# **LDA utilizando Python y Scikit-learn**

Los algoritmos lineales son de gran utilizar para atacar problemas relativamente sencillos, donde es notaria una relación lineal entre los predictores y la variable objetivo.

En esta ocasión se utilizará el algoritmo de Análisis de Discriminante Lineal para un problema de clasificación binaria. Para ello, se utilizará la librería de Scikit-learn de Python que provee métodos de entrenamiento para un modelo LDA.

Se cuenta con un Dataset simple de únicamente dos variables: X e Y. Es importante destacar que la cantidad de ejemplos no es muy grande, hay solamente 20 registros. La clase por predecir tiene dos posibles valores 0 y 1.

## Importar librerías y lectura de Dataset

Para este proceso se utilizarán las librerías de Scikit-learn, Matplotlib y Pandas. Para evitar complicaciones mayores accederé a un documento de Google Colab y escribiré el código Python en un Jupyter Notebook.

Texto

Descripción generada automáticamente

## Graficar Dataset en dos dimensiones

Al disponer de únicamente dos variables, la linealidad del problema puede verse representada mediante un gráfico de tipo “Plot” de ambos predictores. Esto puede ser realizado mediante la librería Matplotlib, manipulando previamente las columnas al invocar el método de graficar.

Gráfico, Gráfico de burbujas

Descripción generada automáticamente

En el gráfico se ve claramente la separación de ambas clases en dos grupos disjuntos. Por lo tanto, podemos asegurar que la utilización de un modelo lineal para trabajar este Dataset es apropiada.

## Entrenamiento del modelo

Habiendo confirmado la disposición de las clases en el gráfico plot, el siguiente paso implicar entrenar un modelo LDA utilizando el Dataset. Para ello, primero es necesario separar los predictores de la variable objetivo.



El método *train\_test\_split* permite generar cuatro conjuntos a partir del Dataset. Separa los datos en predictores y variable objetivo. Además, aparta algunos datos para utilizar como validación del modelo. En este caso se utilizará el 25% (5 ejemplos) del Dataset para validación.



Para entrenar el modelo LDA se utilizan los métodos importados de scikit-learn.

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

## Validación del modelo

Para validar el entrenamiento del modelo, utilizaremos los ejemplos que mantuvimos fuera del entrenamiento. Es importante tener también los valores reales de la variable objetivo para poder compararlos con las predicciones del modelo.

Texto, Carta

Descripción generada automáticamente

Los resultados muestran que el modelo realizó una predicción correcta en todos los casos de validación. También es de utilidad examinar la matriz de confusión de las pruebas.

Esto se puede realizar mediante el método *confusion\_matrix* de Scikit-learn Metrics.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Cada columna representa las predicciones que el modelo realizó, mientras que las filas los valores reales tomados del Dataset de validación. Si todos los valores de la matriz con ceros a excepción de la diagonal principal, el modelo habrá funcionado perfectamente. En caso contrario nos encontraremos con falsos positivos o falsos negativos, situaciones donde el modelo realizó la predicción incorrecta.