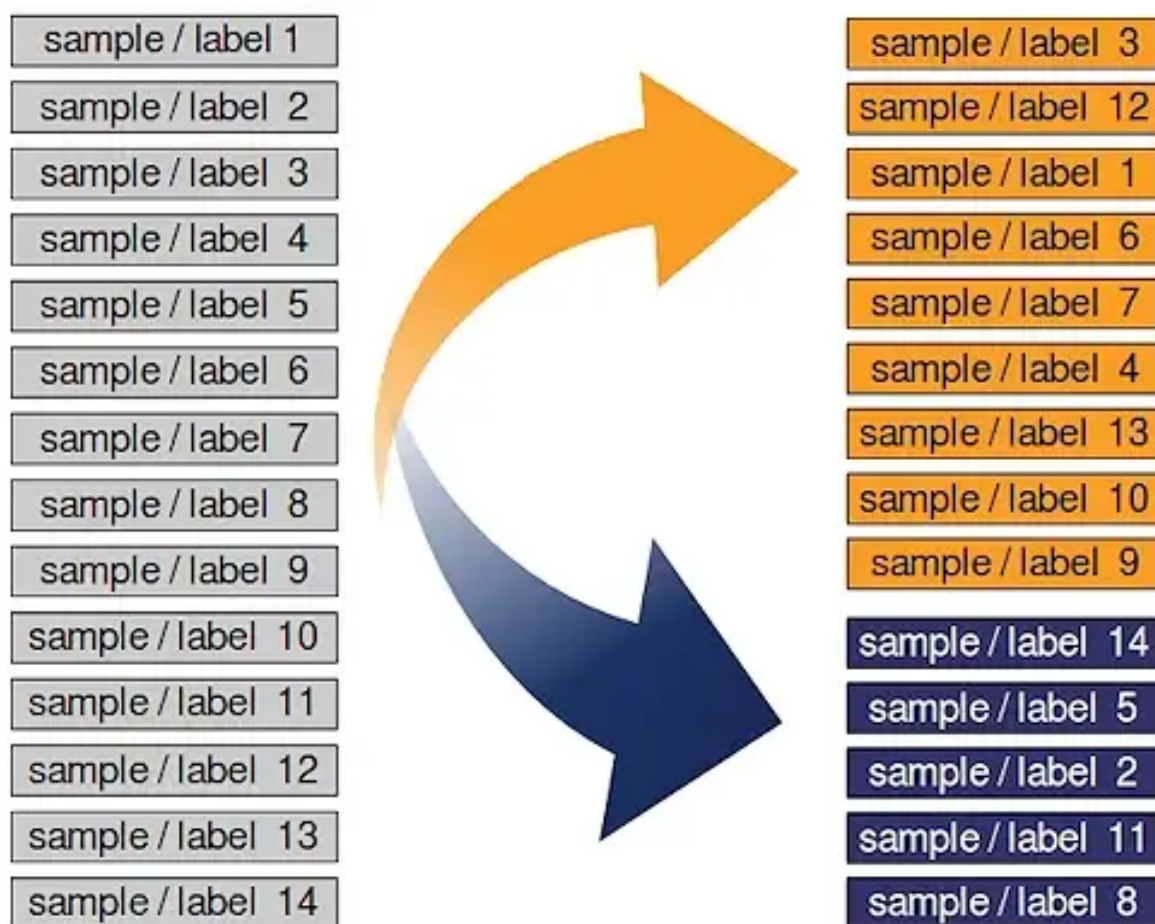


1 MACHINE LEARNING – DIVISI3N DE DATOS – TRAIN TEST

Cuando tenemos un DataSet al que queremos aplicar un modelo de Machine Learning, a este 3ltimo debemos entrenarlo con un conjunto de datos y probarlo con otro conjunto de datos diferente (del que sepamos el resultado) de esta manera podemos saber cu3nto eficiente es el modelo.

Por ejemplo, si tenemos un conjunto de datos como el de la izquierda, debemos separarlos en dos conjuntos de datos diferentes, aleatorios.



Al primero de ellos (el m3s numeroso con un 80% aproximadamente de las muestras) se llama TRAIN DATASET y al resto se llama TEST DATASET.

Lo primero que hacemos es SEPARAR las columnas objetivo (Y) de las caracter3sticas (X)



```
X = df[['Mileage', 'Age(yrs)']]  
y = df['Sell Price($)']
```

Una vez realizado, usamos la función `train_test_split` de `sklearn.model_selection`. Esta función devuelve 4 datasets, 2 para entrenamiento (train) con las características y objetivo y otros dos para el test con sus características y objetivo.

El parámetro `test_size` indica el % de test (30% en este caso)

```
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3)
```

Cuando ya lo tenemos, solamente debemos aplicar el modelo en los datasets de train y predecir el test.

```
from sklearn.linear_model import LinearRegression  
clf = LinearRegression()  
clf.fit(X_train, y_train)  
  
clf.predict(X_test)
```

Con la función `score` del modelo podemos saber el % de predicciones válidas.

```
clf.score(X_test, y_test)
```

1.1 Argumento `random_state` del método `train_test_split`

Este argumento indica la semilla de aleatoriedad, si siempre le pasamos la misma, nos aseguramos que el conjunto de test y train es siempre el mismo, así podemos comparar modelos con los mismos datos de train y de test. Si no se lo indicamos, el conjunto de datos de train y de test son siempre aleatorios y por tanto el modelo puede dar resultados diferentes.