automatically generated

Figura 9. Matriz de covarianza clase a.

A screenshot of a computer

Description automatically generated

Figura 10. Matriz de covarianza clase b.

De las matrices de covarianza, se puede observar que existe una dependencia mínima entre los nuevos atributos del conjunto de datos.

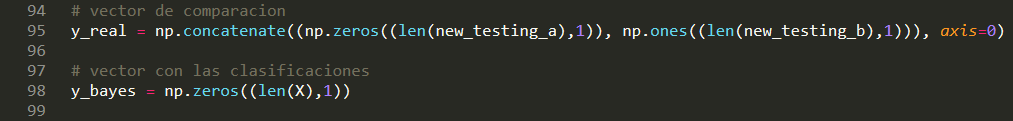
1. Determine la transformación de a utilizando PCA (sobre el conjunto de prueba):

Una vez encontradas cada una de las constantes necesarias para encontrar la función, se creó una matriz de prueba que contenía los conjuntos de prueba de la clase a y b aplicando PCA. Para esto se resto la media encontrada en el conjunto de entrenamiento y se multiplicó por la matriz M’ también encontrada con el conjunto de entrenamiento.

Text

Description automatically generated

Se creó un vector con las etiquetas de clasificación reales para posteriormente calcular el error de la clasificación y finalmente un vector para almacenar la salida de la clasificación a realizar:



Finalmente, se implementó la función, asignando un 0 para la clase a y un 1 para la clase b como se muestra en el siguiente código:

A screenshot of a computer

Description automatically generated with medium confidence

1. Visualice la clasificación realizada sobre el conjunto de prueba:

Para la visualización del conjunto de prueba, inicialmente, se crearon 2 nuevas matrices para cada una de las clases en base a la clasificación realizada y seguidamente se graficó el conjunto haciendo uso de la función scatter ():

Text

Description automatically generated

Adicionalmente, previamente se graficó el conjunto de entrenamiento nuevo para tener noción de la clasificación realizada.

Text

Description automatically generated

Las gráficas obtenidas son las siguientes:

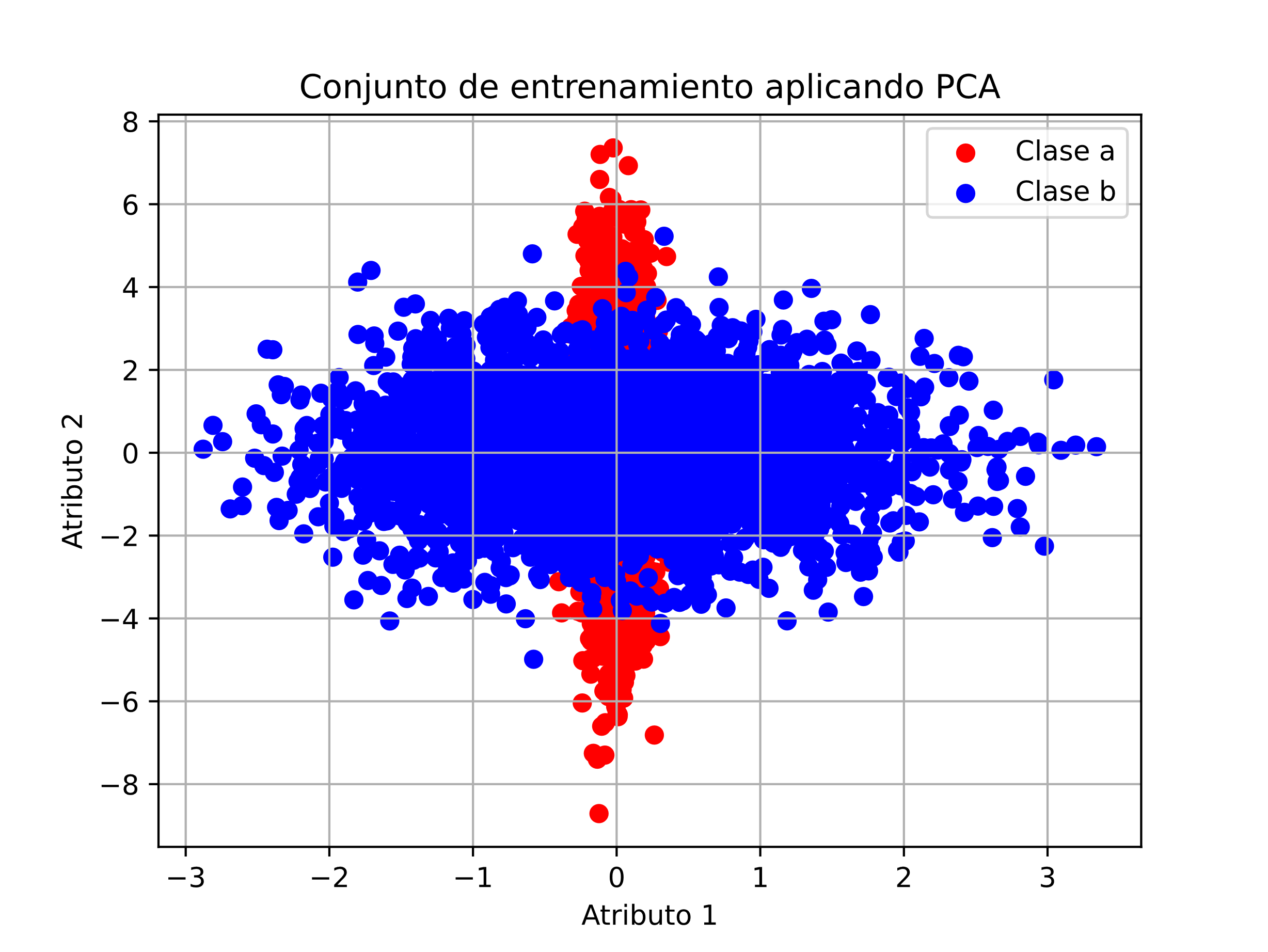


Figura 11. Conjunto de entrenamiento aplicando PCA.

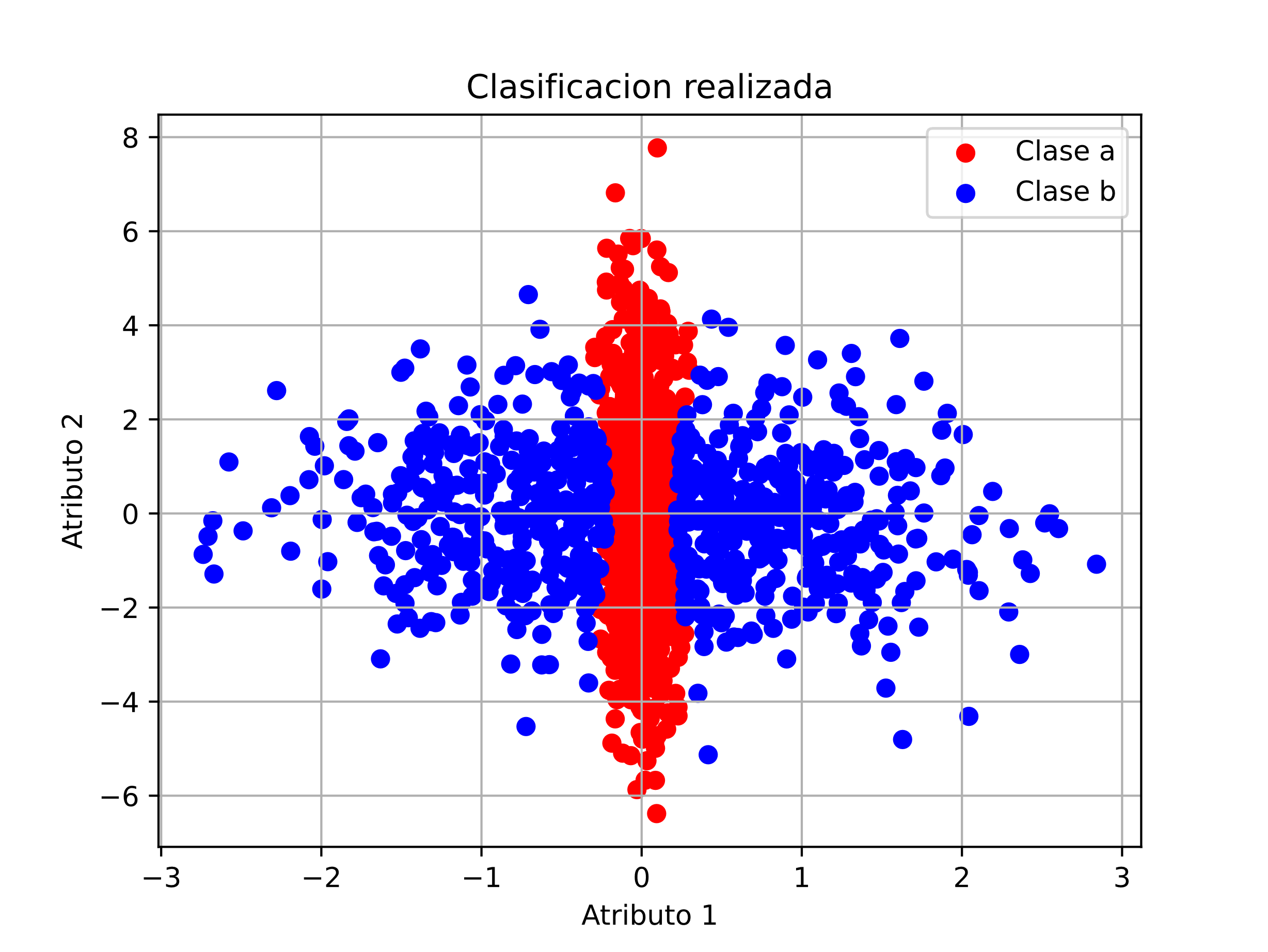
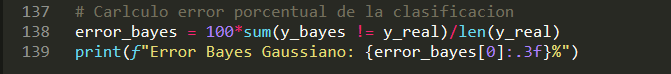


Figura 12. Conjunto de prueba clasificado con Bayes Gaussiano.

De la gráfica de clasificación, se puede apreciar que este realizó una clasificación buena que y solo presenta errores en aquellas secciones donde se superponen las clases.

1. Determine el error de clasificación sobre el conjunto de prueba:

Se determinó el error de clasificación, encontrando la cantidad de clasificaciones mal realizadas dividido el número de elementos clasificados.



Los errores porcentuales obtenidos en 3 ejecuciones del programa fueron:







# Concluya sobre los resultados obtenidos de los clasificadores del Taller 1 con respecto al clasificador desarrollado en este taller.

Para realizar el proceso de comparación de los resultados con los clasificadores del Taller 1, se utilizó el mismo conjunto de entrenamiento para que el proceso de comparación sea justo. Para esto se utilizó Spyder, dejando constante la variable de training para ambos scripts.

Los errores encontrados fueron los siguientes:

|  |  |  |
| --- | --- | --- |
| **Errores porcentuales** | | |
| Bayes Gaussiano | Bayes Gaussiano Naive | Bayes Gaussiano PCA |
| 2.812 % | 7.875 % | 11.625 % |

Partiendo de los errores de clasificación encontrados, se puede decir que si bien es cierto los clasificadores Bayesiano Gaussiano y Bayesiano Gaussiano Naive sin PCA presentan los errores más bajos, el clasificador Bayesiano Gaussiano aplicando la reducción de dimensión, presenta un rendimiento bueno y no muy alejado del Bayesiano Gaussiano Naive sin PCA.