Valorar características individuales

Otra opción para reducir los datos que utilizaremos será elegir sólo alguna de las caracteríistias originales sin cambiarlas. De esta forma, se mantiene la interpretabilidad de los modelos, ya que las características que se utilicen seguirán ahí, al contrario que con el análisis de componentes principales donde las características nuevas son una ecuación de las originales.

Para poder elegir con qué características quedarnos, una primera idea puede ser evaluarlas. A ver cuales de ellas nos pueden ayudar más a nuestro problema de aprendizaje.

Si tenemos un problema de clasificación podemos usar la medida `chi2` (https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html#sklearn.feature_selection.chi2):

In [5]:
```
import numpy as np
import pandas as pd
iris = pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data",
                   names=['sepal length','sepal width','petal length','petal width','target'])
caracteristicas = ['sepal length', 'sepal width', 'petal length', 'petal width']
iris
```

Out[5]:

	sepal length	sepal width	petal length	petal width	target
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [6]:
```
from sklearn.feature_selection import chi2
X = iris[caracteristicas]
y = iris['target']
chi2(X, y)
```

Out[6]:
```
(array([ 10.81782088,   3.59449902, 116.16984746,  67.24482759]),
 array([4.47651499e-03, 1.65754167e-01, 5.94344354e-26, 2.50017968e-15]))
```

Los valores de la primera fila son los valores del estadístico. Cuanto mayores, más se puede asegurar que la clase no es independiente de esa característica. Por tanto, aquellas características con valores más altos deben ser las elegidas.

Si tenemos un problema de regresión (variable dependiente continua), podemos usar `f_regression` (https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_regression.html):

In [8]:
```
# Cargar el conjunto de datos
from sklearn import datasets
dataset = datasets.fetch_openml(name='delta_elevators', version=1, as_frame=True)
delta = dataset.frame
delta
```

Out[8]:

	climbRate	Altitude	RollRate	curRoll	diffClb	diffDiffClb	Se
0	2.0	-50.0	-0.0048	-0.001	0.2	0.00	-0.001
1	6.5	-40.0	-0.0010	-0.009	0.2	0.00	0.003
2	-5.9	-10.0	-0.0033	-0.004	-0.1	0.00	-0.001
3	-6.2	-30.0	-0.0022	-0.011	0.1	0.00	-0.002
4	-0.2	-40.0	0.0059	-0.005	0.1	0.00	0.001
...	...	...	...	...	...	...	...
9512	5.0	-30.0	0.0013	-0.004	0.2	0.00	0.004
9513	1.4	0.0	0.0024	0.019	-0.2	-0.01	-0.001
9514	-3.5	-10.0	-0.0082	0.004	-0.1	0.00	-0.003
9515	-2.4	-10.0	-0.0065	-0.012	0.2	-0.02	-0.001
9516	4.7	-10.0	0.0018	-0.020	0.3	0.00	0.001

9517 rows × 7 columns

In [11]:
```
from sklearn.feature_selection import f_regression
X = delta[delta.columns[:-1]]
y = delta['Se']
f_regression(X, y)
```

Out[11]:
```
(array([7047.14975779,  116.35130948,  213.17334905,  194.46228325,
        6346.09648706,   27.11526825]),
 array([0.00000000e+00, 5.70018042e-27, 9.14696936e-48, 9.08794945e-44,
        0.00000000e+00, 1.95684000e-07]))
```

De nuevo los valores más altos del estadístico (primera fila del resultado) serán las características más relevantes para el problema.

Ejercicio: elegir un conjunto de datos y aplicar a sus características la medida más adecuada según el tipo de problema (variable dependiente discreta o continua).