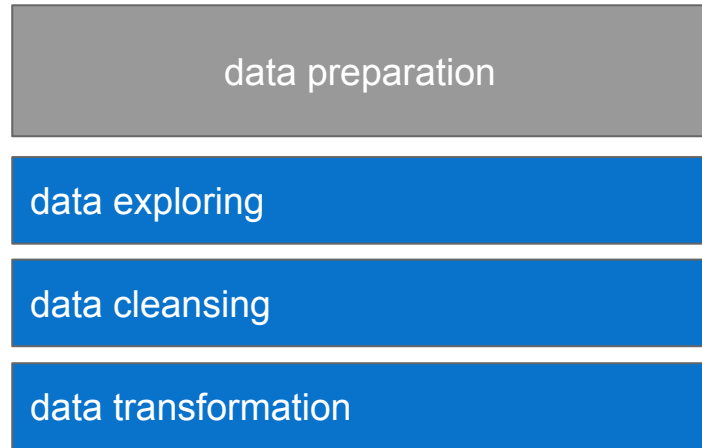
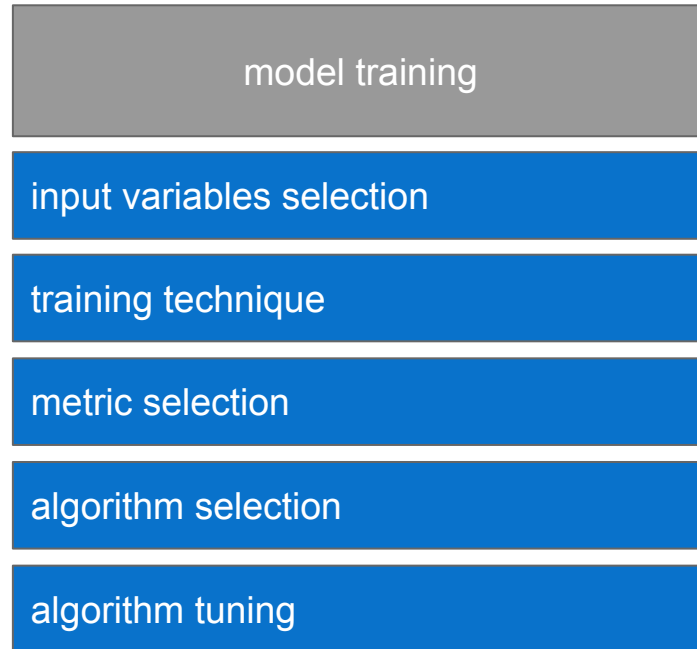


Tema 3: Hands-on Classification Problem

Modeling Process



80% time



20% time



Ejemplo

Load basic libraries

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

sns.set_style('darkgrid')
np.set_printoptions(precision=2)
```

Load data

```
pd_data = pd.read_csv('./data/titanic.csv')
pd_data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Contenido

1. Limpieza de datos
2. Transformación de variables
3. Selección de variables predictivas
4. Técnicas de entrenamiento de los algoritmos
5. Métricas de evaluación (problema de clasificación)
6. Selección de algoritmos (problema de clasificación)
7. Parametrización de algoritmos



Contenido

1. Limpieza de datos
2. Transformación de variables
3. Selección de variables predictivas
4. Técnicas de entrenamiento de los algoritmos
5. Métricas de evaluación (problema de clasificación)
6. Selección de algoritmos (problema de clasificación)
7. Parametrización de algoritmos



Limpieza de datos

También llamada *data cleansing*.

Tras una primera exploración a los datos, vamos a centrarnos ahora en algunas operaciones habituales de limpieza de datos:

- Detectar posibles filas duplicadas
- Detectar posibles columnas no informativas
- Detectar posibles Na/Null values
- Detectar posibles outliers

Limpieza de datos

Filas duplicadas

- Las filas duplicadas sesgan cualquier modelo predictivo.
- Es, por tanto, prioritario, eliminarlos antes de usarlos en cualquier modelo.
- En muchas ocasiones, realizamos la eliminación de los duplicados siempre al principio de cualquier trabajo.
- Pero, al ir transformando nuestro dataset, se van eliminando columnas y pueden existir filas que se distinguían en el dataset original por alguna de las variables eliminadas. Por tanto, conviene realizar este proceso también justo antes de entrenar un modelo.



Limpieza de datos

Columnas poco informativas

- Las columnas categóricas que no tienen todos sus valores iguales, no aportan información para cualquier modelo.
- Éstas variables, se pueden identificar relativamente bien en cualquier análisis exploratorio con un simple barplot.
- Igualmente, las columnas numéricas con poca varianza tienen el mismo problema. Pero para identificarlas cuesta siempre un poco más.



Limpieza de datos

Null / Na values

- No tiene sentido trabajar con valores nulos. Hay que tomar una decisión:
 - O se transforman
 - O se rellenan
 - O se eliminan
- Y pueden estar tanto en las columnas como en las filas. Es decir, podemos tener filas con casi todos sus valores nulos o columnas con casi todos sus valores nulos.
- La decisión tiene un carácter subjetivo que depende del data scientist y de la naturaleza del modelo y los datos.
- Cualquier decisión que tomemos va a implicar un sesgo en el modelo.



Limpieza de datos

Null / Na values

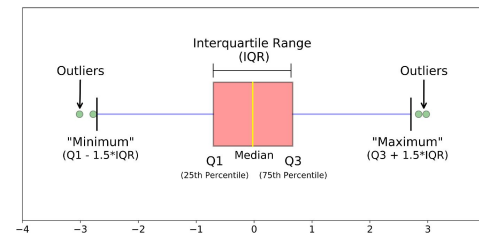
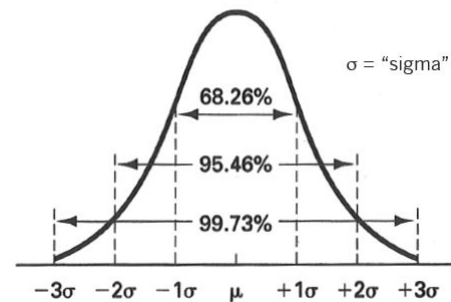
- En muchas ocasiones, se pueden llegar variables dummy (booleanas) que indiquen si la columna tiene o no valor nulo. Generalmente se hace cuando hay un 50-50%.
- Cuando es una serie temporal, se suelen interpolar los datos.
- Cuando representan menos del 5% de los datos, se podrían eliminar.
- Cuando no tenemos claro qué estrategia llevar, se pueden sustituir por valores medios, medianos, modas o el valor más cercano. Aquí podemos utilizar *Imputers*,
<https://scikit-learn.org/stable/modules/classes.html#module-sklearn.impute>
- En ocasiones más específicas se pueden generar números aleatorios que sigan la distribución de la variable (*synthetic data generation*).



Limpieza de datos

Outliers

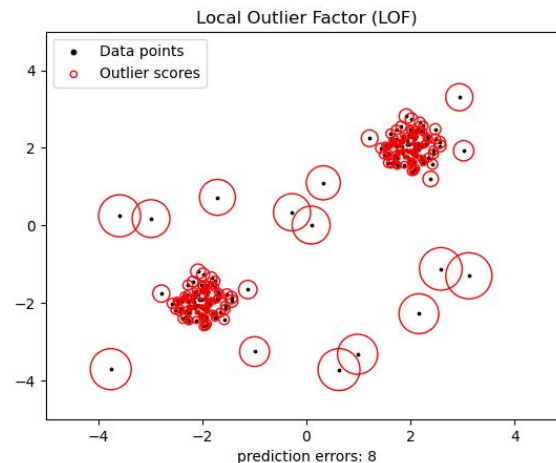
- En distribuciones normales, se cumple que el 99.7% de los datos debe estar en el intervalo "seis-sigma" alrededor de la media. Lo que esté fuera de este intervalo podría ser outlier.
- Otra forma de hacer lo mismo es mediante el Rango Intercuartílico para localizar outliers, pero igualmente, funciona bien en distribuciones normales.



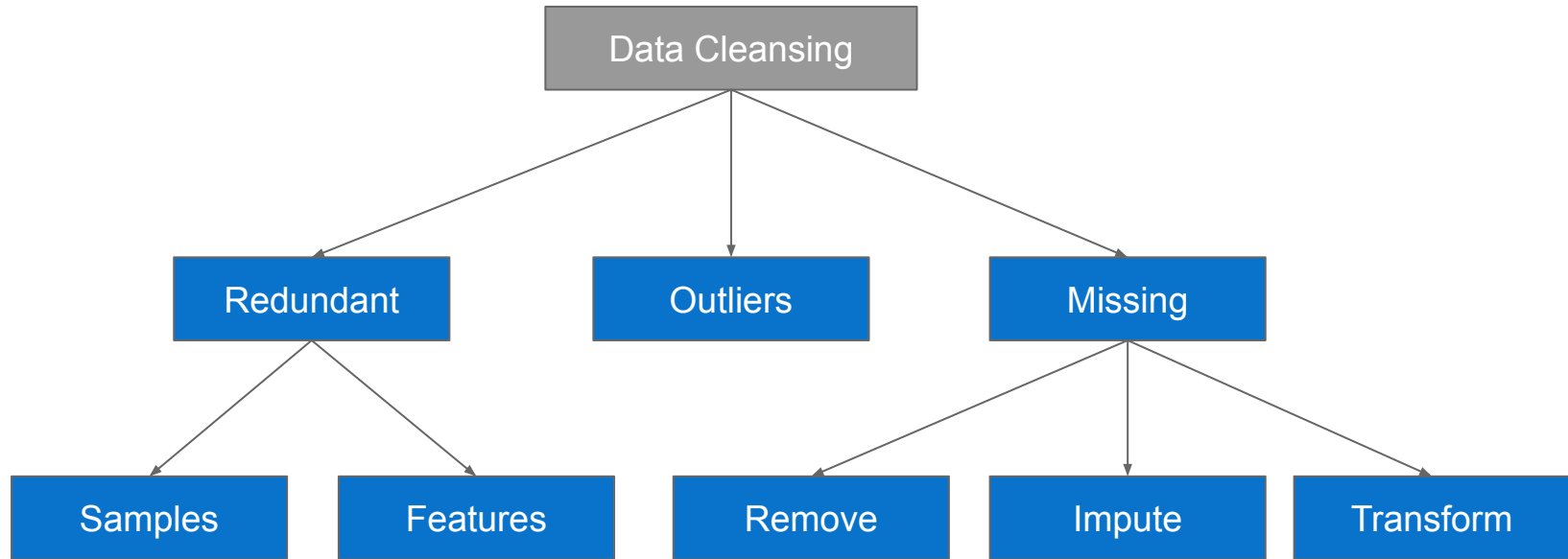
Limpieza de datos

Outliers

- Otra forma interesante es utilizar la técnica de Local Outlier Factor basado en densidad de población.
- Similar al algoritmo DBSCAN.
- Ambos, no supervisados y depende de la parametrización inicial.
- Y además requiere dos o más dimensiones, a diferencia de las técnicas anteriores.



Limpieza de datos



Contenido

1. Limpieza de datos
2. **Transformación de variables**
3. Selección de variables predictivas
4. Técnicas de entrenamiento de los algoritmos
5. Métricas de evaluación (problema de clasificación)
6. Selección de algoritmos (problema de clasificación)
7. Parametrización de algoritmos



Transformación de variables

Vamos a distinguir transformaciones según el tipo de variable.

Para categóricas, veremos aquí:

- *Label encoder*
- *Ordinal encoder*
- *One hot encoder*
- *Dummy variables*

Para numéricas, ya estudiamos *normalize*, *standardize*, *minmaxscale* y *binarize*. En este ejemplo vamos a ver algunas más robustas (los outliers no afecten demasiado):

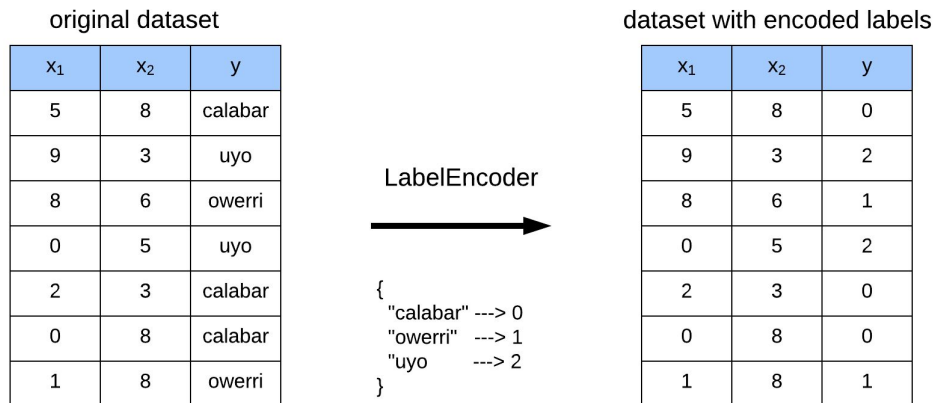
- *Robust Scaler*
- *Box-Cox*



Transformación de variables

Label encoder

- Transforma a numérico (ordinal) la variable target: {0,1,...}




Transformación de variables

Ordinal encoder

- Transforma a numérico (ordinal) cualquiera de las variables predictoras manteniendo sentido de orden: {0,1,2,...}.

Ordinal Encoding

Breakfast		Breakfast
Every day		3
Never		0
Rarely		1
Most days		2
Never		0

Transformación de variables

One hot encoder

- Transforma a boolean cualquiera de las variables predictoras y crea variables para cada una de las opciones.

id	color
1	red
2	blue
3	green
4	blue



id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Transformación de variables

Dummy variables

- Transforma a boolean cualquiera de las variables predictoras y crea variables para cada una de las opciones.

id	color
1	red
2	blue
3	green
4	blue

Pandas Get Dummies



id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Transformación de variables

Robust scaler

- El principal problema de los outliers es que afecta mucho a toda fórmula que conlleve en el cálculo operaciones algebraicas sobre los datos:
 - Medias, desviaciones, varianzas, etc.
 - Algoritmos como regresión lineal, linear discriminant analysis, etc.
- Sin embargo, no afecta tanto a otros tipos de cálculos:
 - Percentiles, Cuartiles, etc.
 - Algunos algoritmos como decision trees, random forest, etc.

$$\text{value} = \frac{\text{value} - \text{mean}}{\text{standard_deviation}} \quad \Rightarrow \quad \text{value} = \frac{\text{value} - \text{median}}{p_{75} - p_{25}}$$

Transformación de variables

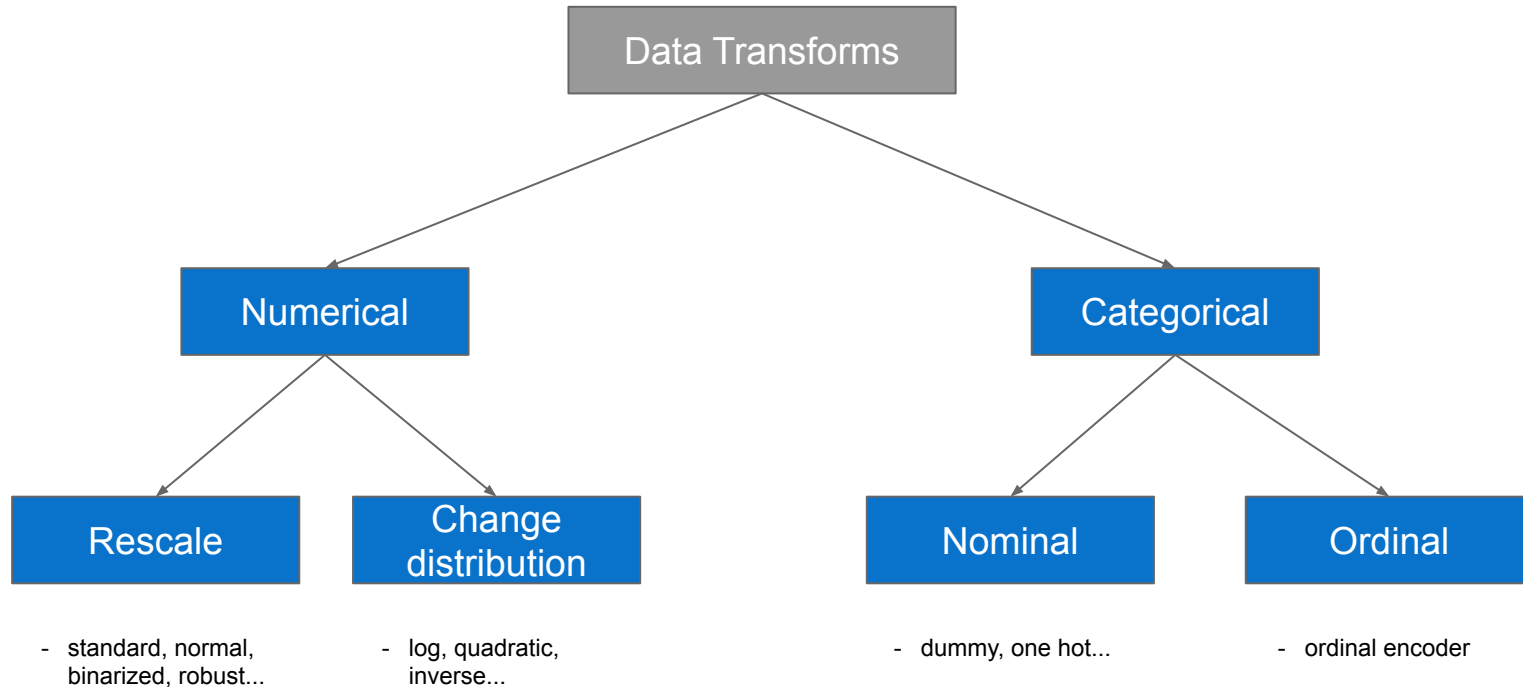
Box-Cox

- Mediante transformaciones no lineales podemos conseguir que una variable aleatoria con una distribución no gaussiana consiga aproximarse a algo más normal y tratable.
- Aunque haya más posibles opciones de transformación de variables. Box-Cox nos permite, rápidamente, intuir cuál sería la transformación más aproximada.

$$y_i^{(\lambda)} = \begin{cases} \frac{y_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(y_i) & \text{if } \lambda = 0, \end{cases}$$

λ	Transformed Data
-2	y^{-2}
-1	y^{-1}
-0.5	$1/\sqrt{y}$
0	$\ln(y)$
0.5	\sqrt{y}
1	y
2	y^2

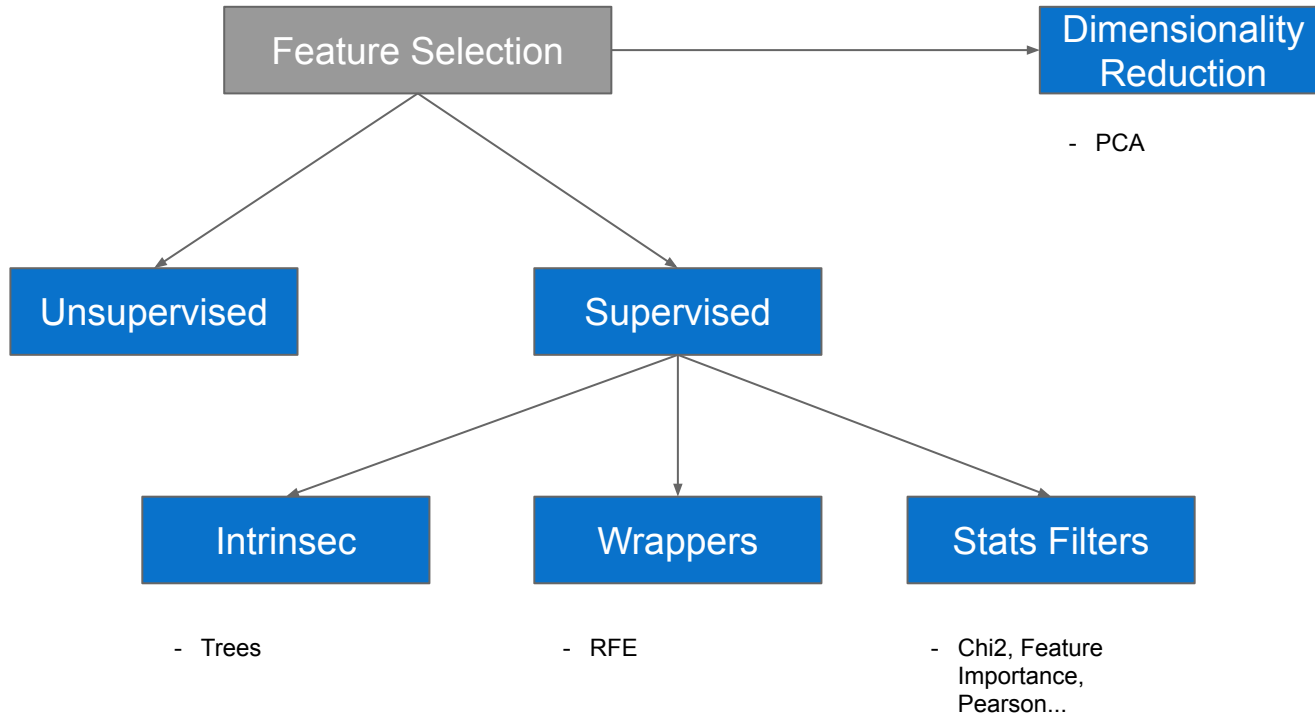
Transformación de variables



Contenido

1. Limpieza de datos
2. Transformación de variables
3. **Selección de variables predictivas**
4. Técnicas de entrenamiento de los algoritmos
5. Métricas de evaluación (problema de clasificación)
6. Selección de algoritmos (problema de clasificación)
7. Parametrización de algoritmos

Selección de variables predictivas



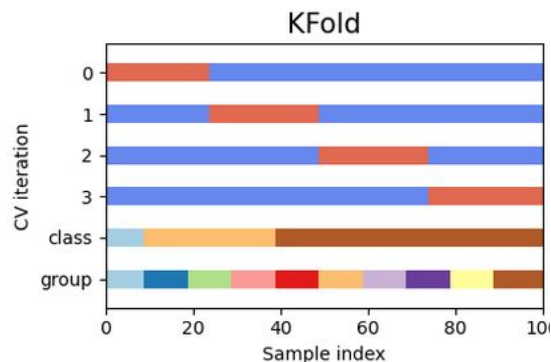
Contenido

1. Limpieza de datos
2. Transformación de variables
3. Selección de variables predictivas
- 4. Técnicas de entrenamiento de los algoritmos**
5. Métricas de evaluación (problema de clasificación)
6. Selección de algoritmos (problema de clasificación)
7. Parametrización de algoritmos



Técnicas de entrenamiento

Cross Validation Splits



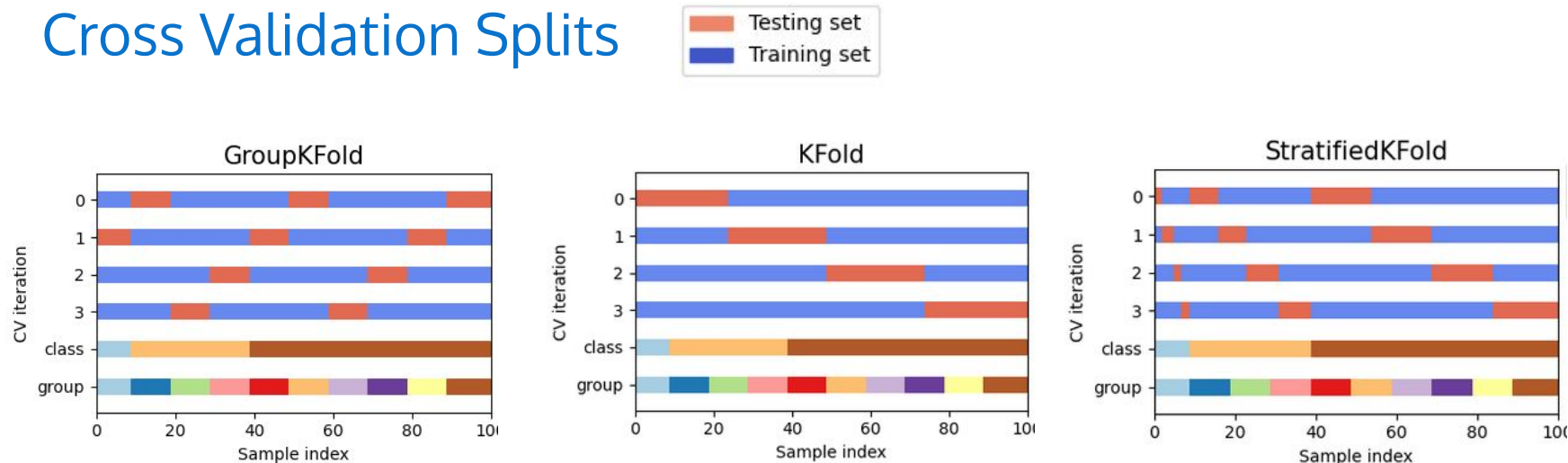
- *cv iteration*: Número de veces que se realiza el fitting del modelo para cada una de las particiones del conjunto de datos original.
- *class*: La etiqueta que tratamos de predecir (*label*). Por ejemplo, si sobrevivió o no al hundimiento del Titanic.
- *group*: Un ejemplo sería cuando se recogen datos médicos de múltiples pacientes, con múltiples muestras tomadas de cada paciente. Y es probable que esos datos dependan del grupo individual. En nuestro ejemplo, la identificación del paciente para cada muestra será su identificador de grupo.

Objetivo: Overfitting & Imbalanced data



Técnicas de entrenamiento

Cross Validation Splits



- El mismo grupo no está representado tanto en *train set* como en *test set*

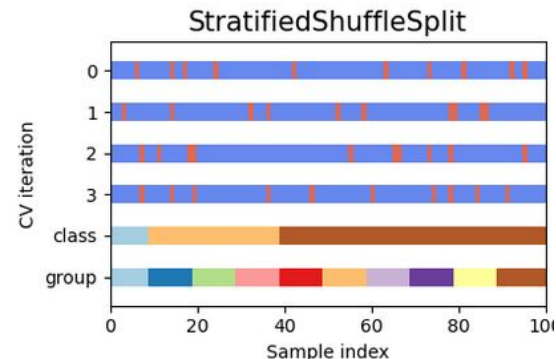
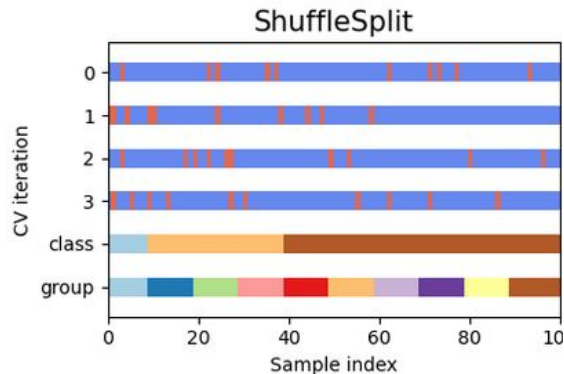
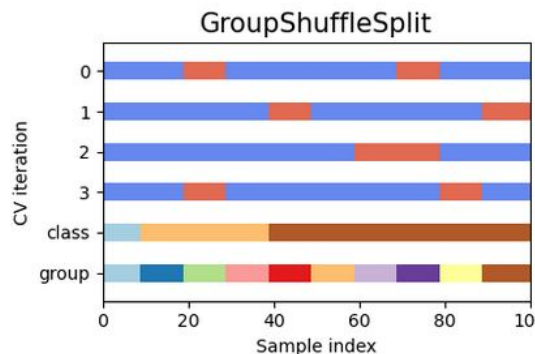
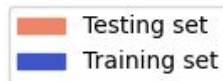
- Cada conjunto contiene aproximadamente el mismo porcentaje de muestras de cada clase que el conjunto completo

- Cada conjunto contiene aproximadamente el mismo porcentaje de muestras de cada clase que el conjunto completo



Técnicas de entrenamiento

Cross Validation Splits



- Las versiones *Shuffle* realizan primero una permutación en el orden de los datos (barajan los registros) y luego seleccionan train/test sets.
- Las versiones *Repeated* permiten hacer cada *cv iteration* varias veces.



Contenido

1. Limpieza de datos
2. Transformación de variables
3. Selección de variables predictivas
4. Técnicas de entrenamiento de los algoritmos
- 5. Métricas de evaluación (problema de clasificación)**
6. Selección de algoritmos (problema de clasificación)
7. Parametrización de algoritmos



Métricas

Métricas de clasificación que veremos aquí se utilizan con bastante frecuencia en Data Science y todas provienen de la matriz de confusión:

- *Precision*
- *Recall*
- *F1*
- *Classification report*

Métricas

Precision-Recall (and F1)

La *precision* es una medida útil del éxito de la predicción cuando las clases están muy desbalanceadas.

La *precision* es una medida de la relevancia de los resultados, mientras que la *recall* es una medida de cuántos resultados verdaderamente relevantes se devuelven.

Finalmente, F1 es la media armónica de ambas.

$$P = \frac{T_p}{T_p + F_p}$$

$$R = \frac{T_p}{T_p + F_n}$$

$$F1 = 2 \frac{P \times R}{P + R}$$

Métricas

Classification report

De dicha matriz de confusión, también podemos mostrar un report.

```
from sklearn.metrics import classification_report

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
model = LogisticRegression()
model.fit(X_train, Y_train)
predicted = model.predict(X_test)
report = classification_report(Y_test, predicted)
print(report)
```

	precision	recall	f1-score	support
B	0.89	0.95	0.92	74
M	0.89	0.78	0.83	40
accuracy			0.89	114
macro avg	0.89	0.86	0.87	114
weighted avg	0.89	0.89	0.88	114



Contenido

1. Limpieza de datos
2. Transformación de variables
3. Selección de variables predictivas
4. Técnicas de entrenamiento de los algoritmos
5. Métricas de evaluación (problema de clasificación)
- 6. Selección de algoritmos (problema de clasificación)**
7. Parametrización de algoritmos



https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html



Contenido

1. Limpieza de datos
2. Transformación de variables
3. Selección de variables predictivas
4. Técnicas de entrenamiento de los algoritmos
5. Métricas de evaluación (problema de clasificación)
6. Selección de algoritmos (problema de clasificación)
7. **Parametrización de algoritmos**



Parametrización de algoritmos

RandomizedSearchCV

A diferencia del *GridSearchCV*, no se prueban todos los valores de los parámetros, sino que se muestrean un número fijo de ajustes de parámetros de las distribuciones especificadas.

El número de ajustes de parámetros que se prueban viene dado por n_iter .

La característica más importante es el ahorro computacional.



https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html#sklearn.model_selection.RandomizedSearchCV

