

# Trabajo práctico final - Introducción al Aprendizaje Automatizado.

---

Stizza, Federico.

Para la implementación de las *Support Vector Machines* utilicé la librería [sklearn](#) de Python.

## Ejercicio 1

---

Para optimizar los valores C de las SVM y G (gamma) en el caso del kernel *Gaussiano*, se utilizaron 8 folds de entrenamiento y 1 de validación y el restante fue apartado para calcular el error de test. Se experimentó los siguientes rangos de valores:

- $C \in [1e-05, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000]$
- $G \in [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$

Para elegir el parámetro óptimo use el valor que menor error de validación daba, aunque además probé utilizando el menor promedio entre el error de validación y de test. El segundo retornaba un valor con un error medio un poco mayor pero con menor desvío estándar.

Dichos valores óptimos resultaron:

- $C = 10000$  para el kernel *Lineal*
- $C = 100$  y  $G = 0.1$  para el kernel *Gaussiano*

Luego con los parámetros *óptimos*, se realizaron los experimentos utilizando 9 folds de entrenamiento y 1 de test, alternando el fold de prueba con los otros, de los datos resultantes se calculó la medio y el desvío estándar.

Clasificador	Error medio de test	Desvío estándar
Árboles:	0.2747	0.0693029580898247
Naive Bayes	0.42568	0.09473577055274432
SVM Gaussianas	0.26553426248548206	0.05239821269023245
SVM Lineal	0.3063617886178862	0.04498106140969842

Como resultado tenemos que el clasificador que menor error medio de test presenta es el *SVM Gaussiano* aunque tiene mayor desvío estandar que el *Lineal*.

## Ejercicio 2

---

Ordenando los clasificadores según su error medio de test, tenemos:

- 1º: SVM con kernel Gaussiano
- 2º: Árboles de decisión

- 3º: SVM con kernel lineal
- 4º: Naïve Bayes

Calculamos el t-test:

- 1º - 2º: 0.33576356048243583
- 1º - 4º: 3.698785050010344

Por lo que en el primer caso no podemos descartar la hipótesis nula y concluir que son diferentes, pero en cambio entre el 1º y el 4º si se podemos afirmar con un 95% de confianza que los métodos son distintos ya que el valor resultado es mayor a **2.26**.

Podemos concluir que para el dataset *BBBs* la máquina de vectores soporte combinado con la técnicas de *folding*, genera un buen modelo de predicción, robusto a pesar de la poca cantidad de ejemplos de entrenamiento.