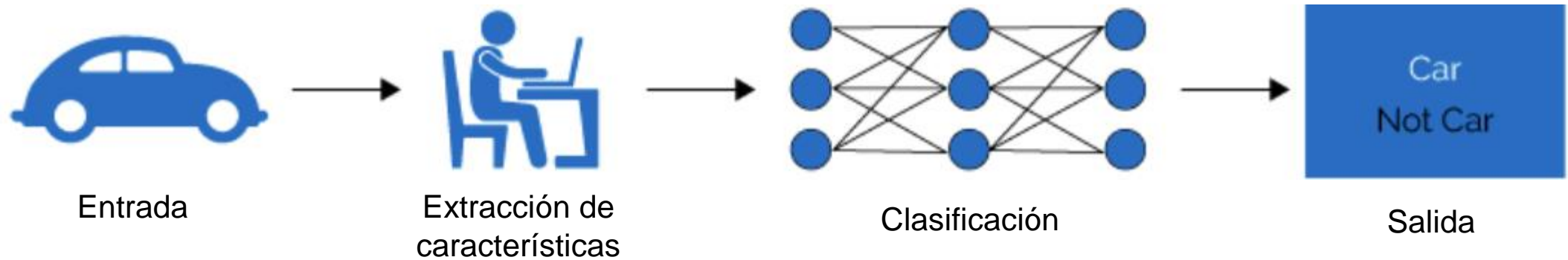


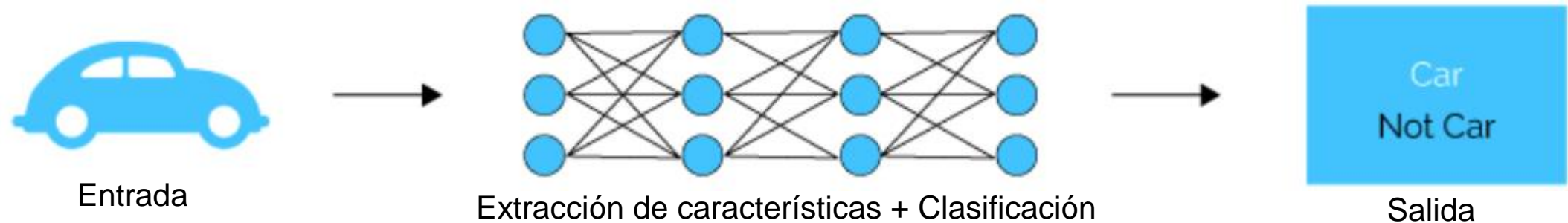
# Contenido del curso

## APRENDIZAJE AUTOMATICO



---

## DEEP LEARNING



# Fase de Preparación de los Datos

- La información almacenada siempre tiene
  - ▣ Datos faltantes
  - ▣ Valores extremos
  - ▣ Inconsistencias
  - ▣ Ruido
- Tareas a realizar
  - ▣ Limpieza (ej: resolver outliers e inconsistencias)
  - ▣ Transformación (ej: discretización)

# Limpieza de los datos

- ✓ En primer lugar, debe tenerse en cuenta que hay distintos tipos de variables o atributos.
- ✓ Para cada tipo se deberá realizar un análisis de sus valores.
  - Luego, se procederá a limpiarlos
    - ▣ Eliminando los valores con ruido
    - ▣ Determinando qué hacer con los valores faltantes.
    - ▣ Eliminando inconsistencias

# Limpieza - Variables con ruido

- Las variables con ruido tendrán valores que caen fuera del rango de sus valores esperados llamados **outliers**.
- Por qué se originan?
  - ▣ Error humano en la carga de datos (ej: una persona puede aparecer con una altura de 5 metros).
  - ▣ Determinados cambios operacionales no han sido registrados en el proceso.

Es preciso analizar los **metadatos**

# Limpieza - Valores faltantes

- Qué hacer con los valores nulos?
  - ▣ Ignorar la tupla.
  - ▣ Rellenar la tupla manualmente.
  - ▣ Usar una constante global para rellenar el valor nulo.
  - ▣ Utilizar el valor de la media u otra medida de centralidad para rellenar el valor.
  - ▣ Utilizar el valor de la media u otra medida de centralidad de los objetos que pertenecen la misma clase.
  - ▣ Utilizar alguna técnica de Aprendizaje Automático para calcular el valor más probable.

# Ejemplo

Premios2020.csv

- Se dispone de la siguiente información de los premios de la Academia otorgados a los mejores actores y actrices desde 1928 hasta 2020.
  - ▣ Año en que fue otorgado el premio
  - ▣ Datos del actor que lo recibió: Nombre, edad, sexo
  - ▣ Datos de la película: Título, género, duración, rating, cantidad de nominaciones que recibió, mes de estreno, sinopsis

# Ejercicio

Premios2020.csv

◆ El archivo **Premios2020.csv** contiene 186 premios otorgados

Year	Age	Actor	Sex	Film	nominat	rating	duration	genre1	genre2	release	synopsis
1928	44	Emil Jannings	M	The Last Command	2	8	88	Drama	History	April	A former Imperial Russian gener
1928	22	Laura Gainor (aka Janet G	F	Sunrise	5	7.8	110	Drama	Romance		A street cleaner saves a young w
1929	37	Mary Pickford	F	Coquette	1	7.3	76	Drama	Romance	April	A flirtatious southern belle is co
...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...
2019	45	Joaquin Phoenix	M	Joker	11	8.5	122	Drama	Thriller	October	Arthur Fleck loves to make peop
2020	63	Frances McDormand	F	Nomadland	6	7.4	108	Drama		September	Nomadland es una película estac
2020	83	Anthony Hopkins	M	The father	6	8.3	97	Drama		January	Anthony tiene casi 83 años. Vive

- ¿Cuántos atributos tiene la tabla?
- ¿De qué tipo es cada uno de ellos?

# Premios2020.csv

Faltantes.ipynb

```
import pandas as pd
import numpy as np
import os
import chardet

os.chdir('../Datos//')

nomArch = 'Premios2020.csv'

with open(nomArch, 'rb') as f:
    result = chardet.detect(f.read())

df= pd.read_csv(nomArch, encoding=result['encoding'])

print(df.isnull().sum())
```



Year	0
Age	0
Actor	0
Sex	0
Film	0
nominations	8
rating	0
duration	0
genre1	0
genre2	37
release	4
synopsis	0



# Reemplazando los valores faltantes

```
import pandas as pd
import numpy as np

df= pd.read_csv('../Datos/Premios2020.csv', encoding='ISO-8859-1')

values = {'nominations': df['nominations'].min(), 'rating': 5}
df3 = df.fillna(value=values)

#-- reemplaza todos los nan con 0
df4 = df.replace(np.nan, 0)
```

Faltantes.ipynb

# Reemplazando los valores faltantes

```
import pandas as pd
import numpy as np

df= pd.read_csv('../Datos/Premios2020.csv',encoding='ISO-8859-1')

#crea una copia de df
df5 = pd.DataFrame(df)
modaGen = df5['genre2'].mode()

df5['genre2'] = df5['genre2'].replace(np.nan, modaGen[0])
```

Faltantes.ipynb

# Atributo GENRE1 - Reducción de valores

```
import pandas as pd
import numpy as np
```

**Modifica\_atrib.ipynb**

```
df= pd.read_csv('../Datos/Premios2020.csv',encoding='ISO-8859-1')
```

```
opciones = pd.value_counts(df['genre1'])
print(opciones)
```

Reemplazar por  
“OTRA”

Drama	91
Biography	41
Comedy	25
Crime	16
Adventure	6
Action	3
Romance	2
Mystery	1
Thriller	1

# Atributo GENRE1 - Reducción de valores

Modifica\_atrib.ipynb

```
import pandas as pd
import numpy as np
df= pd.read_csv('../Datos/Premios2020.csv',encoding='ISO-8859-1')

opciones = pd.value_counts(df['genre1'])
print(opciones)

# Reemplazando valores
df['genre1'] = df['genre1'].replace(['Adventure','Action', \
                                   'Romance', 'Thriller', 'Mystery'], 'Otra')

# revisar cómo quedó
opciones2 = pd.value_counts(df['genre1'])
print(opciones2)
```



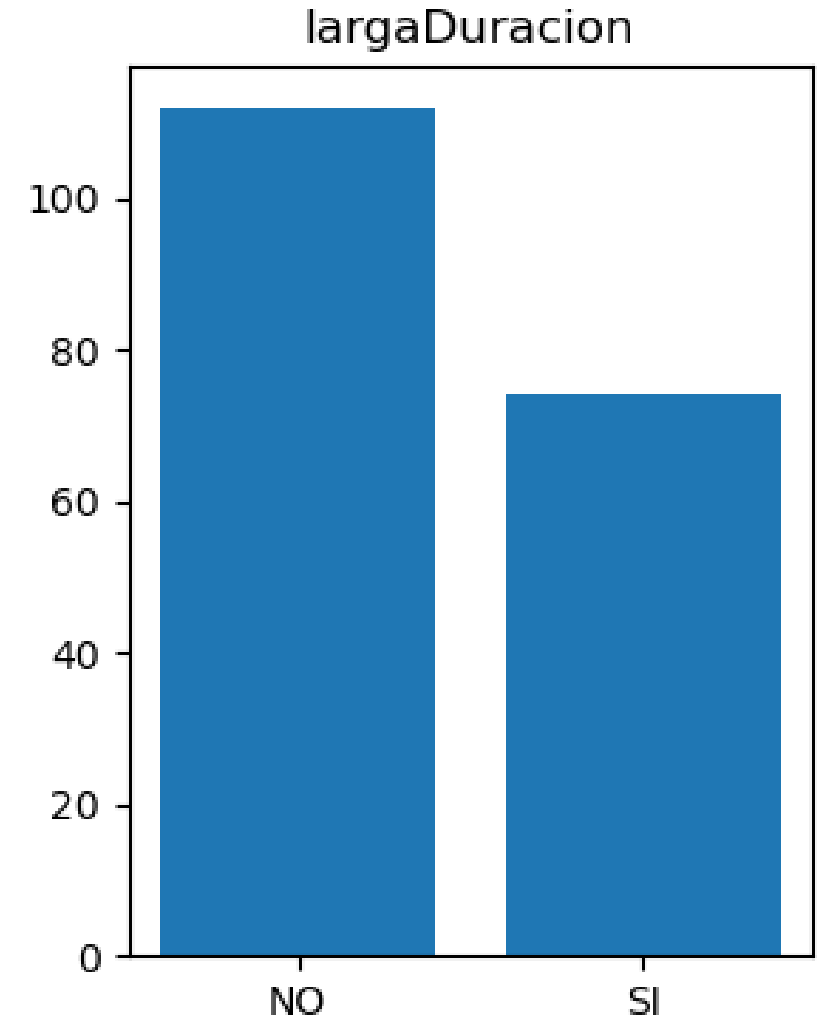
Drama	91
Biography	41
Comedy	25
Crime	16
Otra	13

# Ejemplo de creación de atributos

Atributo derivado	Fórmula
Índice de obesidad	$\text{Altura}^2 / \text{peso}$
Hombre familiar	Casado, varón e (hijos > 0)
Síntomas SARS	3-de-5 (fiebre alta, vómitos, tos, diarrea, dolor de cabeza)
Riesgo de póliza	X-de-N (edad < 25, varón, años que conduce < 2, vehículo deportivo)
Beneficios Brutos	Ingresos – Gastos
Beneficios netos	Ingresos – Gastos – Impuestos
Desplazamiento	Pasajeros * kilómetro
Duración media	Segundos de llamada / número de llamadas
Densidad	Población / Área
Retardo compra	Fecha compra – Fecha campaña

# Generando un atributo nuevo

- Genere un nuevo atributo **largaDuracion** cuyo valor será “SI” si la película tiene una duración superior a 2 horas y “NO” en caso contrario.
- Grafique este nuevo atributo utilizando un diagrama de barras.



# Generando un atributo nuevo


```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
df= pd.read_csv('../Datos/Premios2020.csv',encoding='ISO-8859-1')

LD = ['NO'] * len(df)
for i in range(len(df)):
    if df['duration'][i] > 120:
        LD[i] = 'SI'

# Agregando un atributo al DataFrame
df = df.assign( largaDuracion = LD )
print('Atributo largaDuracion')
print(pd.value_counts(df['largaDuracion']))
```

Modifica\_atrib.ipynb

# Transformación de atributos

- DISCRETIZACION 
  - ▣ Algunos algoritmos de minería de datos sólo operan con atributos cualitativos. La discretización convierte los atributos numéricos en ordinales.
- NUMERIZACION
  - ▣ Es el proceso contrario a la discretización. Convierte atributos cualitativos en numéricos.
- NORMALIZACION
  - ▣ Permite expresar los valores de los atributos sin utilizar las unidades de medida originales facilitando su comparación y uso conjunto.



# Discretización

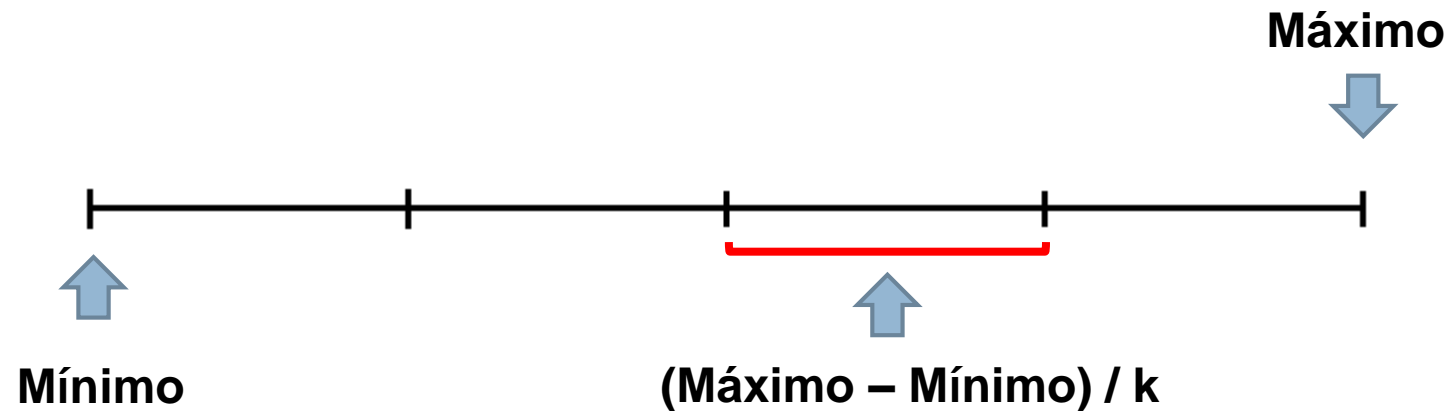
- Convierte un valor numérico en un nominal ordenado (que representa un intervalo o "*bin*")
- **Ejemplo:** Podemos transformar
  - ▣ la edad de la persona en categorías:  $[0, 12]$  niño,  $(12, 21)$  joven,  $[21, 65]$  adulto y  $>65$  anciano.
  - ▣ La calificación de un alumno en:  $[4, 10]$  aprobado o  $[0, 4)$  desaprobado

# Discretización

- Puede discretizarse en un número fijo de intervalos. El ancho del intervalo se calcula
  - ▣ Dividiendo el rango en partes iguales
  - ▣ Dividiendo la cantidad de ejemplos en partes iguales (igual frecuencia)
  - ▣ Indicando los límites de cada intervalo en forma manual.
- Averigüe por otras variantes de discretización

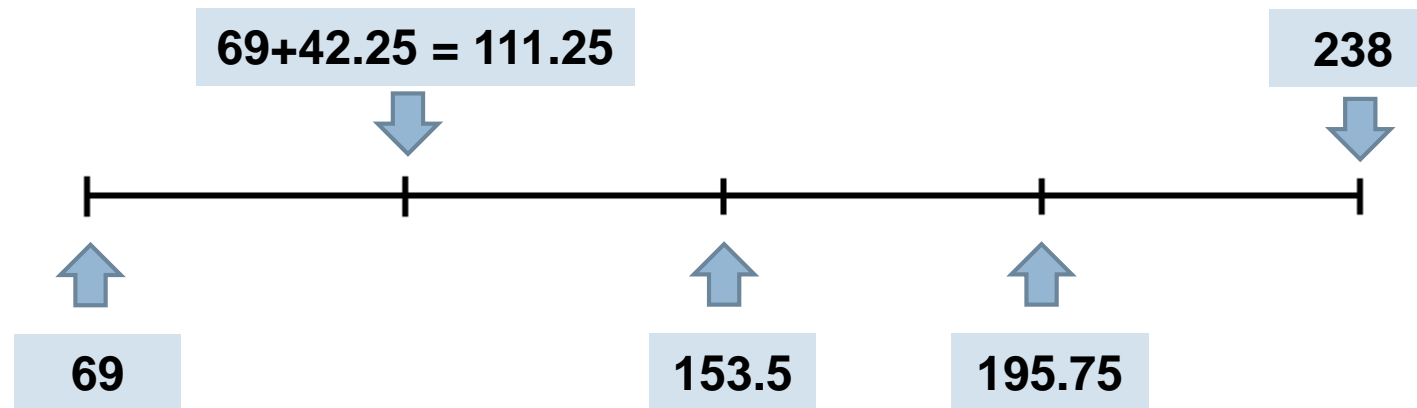
# Discretización por rango

- El objetivo es dividir el rango del atributo (intervalo entre el máximo y el mínimo) en una cierta cantidad  $k$  de partes iguales.
- Los valores comprendidos en una misma parte serán asociados al mismo valor ordinal.
- Ejemplo:  $k=4$



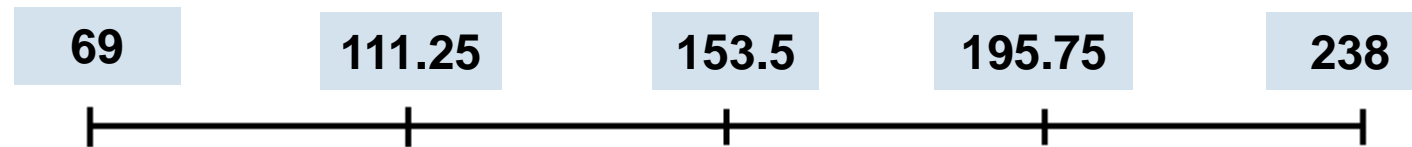
# Discretización por rango

- **Ejemplo: Discretizar el atributo DURATION en 4 intervalos de igual longitud**
  - ▣ DURATION toma valores entre 69 y 238 minutos. Si dividimos el rango en 4 partes iguales, cada una tendría una longitud de  $(238-69)/4 = 42.25$



# Discretización por rango

- **Ejemplo: Discretizar el atributo DURATION en 4 intervalos de igual longitud**



Valor	Intervalo	Frecuencia
Rango1	$[-\infty - 111.25]$	77
Rango2	$(111.25 - 153.5]$	92
Rango3	$(153.5 - 195.75]$	15
Rango4	$(195.75 - \infty]$	2

# DURATION discretizado en 4 intervalos por rango

```
import pandas as pd
import numpy as np
df= pd.read_csv('../Datos/Premios2020.csv',encoding='ISO-8859-1')

etiq = ["bajo","medio","alto", "muy alto"]

# Discretización por RANGO
columna = pd.cut(df["duration"],bins=len(etiq),labels=etiq)

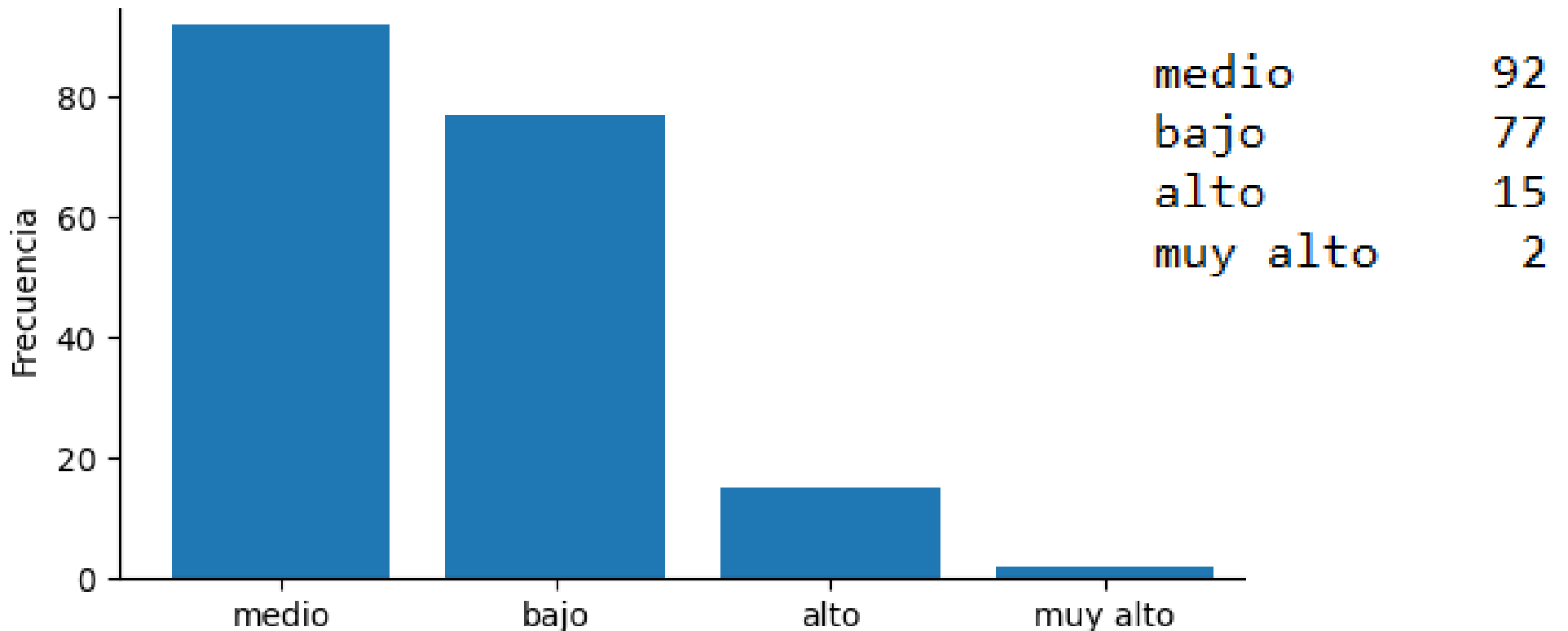
df['duration2']= pd.Series.to_frame(columna)

print(pd.value_counts(df['duration2']))
```

Discretizacion.ipynb

# Discretización por rango

- DURATION discretizado en 4 intervalos de igual longitud



# Discretización por frecuencia

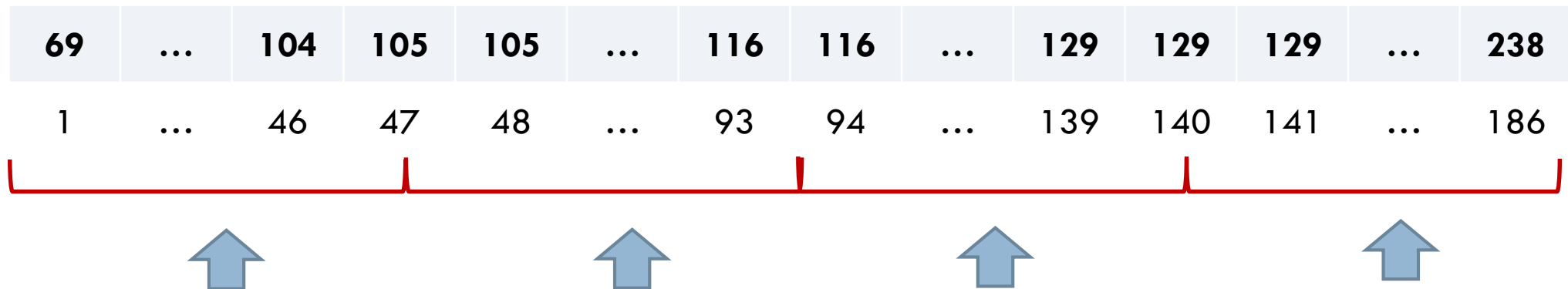
- El objetivo es dividir los valores del atributo numérico en  $k$  partes con la misma cantidad de valores en cada una de ellas.
- El atributo debe tener al menos  $k$  valores diferentes.
- Si hay valores numéricos repetidos los valores ordinales no tendrán la misma frecuencia.



# Discretización por frecuencia

## □ Ejemplo: Discretizar el atributo DURATION en 4 intervalos de igual frecuencia

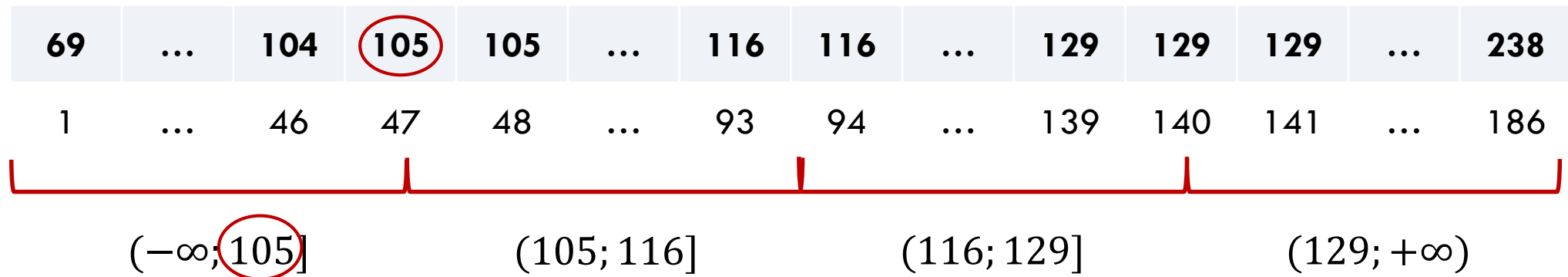
- DURATION tiene 186 valores entre 69 y 238 minutos. Luego de ordenar los valores, los dividimos en k partes con igual cantidad de elementos



Cada intervalo tiene  $N/K = 186/4 = 46.5$  elementos

# Discretización por frecuencia

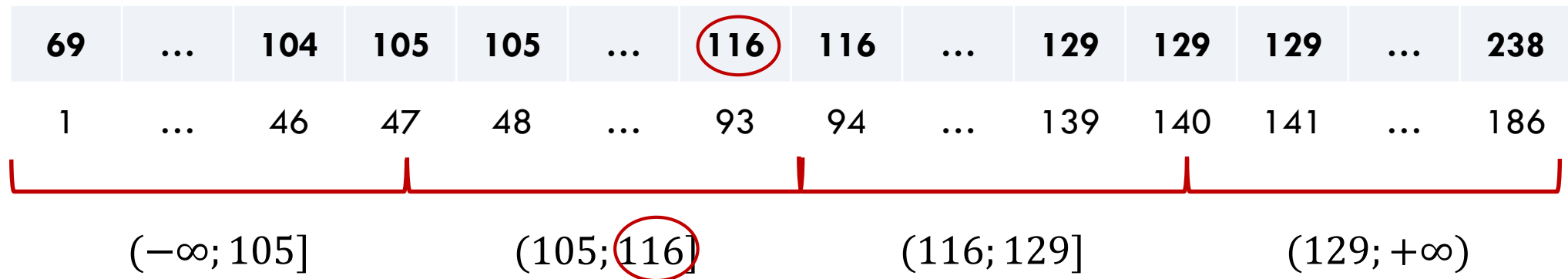
- **Ejemplo: Discretizar el atributo DURATION en 4 intervalos de igual frecuencia**
  - ▣ DURATION tiene 186 valores entre 69 y 238 minutos. Luego de ordenar los valores, los dividimos en k partes con igual cantidad de elementos



# Discretización por frecuencia

## □ Ejemplo: Discretizar el atributo DURATION en 4 intervalos de igual frecuencia

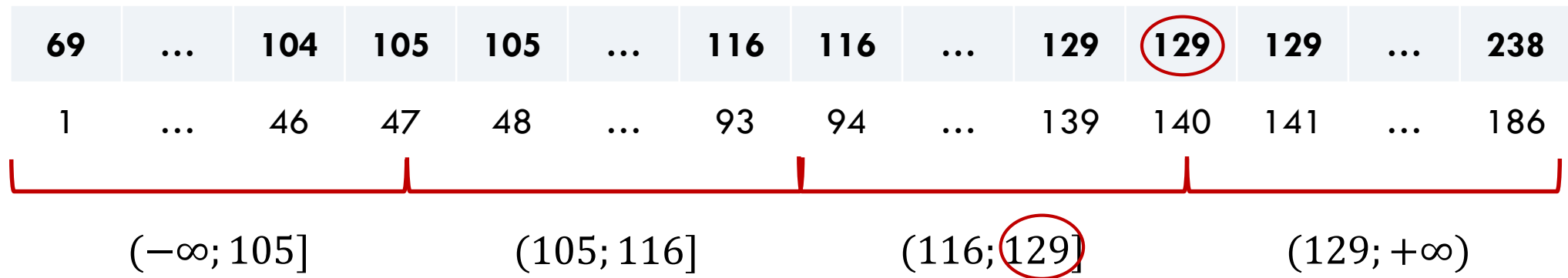
- DURATION tiene 186 valores entre 69 y 238 minutos. Luego de ordenar los valores, los dividimos en k partes con igual cantidad de elementos



# Discretización por frecuencia

## □ Ejemplo: Discretizar el atributo DURATION en 4 intervalos de igual frecuencia

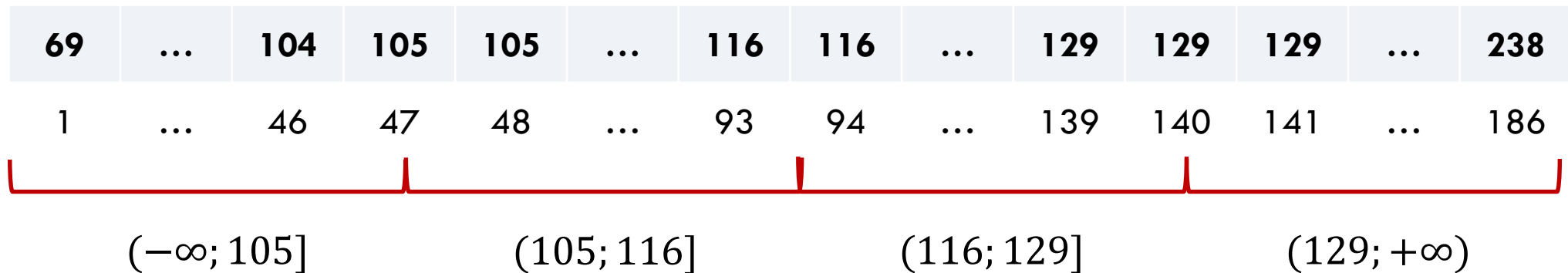
- DURATION tiene 186 valores entre 69 y 238 minutos. Luego de ordenar los valores, los dividimos en k partes con igual cantidad de elementos



# Discretización por frecuencia

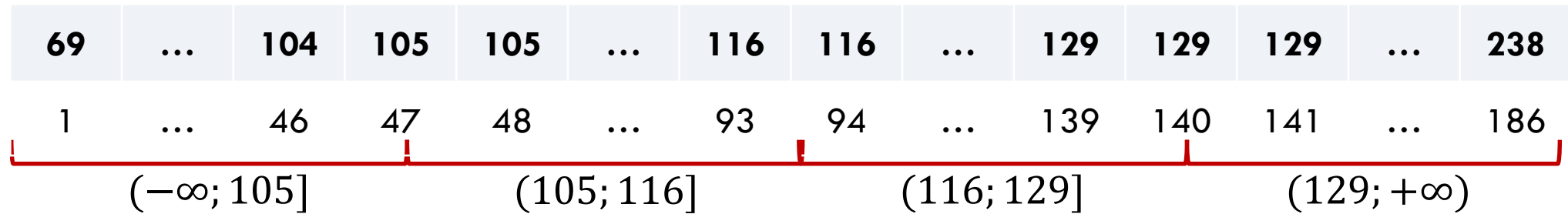
## □ Ejemplo: Discretizar el atributo DURATION en 4 intervalos de igual frecuencia

- DURATION tiene 186 valores entre 69 y 238 minutos. Luego de ordenar los valores, los dividimos en k partes con igual cantidad de elementos



# Discretización por frecuencia

- **Ejemplo: Discretizar el atributo DURATION en 4 intervalos de igual frecuencia**



Valor	Intervalo	Frecuencia
range1	$[-\infty - 105]$	53
range2	$(105 - 116]$	47
range3	$(116 - 129]$	45
range4	$(129 - \infty]$	41

# DURATION discretizado en 4 intervalos por frecuencia

```
import pandas as pd
import numpy as np
df= pd.read_csv('../Datos/Premios2020.csv',encoding='ISO-8859-1')

etiq = ["bajo","medio","alto","muy alto"]

# Discretización por FRECUENCIA
columna = pd.qcut(df["duration"], q=len(etiq), labels=etiq)

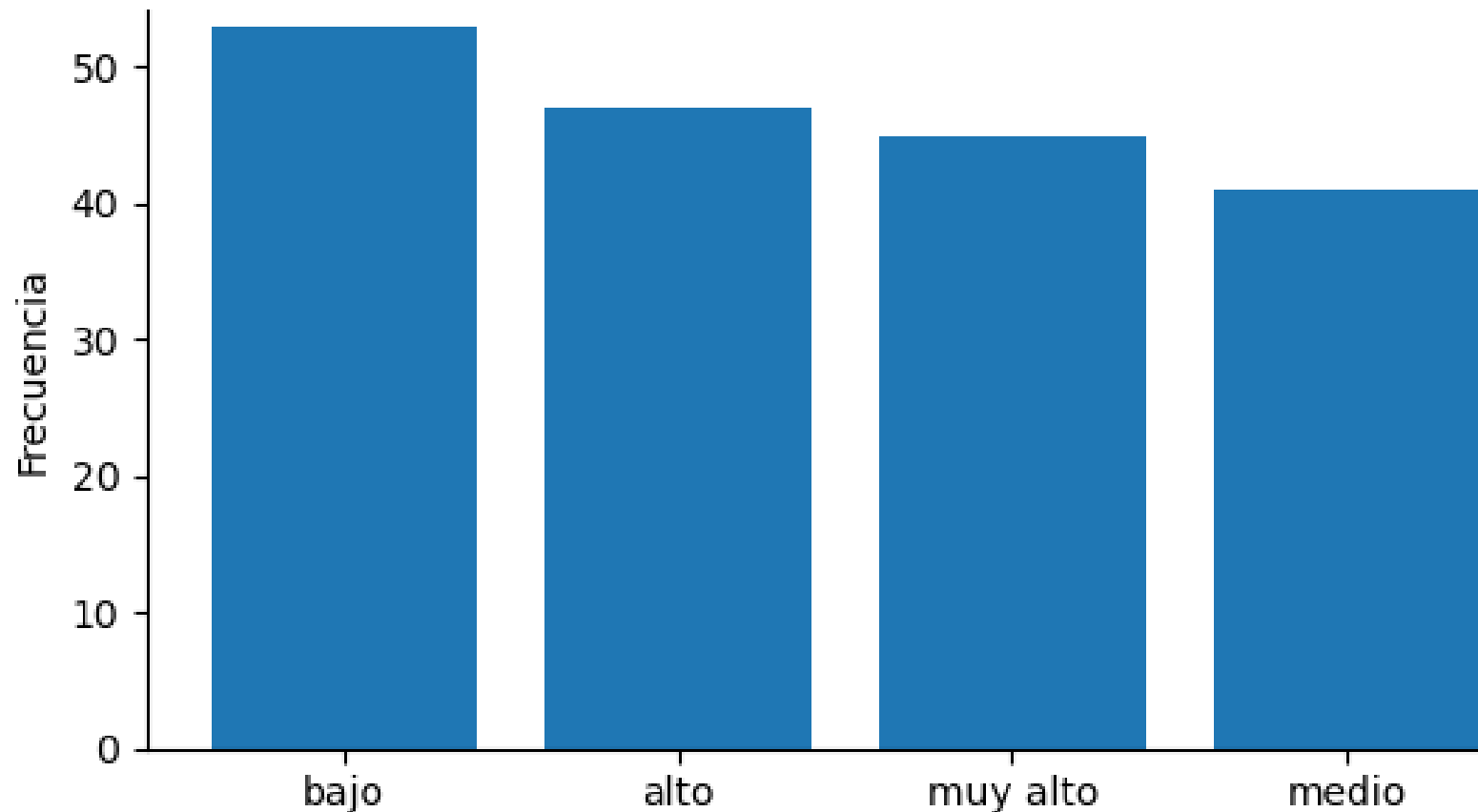
df['duration2']= pd.Series.to_frame(columna)

print(pd.value_counts(df['duration2']))
```

Discretizacion.ipynb

# Discretización por frecuencia

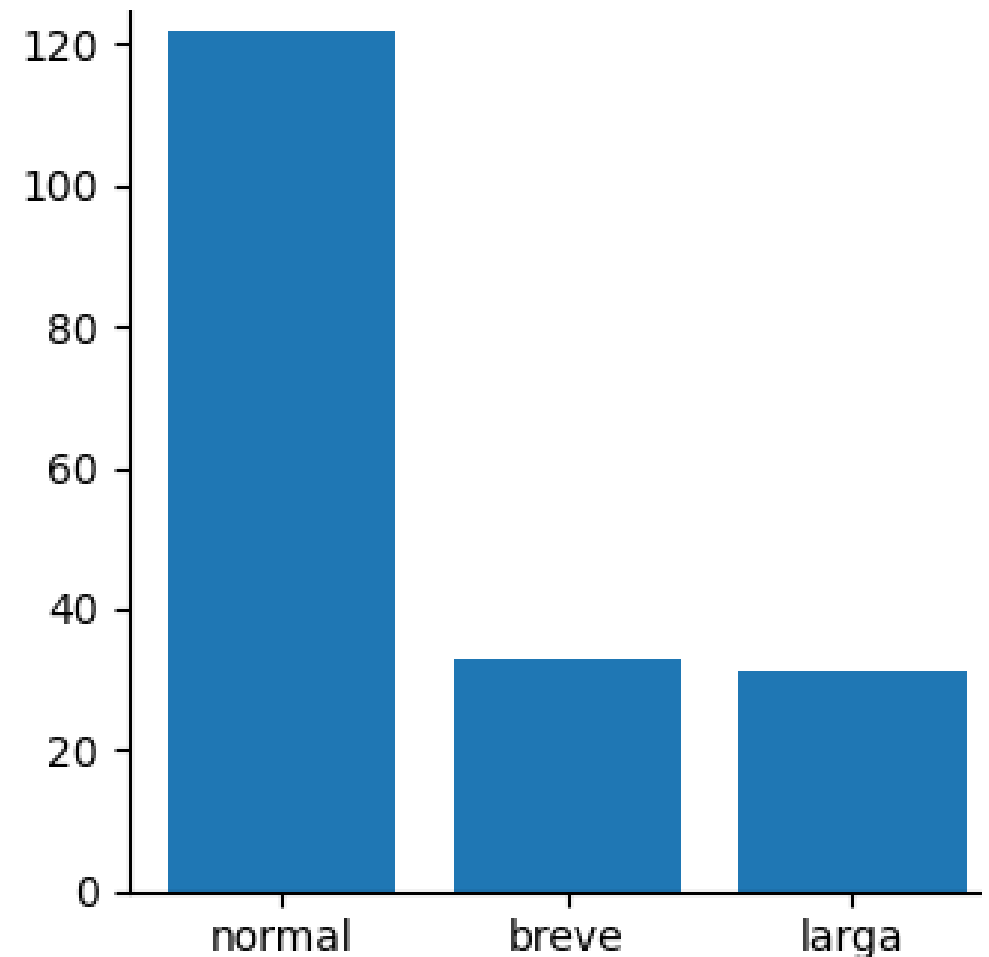
- DURATION discretizado en 4 intervalos de igual frecuencia





# Discretización especificada por el usuario

- Si  $DURATION \leq 100$ , BREVE
- Si  $(DURATION > 100)$  y  $(DURATION \leq 136)$ , NORMAL
- Si  $(DURATION > 136)$ , LARGA



# Discretización especificada por el usuario

```
import pandas as pd
import numpy as np
df= pd.read_csv('../Datos/Premios2020.csv',encoding='ISO-8859-1')

# Discretización indicada por el usuario
etiq = ["breve","normal","larga"]
valores = [-math.inf, 100, 136, math.inf]

columna = pd.cut(df["duration"],bins=valores,labels=etiq)

df['duration2']= pd.Series.to_frame(columna)

print(pd.value_counts(df['duration2']))
```

Discretizacion.ipynb

# Transformación de atributos

## □ DISCRETIZACION

- ▣ Algunos algoritmos de minería de datos sólo operan con atributos cualitativos. La discretización convierte los atributos numéricos en ordinales.

## □ NUMERIZACION

- ▣ Es el proceso contrario a la discretización. Convierte atributos cualitativos en numéricos.

## □ NORMALIZACION

- ▣ Permite expresar los valores de los atributos sin utilizar las unidades de medida originales facilitando su comparación y uso conjunto.

# Numerización

- En ocasiones los atributos nominales u ordinales deben convertirse en números.
- Para los nominales suele utilizarse una representación binaria y para los ordinales suele utilizarse una representación entera.
- Es importante considerar que si se numeran en forma correlativa los valores de un atributo nominal se agrega un orden que originalmente no está presente en la información disponible.

# Ejemplo

Premios2020.csv

◆ El archivo **Premios2020.csv** contiene 186 premios otorgados

Year	Age	Actor	Sex	Film	nominat	rating	duration	genre1	genre2	release	synopsis
1928	44	Emil Jannings	M	The Last Command	2	8	88	Drama	History	April	A former Imperial Russian gener
1928	22	Laura Gaior (aka Janet G	F	Sunrise	5	7.8	110	Drama	Romance		A street cleaner saves a young w
1929	37	Mary Pickford	F	Coquette	1	7.3	76	Drama	Romance	April	A flirtatious southern belle is co
...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...
2019	45	Joaquin Phoenix	M	Joker	11	8.5	122	Drama	Thriller	October	Arthur Fleck loves to make peop
2020	63	Frances McDormand	F	Nomadland	6	7.4	108	Drama		September	Nomadland es una película estac
2020	83	Anthony Hopkins	M	The father	6	8.3	97	Drama		January	Anthony tiene casi 83 años. Vive



Numerizacion.ipynb

Para las variables ordinales  
podemos utilizar la numerización  
de entero único

# Numerización – representación entera

```
import pandas as pd
import numpy as np
df= pd.read_csv('../Datos/Premios2020.csv',encoding='ISO-8859-1')
moda = df['release'].mode()
df['release'] = df['release'].replace([np.nan], moda)
print(pd.value_counts(df['release']))

mapeo = {"release": {"January":1, "February":2, "March":3,"April":4,
                    "May":5, "June":6,"July":7, "August":8, "September":9,
                    "October":10, "November":11, "December":12}}

df.replace(mapeo, inplace=True)
print(df['release'].describe())
```

**Numerizacion.ipynb**

# Numerización Binaria (dummy)

- La numerización binaria reemplaza al atributo nominal por tantos atributos numéricos binarios como valores distintos pueda tomar.
- Las denominaciones de estos nuevos atributos surgen de igualar el nombre original con cada uno de los posibles valores.
- Para un mismo ejemplo sólo uno de estos nuevos atributos tendrá valor 1 y el resto 0.

# Ejemplo

Premios2020.csv

◆ El archivo **Premios2020.csv** contiene 186 premios otorgados

Year	Age	Actor	Sex	Film	nominat	rating	duration	genre1	genre2	release	synopsis
1928	44	Emil Jannings	M	The Last Command	2	8	88	Drama	History	April	A former Imperial Russian gener
1928	22	Laura Gainor (aka Janet G	F	Sunrise	5	7.8	110	Drama	Romance		A street cleaner saves a young w
1929	37	Mary Pickford	F	Coquette	1	7.3	76	Drama	Romance	April	A flirtatious southern belle is co
...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...
2019	45	Joaquin Phoenix	M	Joker	11	8.5	122	Drama	Thriller	October	Arthur Fleck loves to make peop
2020	63	Frances McDormand	F	Nomadland	6	7.4	108	Drama		September	Nomadland es una película estac
2020	83	Anthony Hopkins	M	The father	6	8.3	97	Drama		January	Anthony tiene casi 83 años. Vive



Las variables nominales deben numerizarse utilizando una representación binaria

Numerizacion.ipynb



# Numerización binaria

```
import pandas as pd
import numpy as np
df= pd.read_csv('../Datos/Premios2020.csv', encoding='ISO-8859-1')

# atributo sexo con codificación binaria
NuevasColumnas = pd.get_dummies(df['Sex'], prefix= 'Sex')

# Agregamos las nuevas columnas al DataFrame
df = pd.concat([NuevasColumnas, df], axis=1)

# Borramos la columna anterior
df.drop('Sex',axis=1, inplace=True)
```

Numerizacion.ipynb

# Numerización Binaria de SEX

Row No.	Sex = M	Sex = F	Year	Age	Actor	Film	nominatio
1	1	0	1928	44	Emil Jannings	The Last Co...	2
2	0	1	1928	22	Laura Gainor ...	Sunrise	5
3	1	0	1929	38	Warner Baxter	In Old Arizona	5
4	0	1	1929	37	Mary Pickford	Coquette	2
5	1	0	1930	62	George Arliss	Disraeli	3
6	0	1	1930	30	Norma Shear...	The Divorcee	4
7	1	0	1931	53	Lionel Barry...	A Free Soul	3
8	0	1	1931	62	Marie Dressler	Min and Bill	2
9	1	0	1932	41	W. Beery(47)/...	The Champ/...	4
10	0	1	1932	32	Helen Hayes	Sin of Madelon	2

- Se dispone de información de pacientes afectados de rinitis alérgica:
  - **Age:** Edad
  - **Sex:** Sexo
  - **BP:** Presión sanguínea.
  - **Cholesterol:** Nivel de colesterol.
  - **Na:** Nivel de sodio en la sangre.
  - **K:** Nivel de potasio en la sangre.
  - **Drug:** fármaco suministrado (opciones DrugA, DrugB, DrugC, DrugX, DrugY)
- Se busca predecir si el tipo de fármaco que se debe administrar a un paciente afectado de rinitis alérgica es el habitual (DrugY) o no.

# Drug5.csv - Numerización

- Drug5.csv contiene 200 muestras de pacientes atendidos previamente

Nro.	Age	Sex	BP	Colesterol	Na	K	Drug
1	23	F	HIGH	HIGH	0,792535	0,031258	drugY
2	47	M	LOW	HIGH	0,739309	0,056468	drugC
3	47	M	LOW	HIGH	0,697269	0,068944	drugC
4	28	F	NORMAL	HIGH	0,563682	0,072289	drugX
5	61	F	LOW	HIGH	0,559294	0,030998	drugY
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
197	16	M	LOW	HIGH	0,743021	0,061886	drugC
198	52	M	NORMAL	HIGH	0,549945	0,055581	drugX
199	23	M	NORMAL	NORMAL	0,78452	0,055959	drugX
200	40	F	LOW	NORMAL	0,683503	0,060226	drugX

# Transformación de atributos

## □ DISCRETIZACION

- ▣ Algunos algoritmos de minería de datos sólo operan con atributos cualitativos. La discretización convierte los atributos numéricos en ordinales.

## □ NUMERIZACION

- ▣ Es el proceso contrario a la discretización. Convierte atributos cualitativos en numéricos.

## □ NORMALIZACION

- ▣ Permite expresar los valores de los atributos sin utilizar las unidades de medida originales facilitando su comparación y uso conjunto.

# Normalización

- Se aplica según el modelo que se va a construir.
- La más común es la **normalización lineal uniforme**

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- Es muy sensible a valores fuera de rango (outliers).
- Si se recortan los extremos se obtiene valor negativos y/o mayores a 1.

# Normalización Lineal Uniforme

```
import pandas as pd
import numpy as np

df= pd.read_csv('../Datos/Premios2020.csv', encoding='ISO-8859-1')

# -- Escala los valores entre 0 y 1 --
mini = df['Age'].min()
maxi = df['Age'].max()
df['AgeLineal']= (df['Age']-mini) / (maxi-mini)
```

Normalizacion.ipynb

# Normalización

- Existen otras transformaciones. Por ejemplo, si los datos tienen distribución normal se pueden **tipificar**

$$X' = \frac{X - \text{media}(X)}{\text{desviacion}(X)}$$

- De esta forma los datos se distribuyen normalmente alrededor de 0 con desviación 1.



# Normalización usando media y desvio

```
import pandas as pd
import numpy as np

df= pd.read_csv('../Datos/Premios2020.csv', encoding='ISO-8859-1')

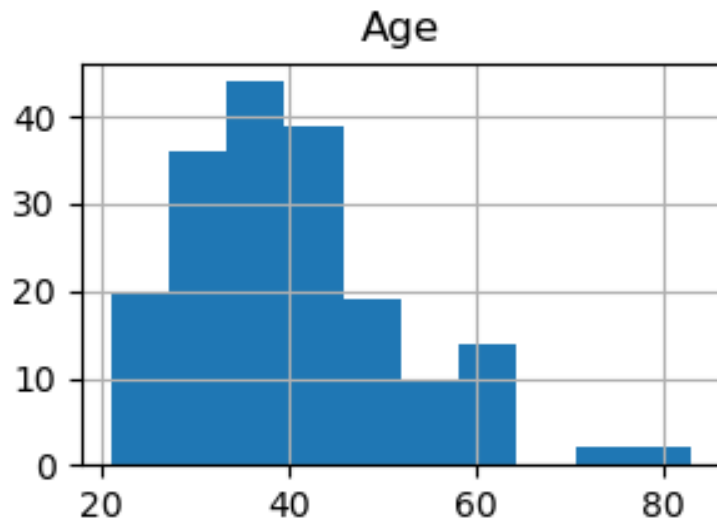
# -- Estandarización --
media = df['Age'].mean()
desvio = df['Age'].std()
df['AgeNorm']= (df['Age']-media)/desvio
```

Normalizacion.ipynb

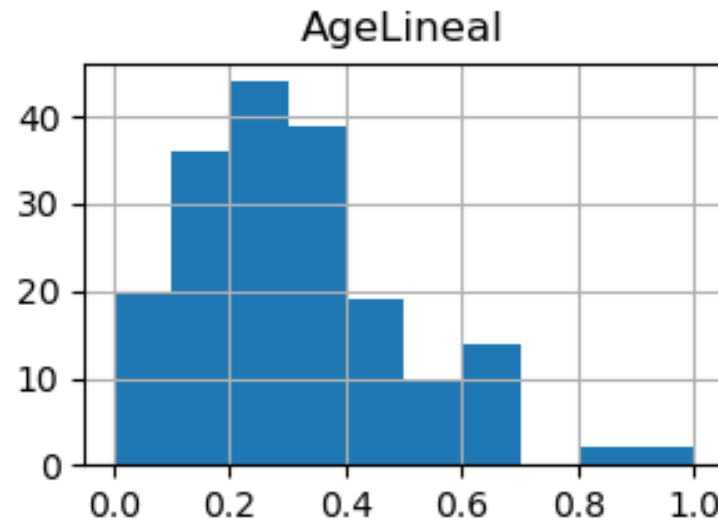
# Normalización del atributo AGE

```
plt.figure()  
df[['Age', 'AgeLineal', 'AgeNorm']].hist()
```

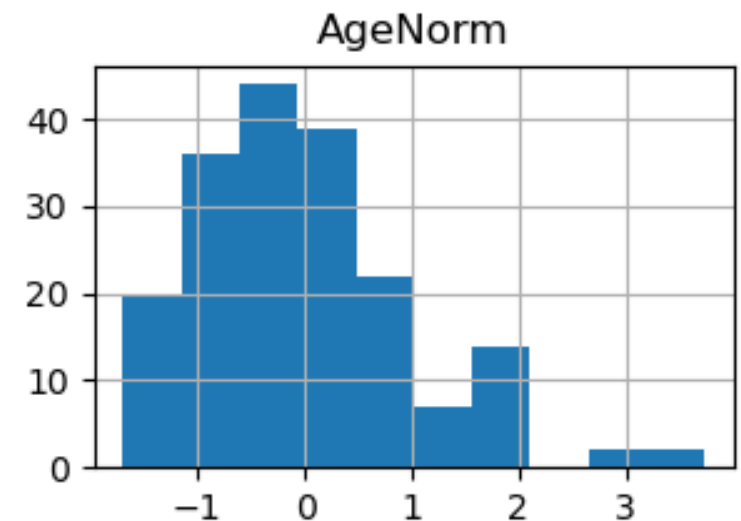
Normalizacion.ipynb



$X$



$$X' = \frac{X - \min(X)}{\max(X) - \min(X)}$$



$$X' = \frac{X - \text{media}(X)}{\text{desvio}(X)}$$

# Normalización del atributo AGE

```
mini = df['Age'].min()
maxi = df['Age'].max()
df['AgeLineal'] = (df['Age'] - mini) / (maxi - mini)

media = df['Age'].mean()
desvio = df['Age'].std()
df['AgeNorm'] = (df['Age'] - media) / desvio
```

	Age	AgeLineal	AgeNorm
count	186.0000	186.0000	186.0000
mean	40.3656	0.3123	-0.0000
std	11.4371	0.1845	1.0000
min	21.0000	0.0000	-1.6932
25%	32.2500	0.1815	-0.7096
50%	38.0000	0.2742	-0.2068
75%	45.7500	0.3992	0.4708
max	83.0000	1.0000	3.7277

```
round(df[['Age', 'AgeLineal', 'AgeNorm']].describe(), 4)
```

# Comparación de atributos numéricos

## ❑ Valores originales

	Year	Age	nominations	rating	duration
0	1928	44	2.0	8.0	88
1	1928	22	5.0	7.8	110
2	1929	37	1.0	7.3	76
3	1929	38	5.0	5.8	95
4	1930	62	3.0	6.5	90
...	...	...	...	...	...
181	2018	44	10.0	7.5	119
182	2019	50	2.0	6.8	118
183	2019	45	11.0	8.5	122
184	2020	63	6.0	7.4	108
185	2020	83	6.0	8.3	97

# Comparación de atributos numéricos

- Valores normalizados linealmente entre 0 y 1

	Year	Age	nominations	rating	duration
0	0.000	0.371	0.083	0.647	0.112
1	0.000	0.016	0.333	0.588	0.243
2	0.011	0.258	0.000	0.441	0.041
3	0.011	0.274	0.333	0.000	0.154
4	0.022	0.661	0.167	0.206	0.124
...	...	...	...	...	...
181	0.978	0.371	0.750	0.500	0.296
182	0.989	0.468	0.083	0.294	0.290
183	0.989	0.387	0.833	0.794	0.314
184	1.000	0.677	0.417	0.471	0.231
185	1.000	1.000	0.417	0.735	0.166

# Comparación de atributos numéricos

- Valores normalizados utilizando media y desvío

	Year	Age	nominations	rating	duration
0	-1.709	0.318	-1.433	0.622	-1.302
1	-1.709	-1.606	-0.423	0.242	-0.386
2	-1.672	-0.294	-1.769	-0.710	-1.802
3	-1.672	-0.207	-0.423	-3.567	-1.010
4	-1.635	1.892	-1.096	-2.234	-1.219
...	...	...	...	...	...
181	1.635	0.318	1.259	-0.330	-0.011
182	1.672	0.842	-1.433	-1.663	-0.052
183	1.672	0.405	1.595	1.574	0.114
184	1.709	1.979	-0.087	-0.520	-0.469
185	1.709	3.728	-0.087	1.194	-0.927

- El archivo **SEMILLAS.csv** contiene información de granos que pertenecen a tres variedades diferentes de trigo: Kama, Rosa y Canadiense.
  - ▣ área  $A$ ,
  - ▣ perímetro  $P$ ,
  - ▣ compacidad  $C = 4 * \pi * A / P^2$ ,
  - ▣ longitud del núcleo,
  - ▣ ancho del núcleo,
  - ▣ coeficiente de asimetría
  - ▣ longitud del surco del núcleo

# Resumen

## PREPARACION DE LOS DATOS

- Completar datos faltantes
- Generación de características o atributos nuevos
- Reducción de valores en atributos cualitativos
- Transformaciones
  - ▣ Discretización por rango, por frecuencia e indicada por el usuario
  - ▣ Numerización: codificación entera y codificación binaria
  - ▣ Normalización: Lineal y Estandarización