

Tema: Introducción a Scikit-Learn

En esta notebook, se ejemplifican algunas funcionalidades básicas de Scikit-Learn usando el dataset "Iris". Los datos se cargan como un DataFrame mediante un método de la biblioteca seaborn y se clasifican con el método knn. En la segunda versión de esta notebook, los datos se cargan como un objeto Bunch y se muestran otras funcionalidades y métodos de aprendizaje adicionales

Ejemplos

In [1]:

```
import seaborn as sns
iris = sns.load_dataset('iris')
iris.head()
```

Out[1]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [2]:

```
iris.shape
```

Out[2]:

(150, 5)

In [3]:

```
X_iris = iris.drop('species', axis=1)
X_iris.shape
```

Out[3]:

(150, 4)

In [4]:

```
y_iris = iris['species']
y_iris.shape
```

Out[4]:

(150,)

In [5]:

```
from sklearn.neighbors import KNeighborsClassifier
```

In [6]:

```
model = KNeighborsClassifier(n_neighbors=15)
```

In [7]:

```
model.fit(X_iris, y_iris)
```

Out[7]:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',  
                    metric_params=None, n_jobs=1, n_neighbors=15, p=2,  
                    weights='uniform')
```

In [8]:

```
y_model = model.predict(X_iris)  
y_model.shape
```

Out[8]:

```
(150,)
```

In [9]:

```
from sklearn.metrics import accuracy_score  
accuracy_score(y_iris, y_model)
```

Out[9]:

```
0.9866666666666667
```

In [10]:

```
from sklearn.model_selection import train_test_split  
Xtrain, Xtest, ytrain, ytest = train_test_split(X_iris, y_iris, random_state=1)
```

In [11]:

```
model.fit(Xtrain, ytrain)  
y_model = model.predict(Xtest)  
accuracy_score(ytest, y_model)
```

Out[11]:

```
0.9736842105263158
```