



Profundiza más

Recurso de Profundización

Clase 6 - Implementación en Python de carga y visualización de datos que no son linealmente separables

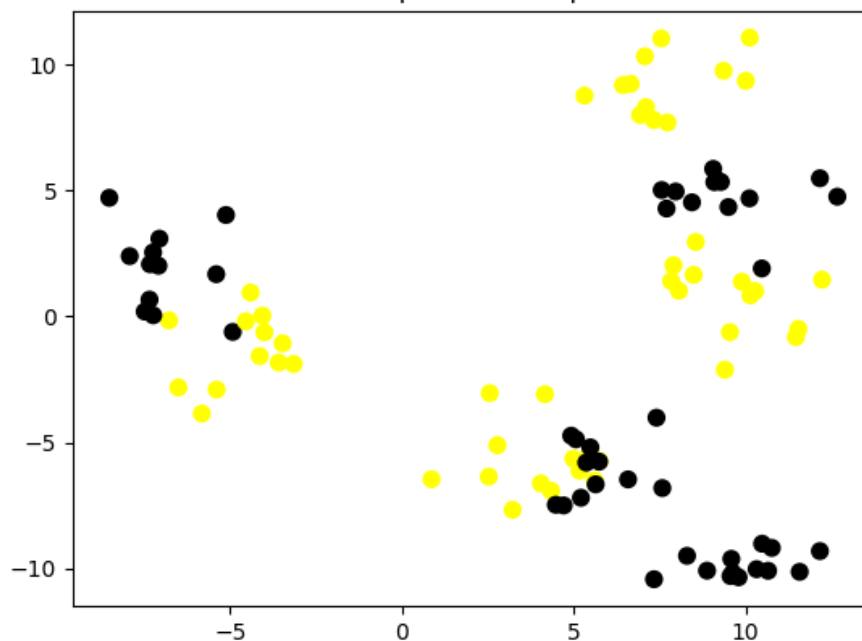
```
# Cargamos librerías
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from matplotlib.colors import ListedColormap

cmap_bold = ListedColormap(['#FFFF00', '#00FF00', '#0000FF', '#000000'])

# Cargamos los datos
X_D2, y_D2 = make_blobs(n_samples = 100, n_features = 2,
                        centers = 8, cluster_std = 1.3,
                        random_state = 4)
y_D2 = y_D2 % 2

# Graficamos
plt.figure()
plt.title('Problema de la clasificación binaria para clases que no son linealmente separables')
plt.scatter(X_D2[:,0], X_D2[:,1], c=y_D2,
            marker= 'o', s=50, cmap=cmap_bold)
plt.show()
```

Problema de la clasificación binaria para clases que no son linealmente separables





Profundiza más

Implementación de RandomForestClassifier en *Scikit-Learn*

```
# Importamos librerías
from sklearn.datasets import make_blobs
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

# Cargamos los datos
X_D2, y_D2 = make_blobs(n_samples = 100, n_features = 2,
                        centers = 8, cluster_std = 1.3,
                        random_state = 4)
y_D2 = y_D2 % 2

# Separamos en train y test
X_train, X_test, y_train, y_test = train_test_split(X_D2, y_D2,
                                                    random_state = 0)

# Instanciamos el clasificador
clf = RandomForestClassifier().fit(X_train, y_train)

print('score train:', clf.score(X_train, y_train), 'score test: ', clf.score(X_test, y_test))
```