

ML_CLASIFICACION_SVM_01	Modelo de Clasificación SVM	ML
<p>Esta práctica consiste en construir un modelo de aprendizaje maquina (machine learning), basado en el algoritmo de SVM (Support Vector Machines) que aprenda a clasificar una determinada clase, encontrar la función de borde y los vectores soporte.</p> <p>Se utiliza la base de datos 'make_blobs' de la librería 'sklearn.datasets'.</p>		
<p><u>SOLUCIÓN</u></p> <p>Importar las librerías necesarias para realizar la práctica.</p> <pre># Librerías import numpy as np import matplotlib.pyplot as plt from sklearn import svm from sklearn.datasets import make_blobs</pre> <p>Generar X e y.</p> <pre># Generar X e y de la base de datos 'make_blobs' de la librería sklearn.datasets X, y = make_blobs(n_samples=50, centers=2, random_state=6)</pre> <p>Crear el clasificador y ajustarlo</p> <pre># Crear el clasificador SVM con kernel lineal y ajustar clf = svm.SVC(kernel='linear', C=1000) clf.fit(X, y)</pre> <p>Visualizar los puntos</p> <pre># Visualizar los puntos plt.scatter(X[:, 0], X[:, 1], c=y, s=30, cmap=plt.cm.Paired)</pre> <p>Visualizar la función de decisión 'Z' y el margen</p> <pre># Definir el eje y los límites de visualización ax = plt.gca() xlim = ax.get_xlim() ylim = ax.get_ylim() # Crear la rejilla para evaluar el modelo xx = np.linspace(xlim[0], xlim[1], 30) yy = np.linspace(ylim[0], ylim[1], 30) YY, XX = np.meshgrid(yy, xx) xy = np.vstack([XX.ravel(), YY.ravel()]).T</pre>		

```
# Crear la función de decisión 'Z'
Z = clf.decision_function(xy).reshape(XX.shape)

# Visualizar el borde de decisión y los márgenes
ax.contour(XX, YY, Z, colors='k', levels=[-1, 0, 1], alpha=0.5,
           linestyles=['--', '-', '--'])

# Visualizar los vectores soporte (support vectors)
ax.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1], s=100,
           linewidth=1, facecolors='none', edgecolors='k')
```

Visualizar el gráfico

plt.show()

