



UNIVERSIDAD
NACIONAL
DE COLOMBIA

CLASIFICACIÓN Y RECONOCIMIENTO DE PATRONES

Extracción de Características y Reducción de la Dimensionalidad

Jorge E. Espinosa

Profesor

Departamento de Ciencias de la Computación y de la Decisión

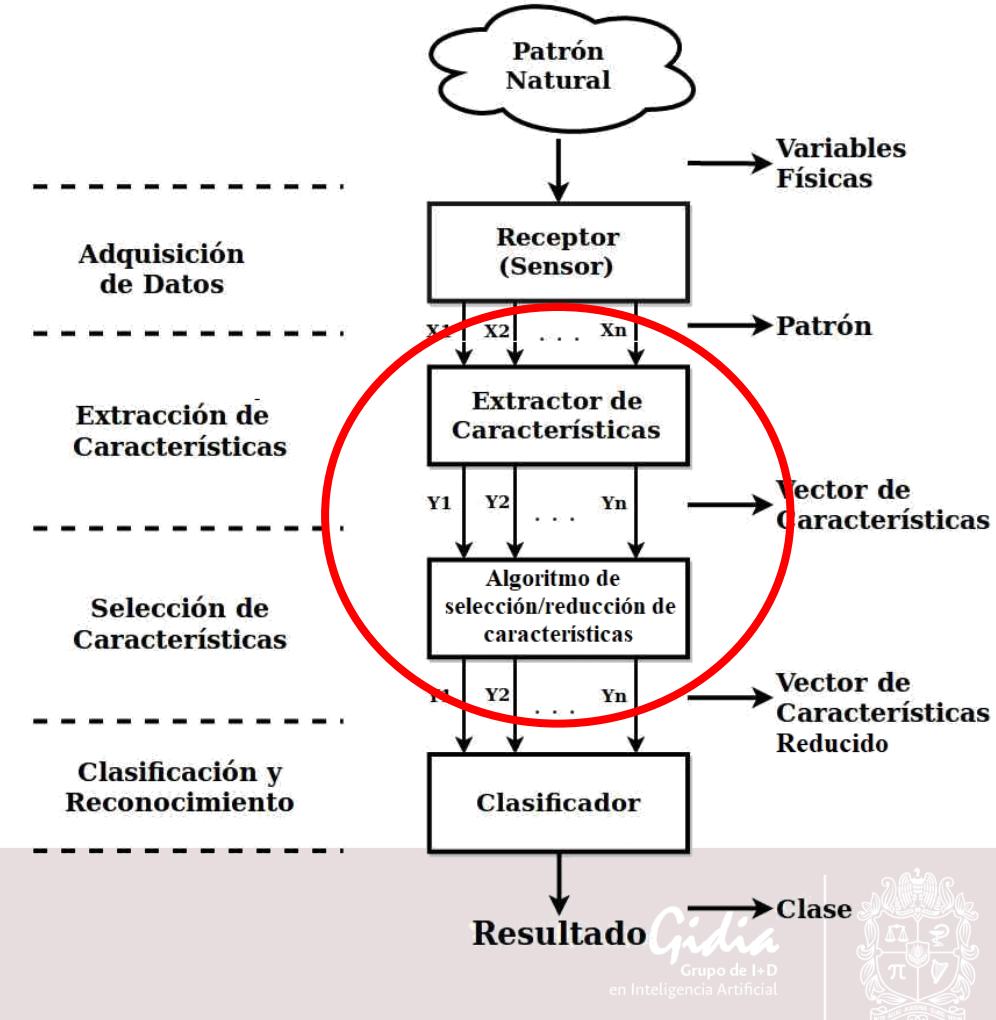
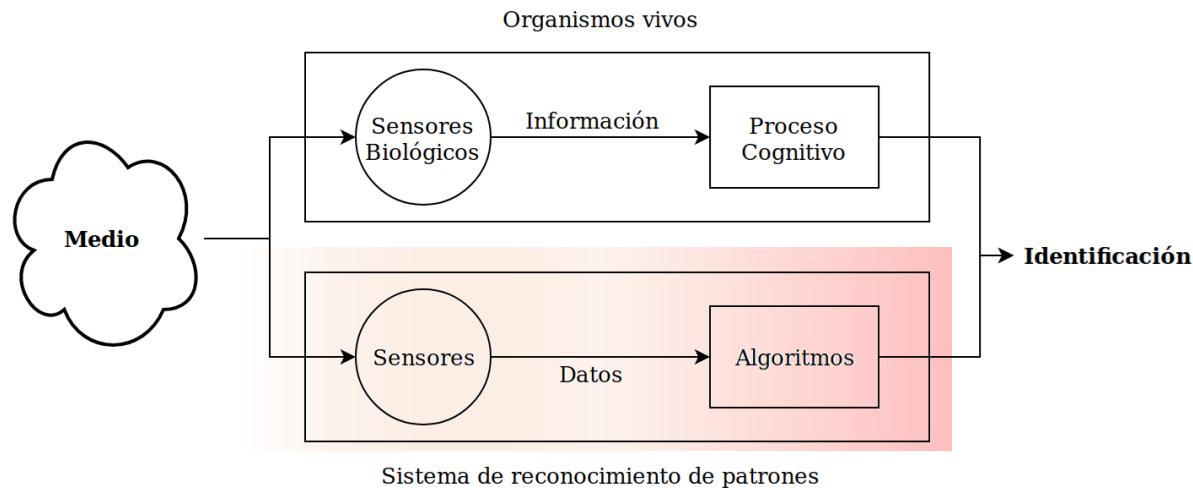
Investigador del Grupo de I+D en Inteligencia Artificial – GIDIA

jeespinosao@unal.edu.co

Contenido

1. Generalidades
2. Por qué la Reducción de la Dimensionalidad ?
3. Principales técnicas de reducción de la dimensionalidad
 - a. Selección de Características
 - b. Extracción de Características
4. Reducción de la dimensionalidad en Python
 - a. Selección Vs Extracción
5. t-SNE
6. Lab 1 ☺
7. La Maldición de la Dimensionalidad (The curse of dimensionality)
8. Correlación de Valores
9. Lab 2 ☺

Etapas Sistema de Clasificación y Reconocimiento de Patrones

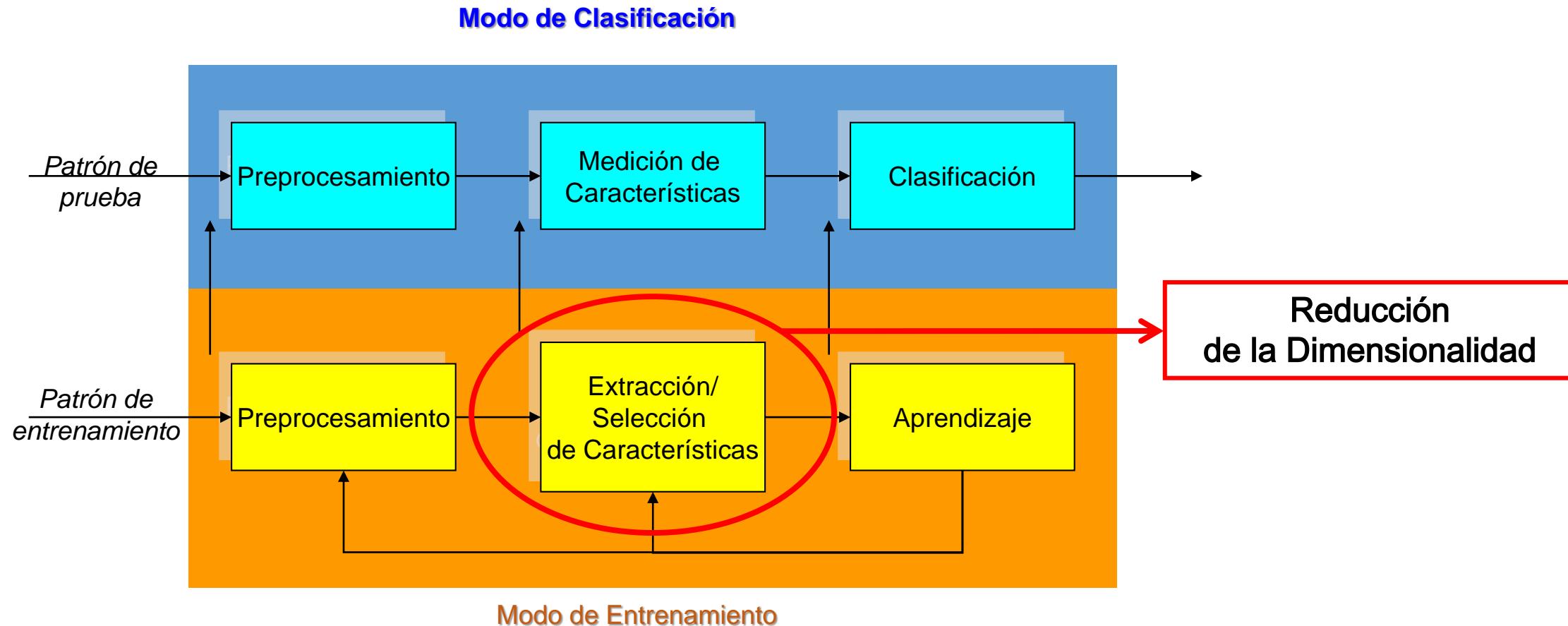


Representación de un Patrón

- Un patrón es representado por un conjunto de *d* **características**, o atributos, visto como un vector *d*-dimensional *vector de características*.

$$\mathbf{x} = (x_1, x_2, \dots, x_d)^T$$

Dos procesos para la Implementación de sistemas de reconocimiento de Patrónes



Por qué la Reducción de la Dimensionalidad ?

- En la actualidad es muy fácil **recopilar Datos**
 - Un experimento
- Los datos **no son únicamente recopilados para análisis**
- Los **datos se acumulan con una velocidad sin precedentes**
- El **preprocesamiento de los datos** (Reduciendo Dimensionalidad?) resulta *efectivo* en el reconocimiento de patrones y el aprendizaje automático
- La **reducción de la dimensionalidad** es una técnica efectiva en la optimización de la información

Por qué la Reducción de la Dimensionalidad ?

- La mayoría de las técnicas de aprendizaje automático y de reconocimiento de patrones **pueden no ser efectivas** en datos con alta dimensionalidad
 - **La Maldición de la dimensionalidad**
 - La precisión y la eficiencia de la consulta se degradan rápidamente a medida que aumenta la dimensión.
- La dimensión **intrínseca** puede ser pequeña.
 - Por ejemplo, el número de genes responsables de cierto tipo de enfermedad puede ser pequeño.

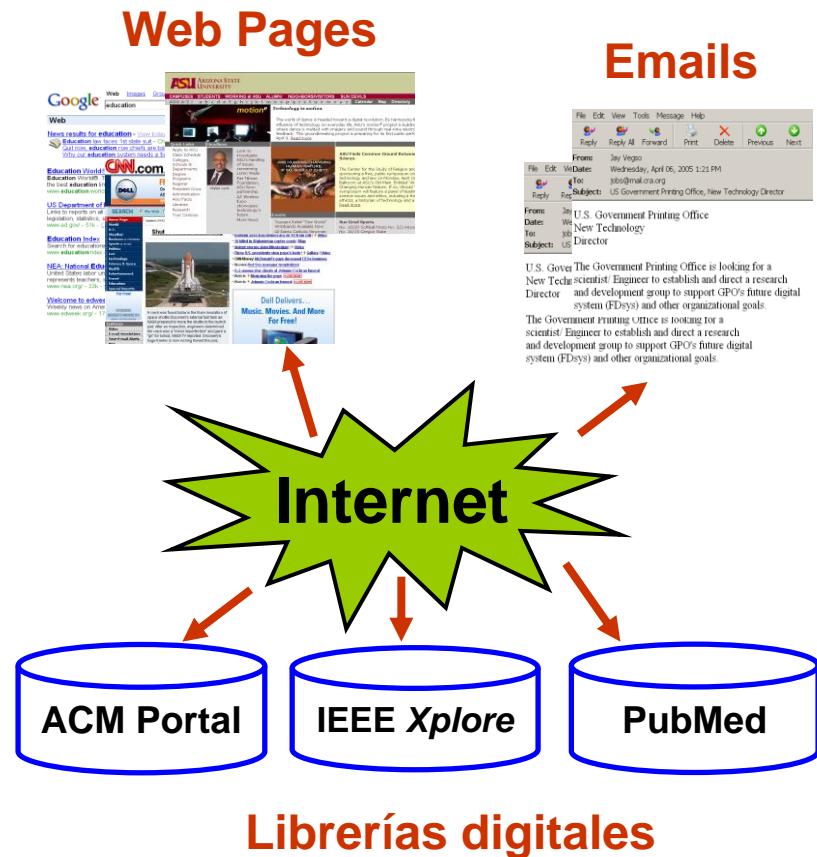
Por qué la Reducción de la Dimensionalidad ?

- **Visualización:** proyección de datos multidimensionales en 2D o 3D.
- **Compresión de datos:** almacenamiento y recuperación eficientes.
- **Eliminación de ruido:** efecto positivo en la precisión de la consulta.

Aplicaciones de la Reducción de la Dimensionalidad

- Gestión de la relación con el cliente
- Extracción de textos
- Recuperación de imágenes
- Análisis de datos de microarrays
- Clasificación de proteínas
- Reconocimiento facial
- Reconocimiento de dígitos escritos a mano
- Detección de intrusiones

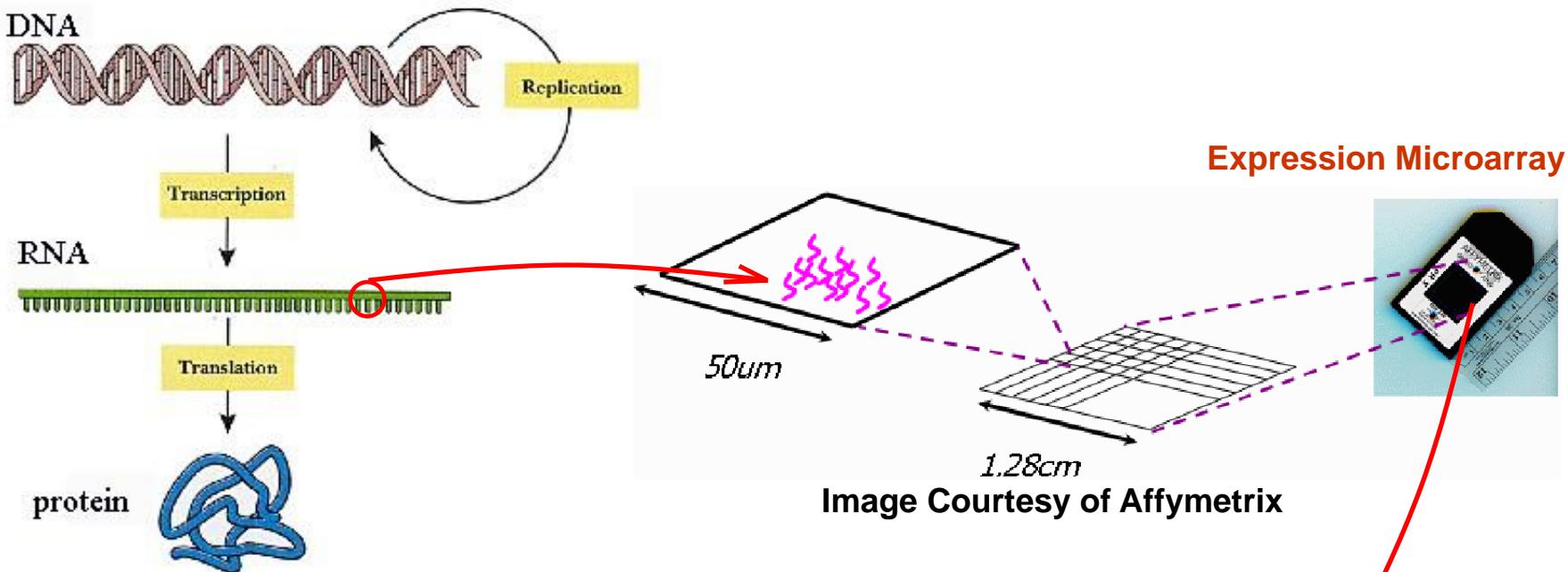
Clasificación de documentos



Términos		C	
T_1	T_2		T_N
12	0	6
D ₁	D ₂	Sports
3	10	Travel
D _M	0	Jobs
11	16

- **Tarea:** clasificar documentos sin etiqueta en categorías
- **Desafío:** miles de términos
- **Solución:** Aplicar reducción de la dimensionalidad

Análisis de microarrays de expresión genética



- **Tarea:** Clasificar nuevas muestras en tipos de enfermedades conocidas (diagnóstico de enfermedades)
- **Desafío:** miles de genes, pocas muestras
- **Solución:** aplicar reducción de dimensionalidad

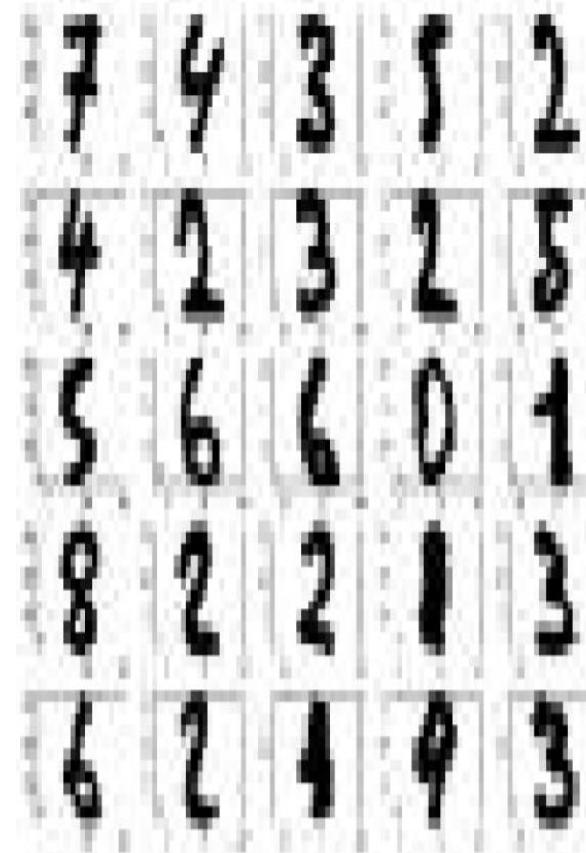
Sample \ Gene	M23197_at	U66497_at	M92287_at	...	Class
Sample 1	261	88	4778	...	ALL
Sample 2	101	74	2700	...	ALL
Sample 3	1450	34	498	...	AML
.
.

Expression Microarray Data Set

Otros tipos de datos multidimensionales



Imágenes de rostros



Dígitos manuscritos

Principales técnicas de reducción de dimensionalidad

- Selección de características
 - Definición
 - Objetivos
- Extracción de características (reducción)
 - Definición
 - Objetivos
- Diferencias entre las dos técnicas

Selección de características

- **Definición**

- Un proceso que elige un subconjunto óptimo de características de acuerdo con una función objetivo

- **Objetivos:**

- Para reducir la dimensionalidad y eliminar el ruido.
- Para mejorar el desempeño del reconocimiento de patrones
 - Velocidad de aprendizaje
 - Precisión predictiva
 - Simplicidad y comprensibilidad de los resultados de reconocimiento de patrones.

Extracción de Características (Reducción)

- La reducción de características se refiere al mapeo de los datos originales de alta dimensión en un espacio de menor dimensión
- Dado un conjunto de puntos de datos de p variables $\{x_1, x_2, \dots, x_n\}$
Computar su representación en una dimensión más reducida:

$$x_i \in \Re^d \rightarrow y_i \in \Re^p \quad (p \ll d)$$

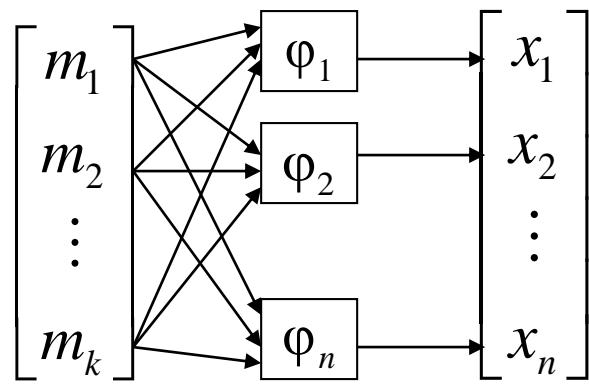
- El criterio para la reducción de características puede ser diferente de acuerdo a las diferentes configuraciones de problemas.
 - Configuración **no supervisada**: minimice la pérdida de información
 - Entorno **supervisado**: maximizar la discriminación de clase

Reducción de características vs. Selección de características

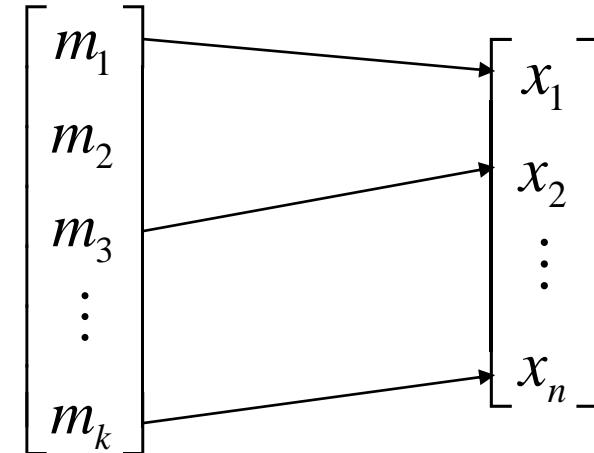
- **Reducción de características**
 - Se utilizan todas las características originales.
 - Las características transformadas son combinaciones lineales (en la mayoría de los casos) de las características originales.
- **Selección de las características**
 - Solo se selecciona un subconjunto de las características originales.

Métodos de extracción de Características

Extracción de Características



Selección de Características



El problema puede expresarse como la optimización de los parámetros $\varphi(\theta)$ del extractor de características.

Métodos supervisados: la función objetivo es un criterio de separabilidad (discriminabilidad) de ejemplos etiquetados, por ejemplo, análisis de discriminación lineal (LDA).

Métodos no supervisados: se busca una representación dimensional más baja que conserve las características importantes de los datos de entrada, por ejemplo, análisis de componentes principales (PCA).

Datos Organizados

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1

Datos Organizados

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1

Datos Organizados

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1

Datos Organizados (comando shape)

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1

```
pokemon_df.shape
```

(5, 7)

Cuando Utilizamos Reducción de Características

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1

Datos Organizados (comando describe())

	HP	Attack	Defense	Speed	Generation
count	5.0	5.0	5.0	5.0	5.0
mean	56.4	61.8	59.2	66.0	1.0
std	15.9	13.0	15.4	14.7	0.0
min	39.0	49.0	43.0	45.0	1.0
25%	45.0	52.0	49.0	60.0	1.0
50%	58.0	62.0	58.0	65.0	1.0
75%	60.0	64.0	63.0	80.0	1.0
max	80.0	82.0	83.0	80.0	1.0

```
pokemon_df.describe()
```

Datos Organizados (comando describe())

	HP	Attack	Defense	Speed	Generation
count	5.0	5.0	5.0	5.0	5.0
mean	56.4	61.8	59.2	66.0	1.0
std	15.9	13.0	15.4	14.7	0.0
min	39.0	49.0	43.0	45.0	1.0
25%	45.0	52.0	49.0	60.0	1.0
50%	58.0	62.0	58.0	65.0	1.0
75%	60.0	64.0	63.0	80.0	1.0
max	80.0	82.0	83.0	80.0	1.0

```
pokemon_df.describe()
```

Selección de Características

income	age	favorite color
--------	-----	----------------

10000	18	Black
-------	----	-------

50000	47	Blue
-------	----	------

20000	40	Blue
-------	----	------

30000	29	Green
-------	----	-------

20000	22	Purple
-------	----	--------

Selección de Características

income	age	favorite color
10000	18	Black
50000	47	Blue
20000	40	Blue
30000	29	Green
20000	22	Purple

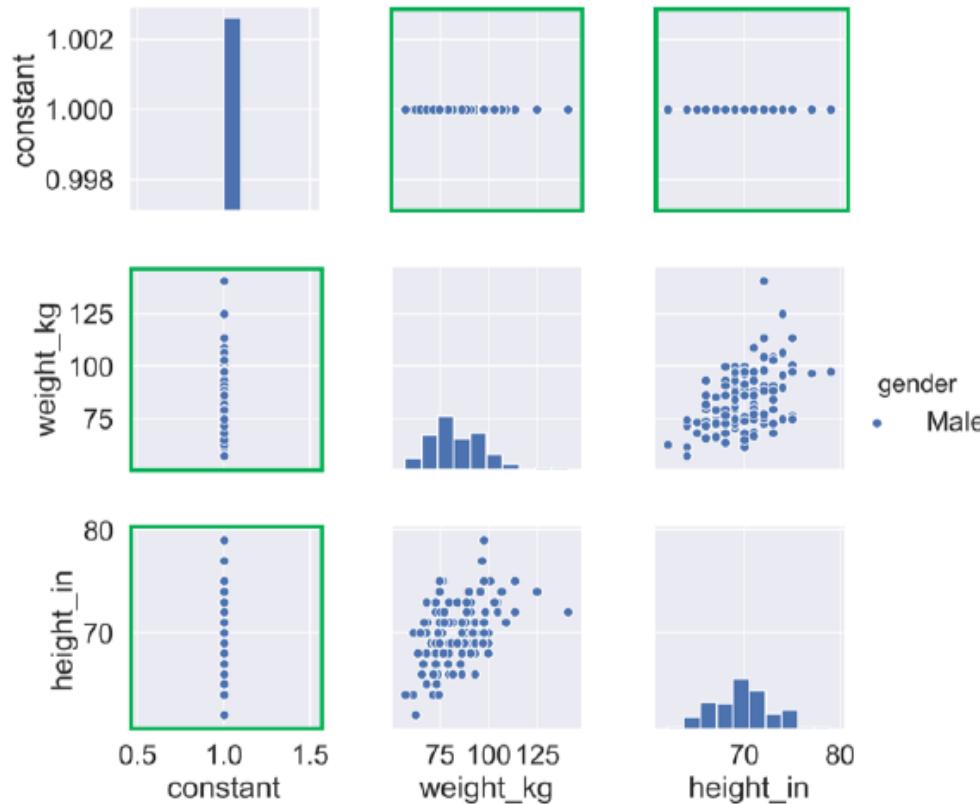


income	age
10000	18
50000	47
20000	40
30000	29
20000	22

```
insurance_df.drop('favorite color', axis=1)
```

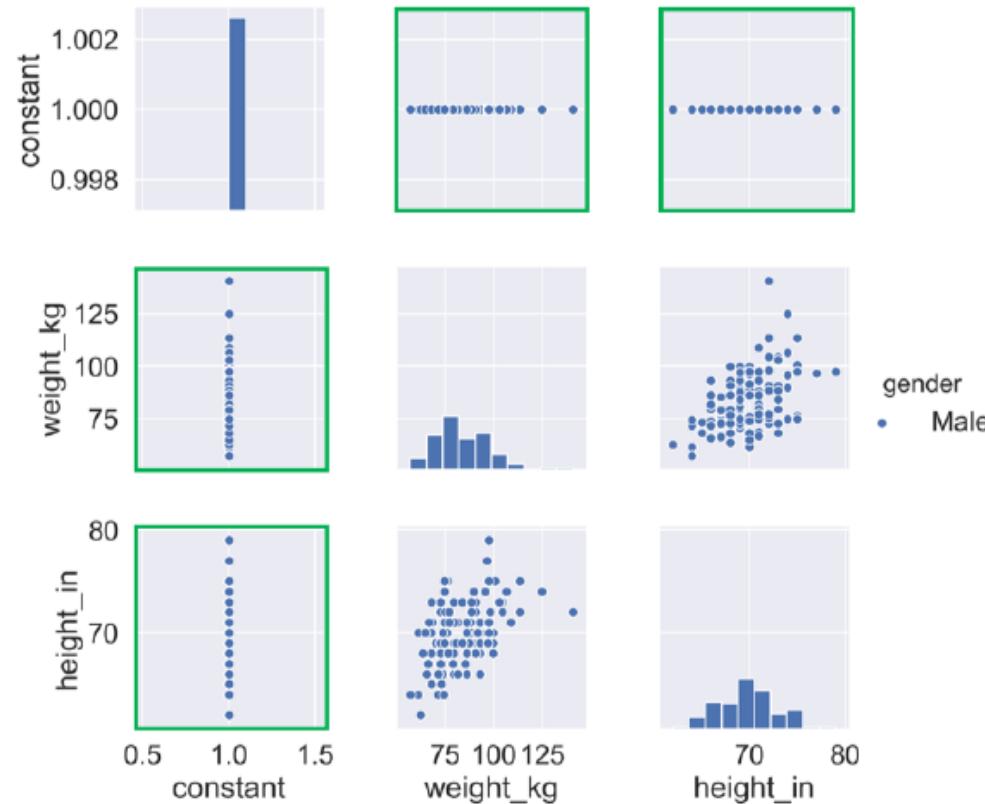
Selección de Características a partir del análisis de los datos

```
sns.pairplot(ansur_df, hue="gender", diag_kind='hist')
```

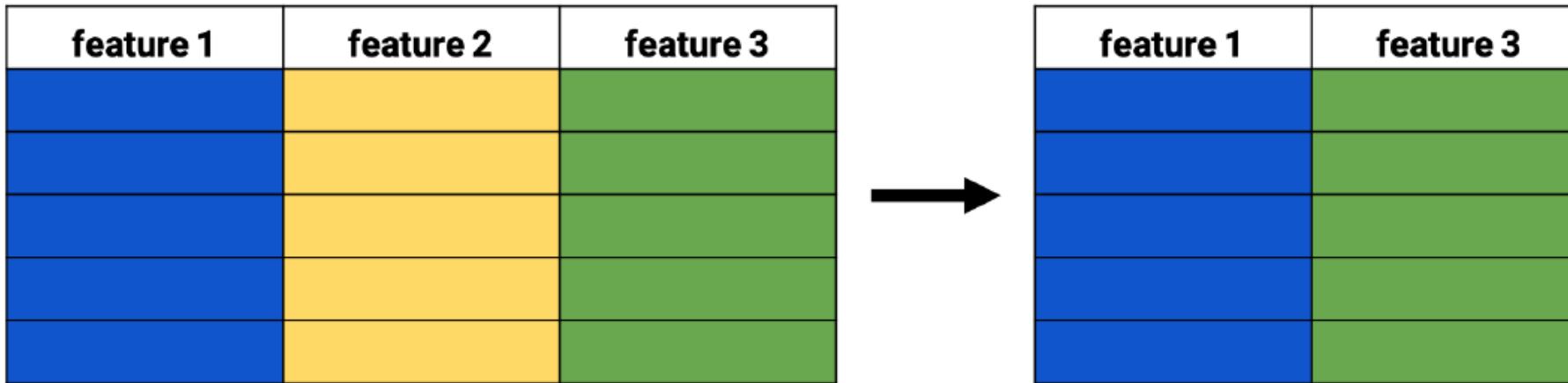


Selección de Características a partir del análisis de los datos

```
sns.pairplot(ansur_df, hue="gender", diag_kind='hist')
```



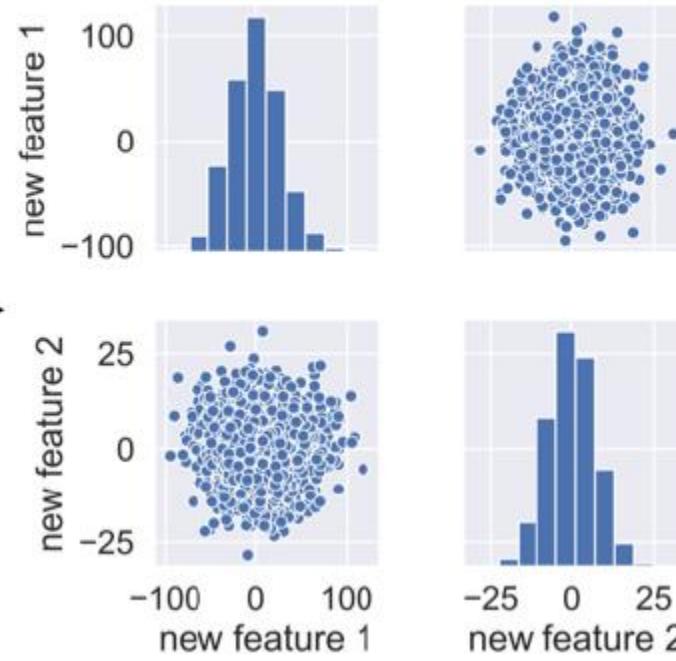
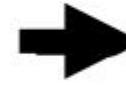
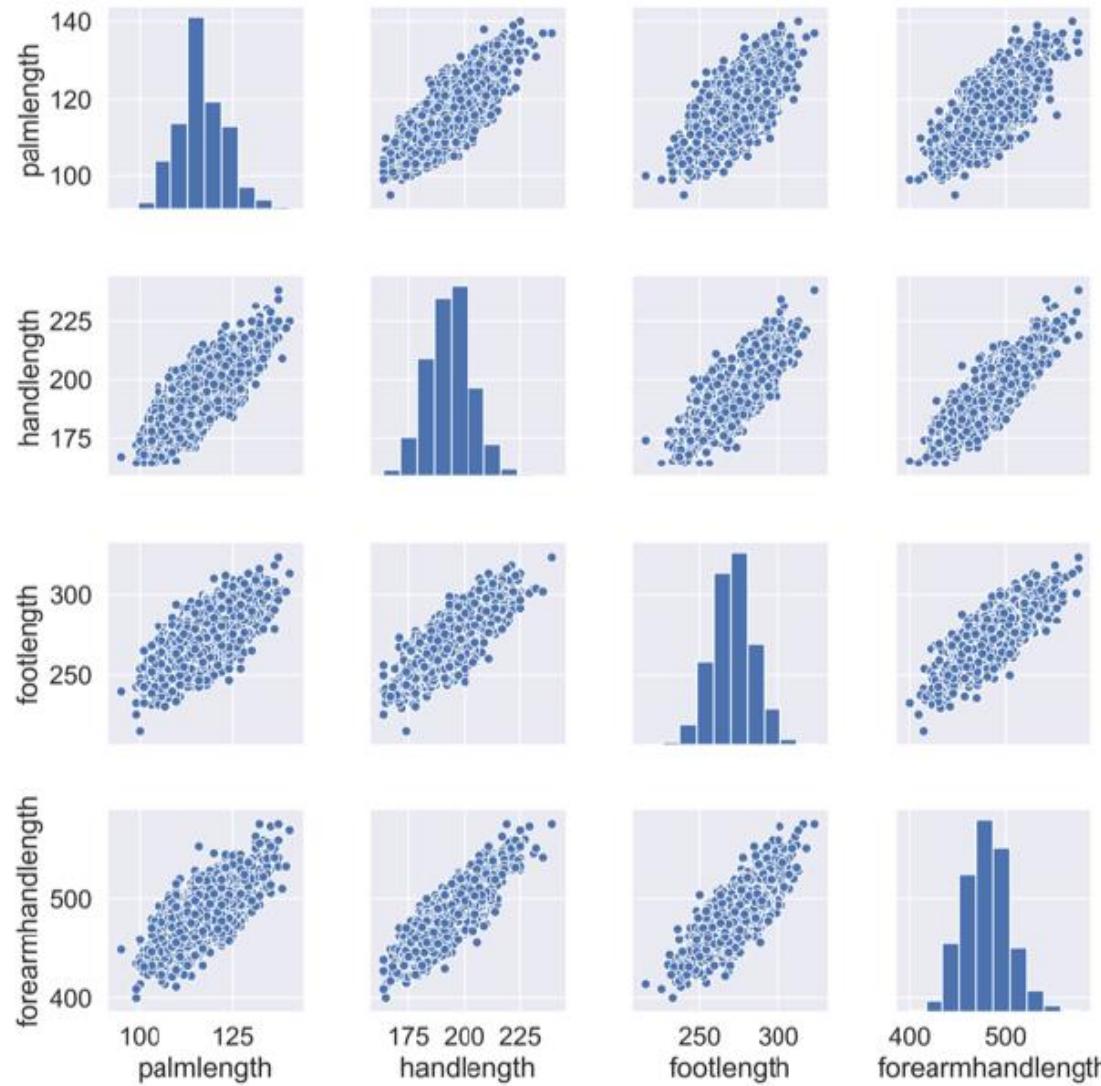
Selección de Características



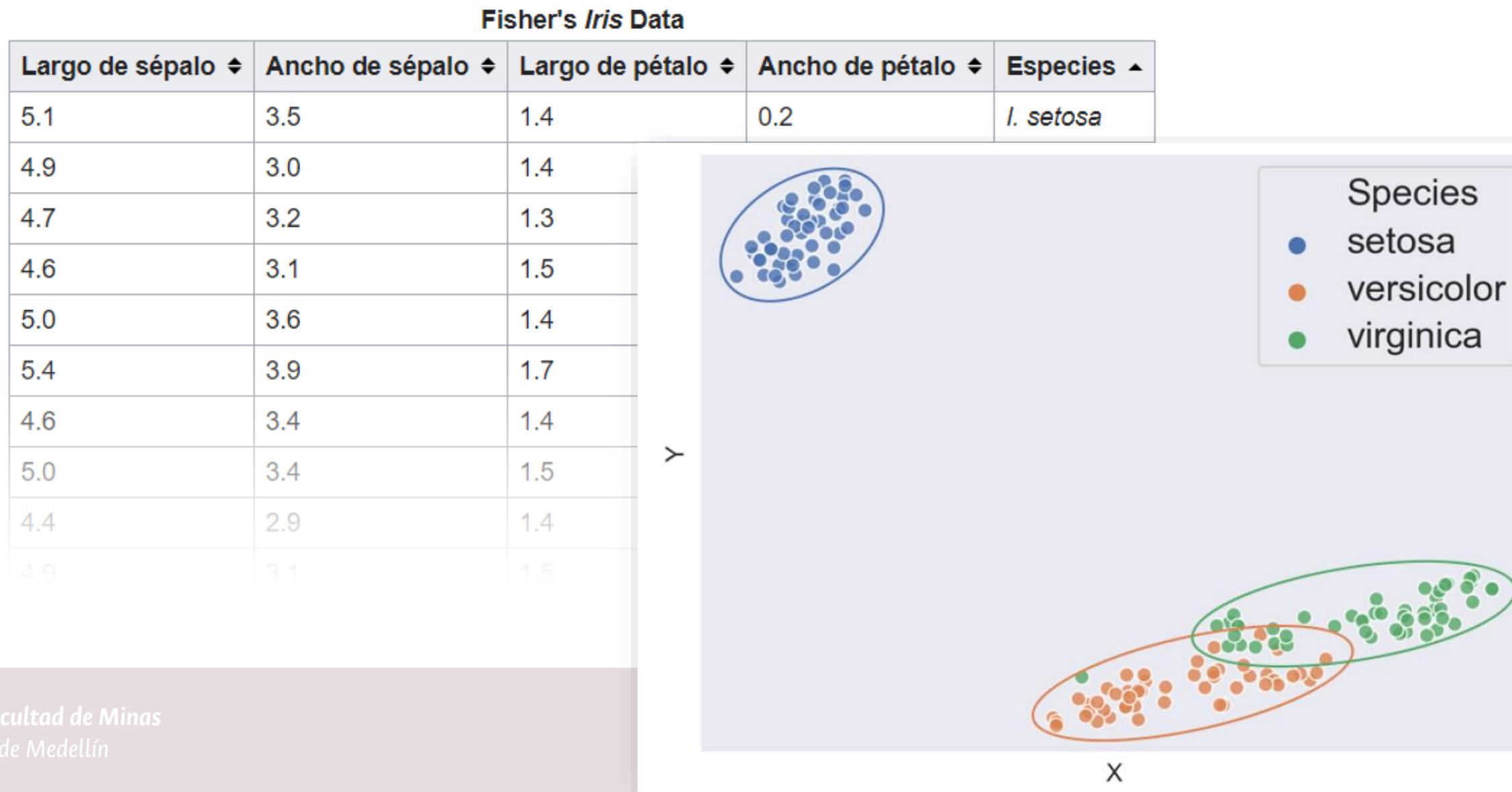
Extracción de Características



Extracción de Características a partir del análisis de los datos



Como hacemos para visualizar datos multidimensionales?



t-SNE resulta ideal para visualizar datos en 2D o 3D

= *t-distributed stochastic neighbor embedding*

(*incrustación vecina estocástica distribuida en t*)

t-SNE intenta reproducir vecindades de datos de alta D en una imagen 2D o 3D mediante:

1. Definir una distribución de probabilidad sobre pares de objetos de alta D tal que los objetos "similares" tienen una alta probabilidad de ser recogidos, mientras que "diferentes" los objetos tienen una probabilidad extremadamente pequeña de ser recogidos.
2. Definir una distribución de probabilidad similar sobre los puntos en el mapa de dimensionalidad reducida.
3. Minimizando la divergencia Kullback-Leibler entre las dos distribuciones variando las ubicaciones de los puntos en el mapa de baja D, es decir
minimizar:

$$\sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

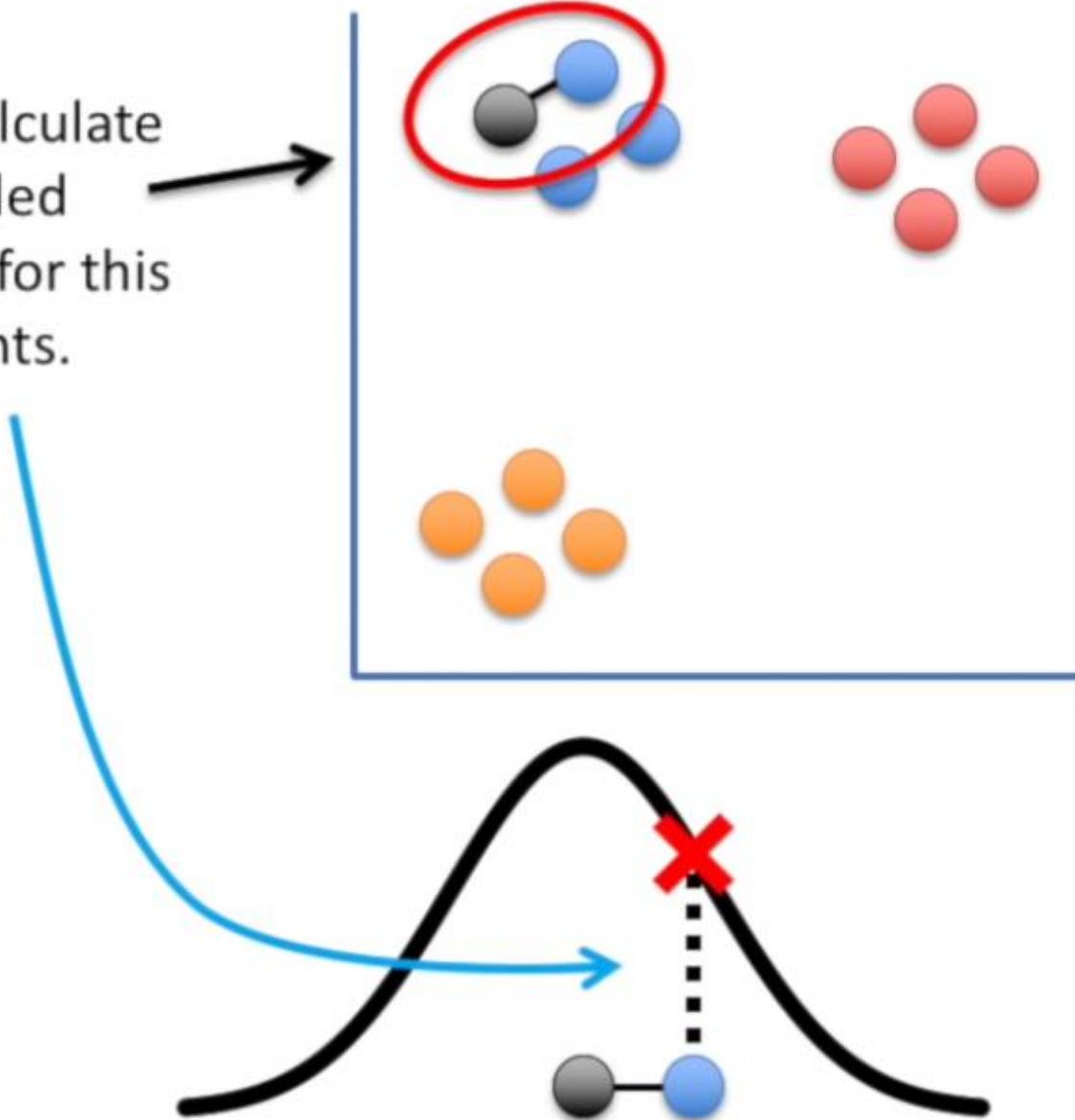
Probabilidad de que i y j sean cercanos en el espacio multidimensional

Probabilidad de que i y j sean cercanos en el espacio reducido

Suma sobre todos los puntos

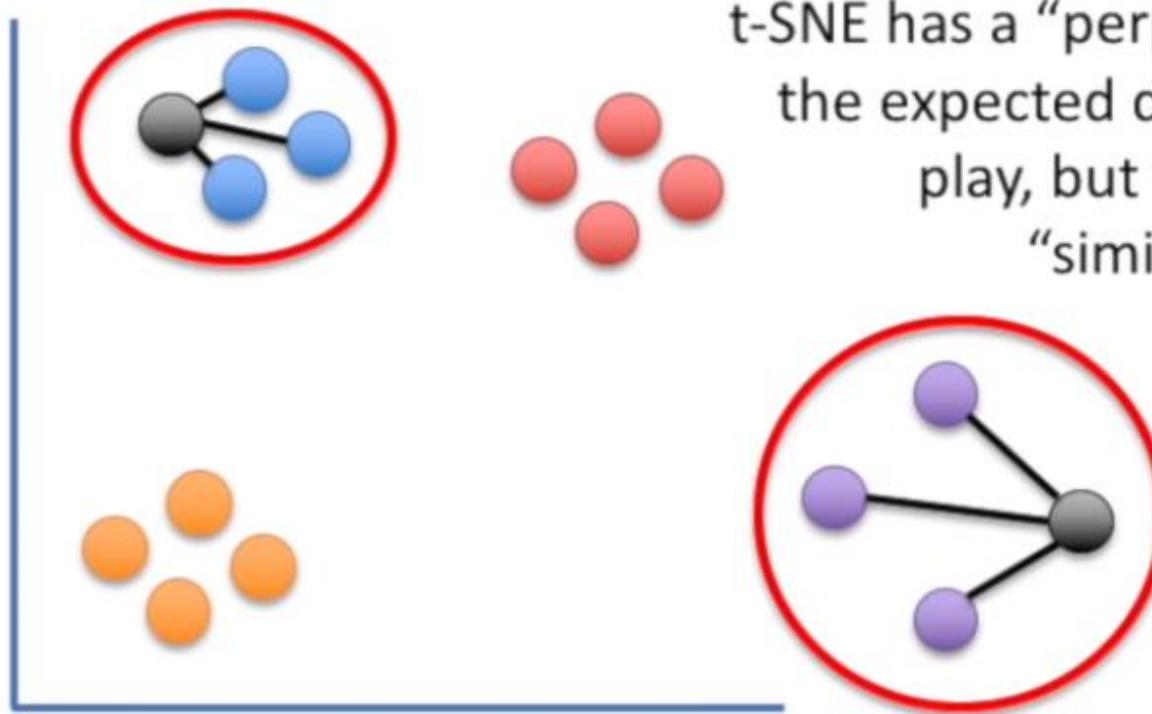
Como funciona T-SNE

Now we calculate
the “unscaled
similarity” for this
pair of points.



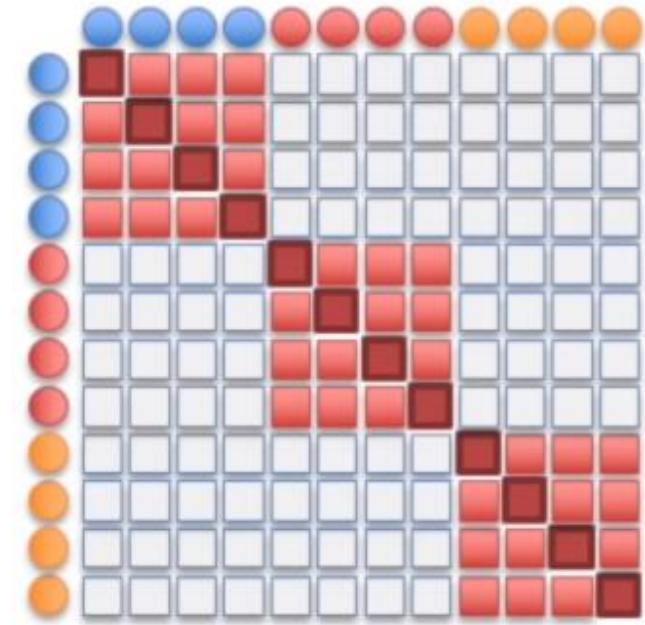
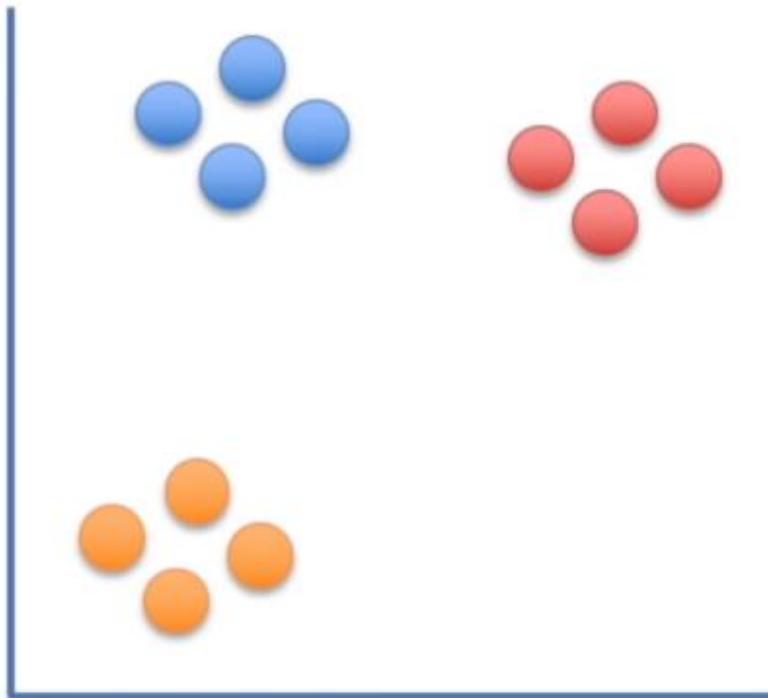
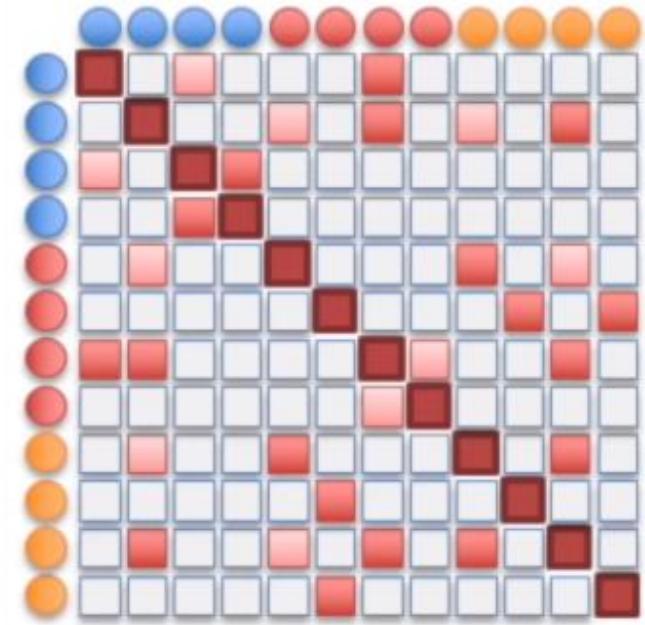
Como funciona T-SNE

The reality is a little more complicated, but only slightly.



t-SNE has a “perplexity” parameter equal to the expected density, and that comes into play, but these clusters are still more “similar” than you might expect.

Como funciona T-SNE



t-SNE moves the points a little bit at a time, and each step it chooses a direction that makes the matrix on the left more like the matrix on the right.



Laboratorio 1

<https://github.com/srobles05/CRP-2019S2/>

Exploración de Datos Multidimensionales

Este Laboratorio es Basado en el Curso de Dimensionality Reduction de DATACAMP®

En este laboratorio se presentará el concepto de reducción de dimensionalidad y aprenderá cuándo y por qué esto es importante. Aprenderá la diferencia entre la selección de características y la extracción de características y aplicará ambas técnicas para la exploración de datos. Al final se termina con una lección sobre t-SNE, una poderosa técnica de extracción de características que le permitirá visualizar un conjunto de datos de alta dimensión.

Encontrar el número de dimensiones en un conjunto de datos

Se ha cargado una muestra más grande del conjunto de datos de Pokemon como el marco de datos de Pandas `pokemon_df`.

```
: # import required Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
pokemon_df=pd.read_csv("pokemon-c.csv")#reading a dataset in a dataframe using pandas
```

¿Cuántas dimensiones o columnas hay en este conjunto de datos?

```
: pokemon_df.shape
for col in pokemon_df.columns:
    print(col)
```

```
HP
Attack
Defense
Generation
Name
Type
Legendary
```

```
: pokemon_df.shape
: (160, 7)
```

Eliminar características sin variación

Se ha cargado una muestra del conjunto de datos de Pokemon como `pokemon_df`. Para tener una idea de qué características tienen poca variación, debe usar el Shell de IPython para calcular estadísticas de resumen en esta muestra. Luego ajuste el código para crear un conjunto de datos más pequeño y fácil de entender. Utilice el `.describe()` método para buscar la función numérica sin variación y elimine su nombre de la lista asignada a `number_cols`.

pokemon_df.describe()

Labs-ExploracionMultiD

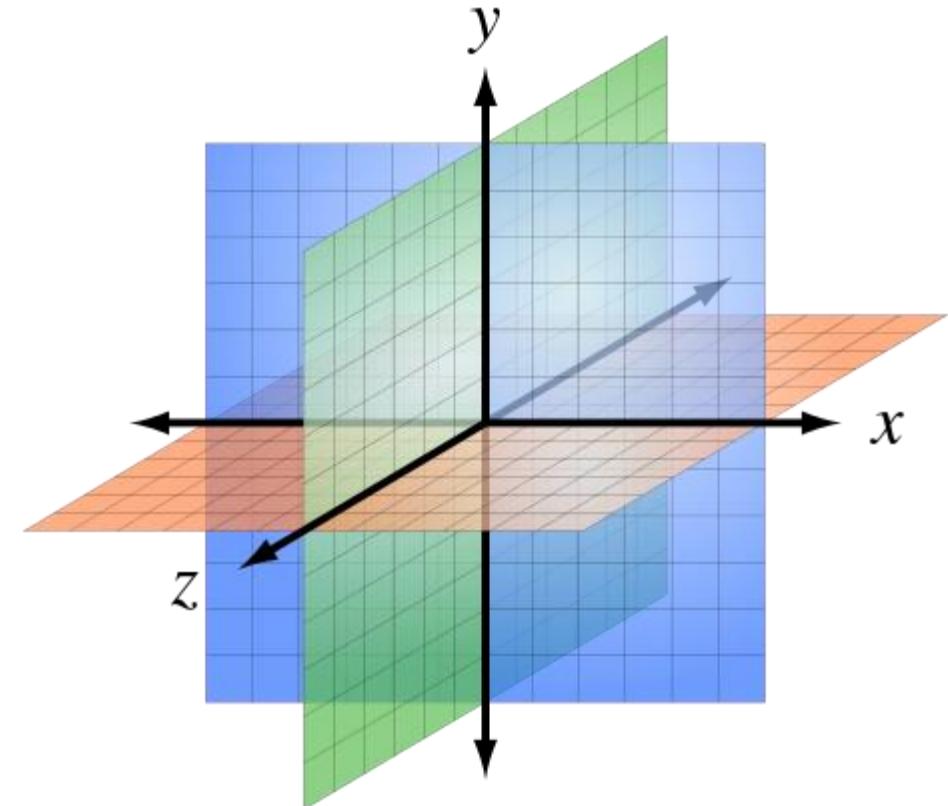


La maldición de la dimensionalidad

(The Curse of Dimensionality)

Cuando el número de características es muy grande en relación con el número de observaciones en su conjunto de datos, ciertos algoritmos luchan por entrenar modelos efectivos.

Esto se llama la "**Maldición de la dimensionalidad**", y es especialmente relevante para los algoritmos de agrupamiento que se basan en cálculos de distancia.



La maldición de la dimensionalidad

(Analogía)

“Digamos que tiene una línea recta de 100 metros de largo y dejó caer una moneda en algún lugar. No sería muy difícil de encontrar. Caminas a lo largo de la línea y te lleva dos minutos.

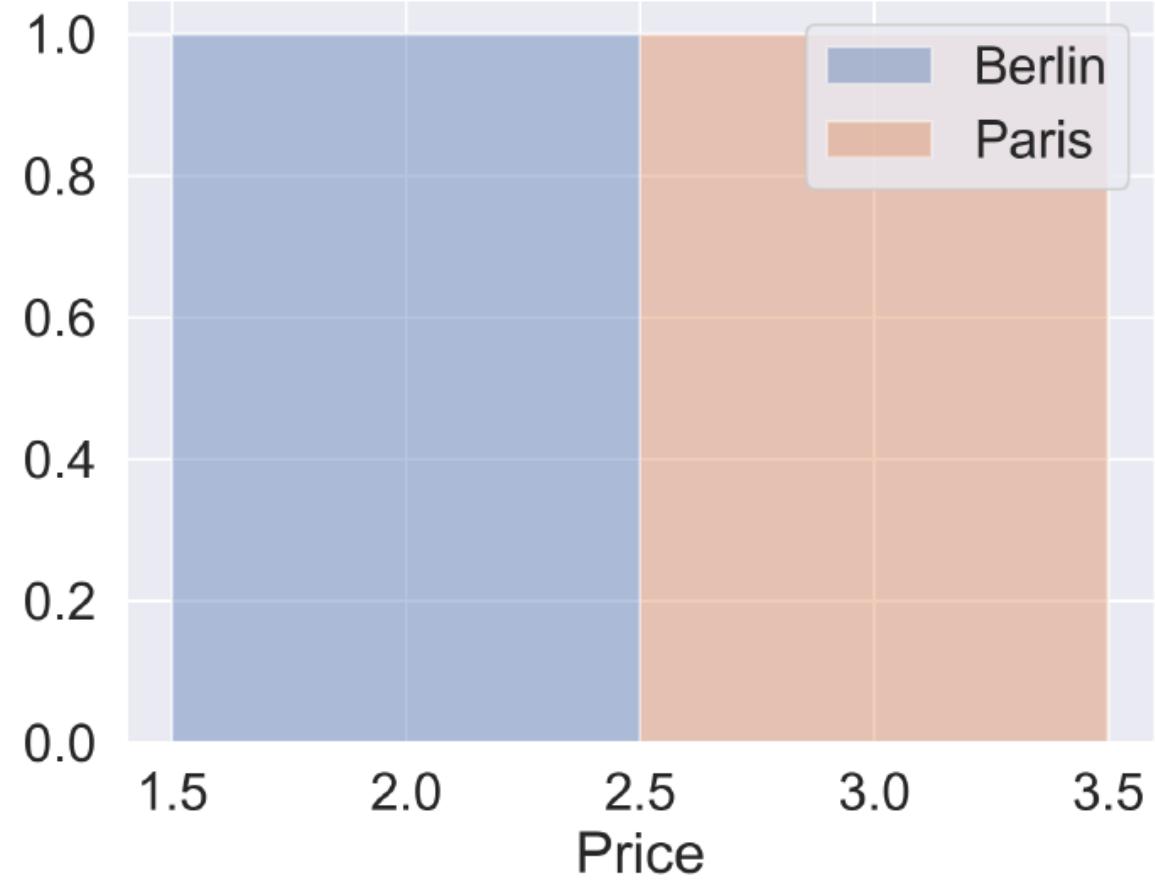
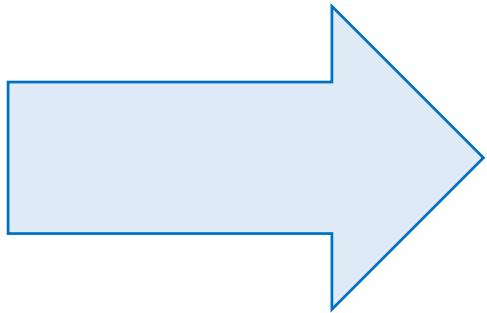
Ahora supongamos que tiene un cuadrado de 100 metros a cada lado y dejó caer un centavo en algún lugar. Sería bastante difícil, como buscar en dos campos de fútbol unidos. Podría llevar días.

Ahora un cubo de 100 metros de arista. Es como buscar en un edificio de 30 pisos del tamaño de un estadio de fútbol. Ugh

La dificultad de buscar en el espacio se vuelve mucho más difícil a medida que tienes más dimensiones.”

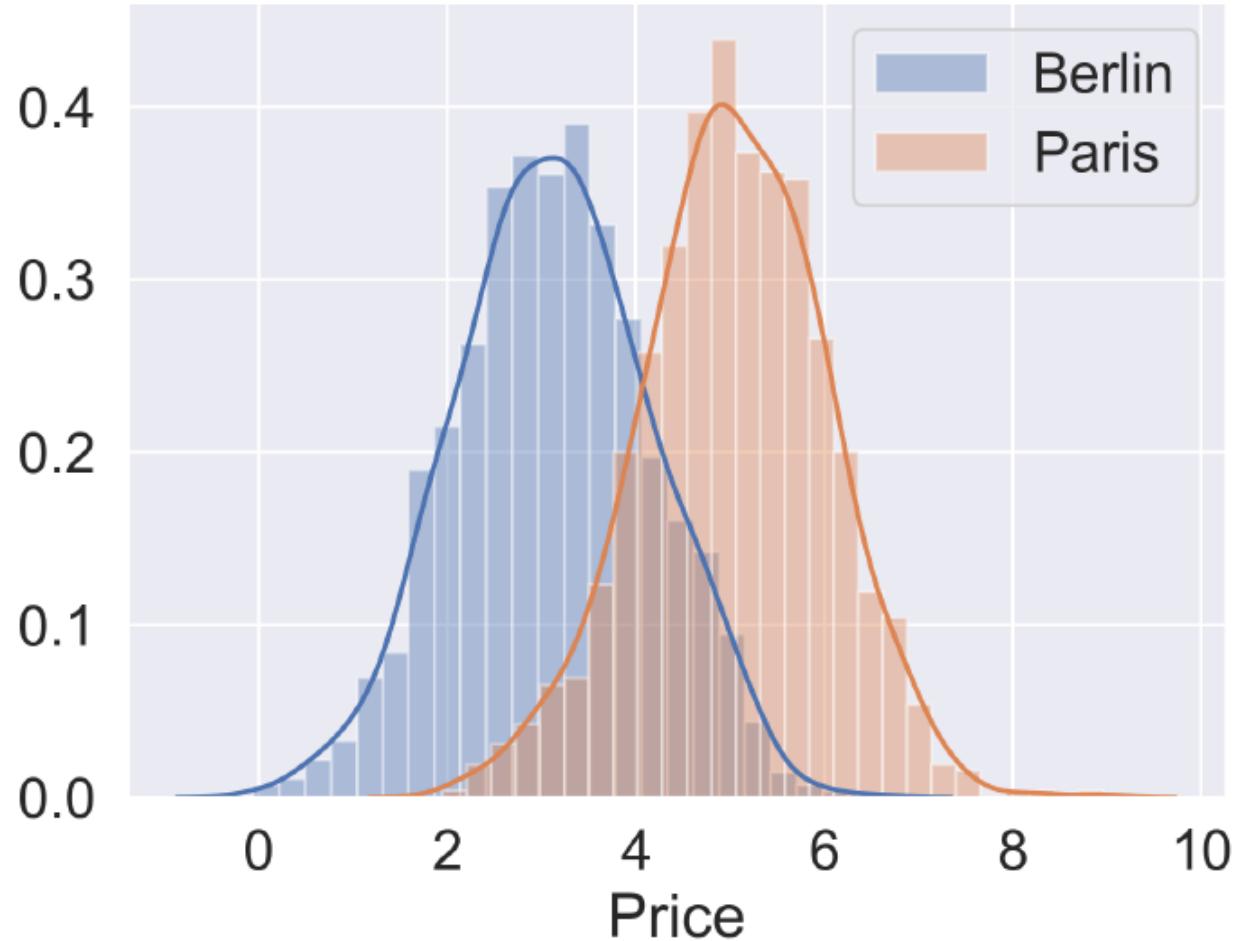
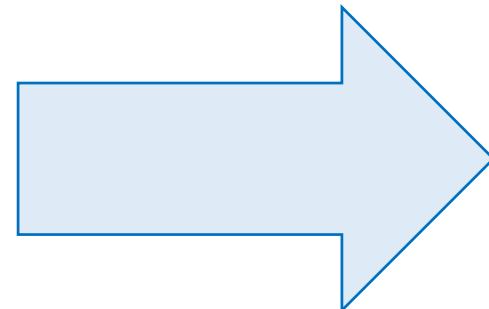
La maldición de la dimensionalidad (Ejemplo)

City	Price
Berlin	2
Paris	3



La maldición de la dimensionalidad (Ejemplo)

City	Price
Berlin	2.0
Berlin	3.1
Berlin	4.3
Paris	3.0
Paris	5.2
...	...



La maldición de la dimensionalidad

Construimos un Clasificador

Separate the feature we want to predict from the ones to train the model on.

```
y = house_df['City']  
  
X = house_df.drop('City', axis=1)
```

Perform a 70% train and 30% test data split

```
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)
```

La maldición de la dimensionalidad

Construimos un Clasificador

Create a Support Vector Machine Classifier and fit to training data

```
from sklearn.svm import SVC  
  
svc = SVC()  
  
svc.fit(X_train, y_train)
```

La maldición de la dimensionalidad

Evaluamos la calidad de la Predicción

```
from sklearn.metrics import accuracy_score  
  
print(accuracy_score(y_test, svc.predict(X_test)))
```

0.826

```
print(accuracy_score(y_train, svc.predict(X_train)))
```

0.832

La maldición de la dimensionalidad

Incrementamos dimensiones (Características)

City	Price	n_floors	n_bathroom	surface_m2
Berlin	2.0	1	1	190
Berlin	3.1	2	1	187
Berlin	4.3	2	2	240
Paris	3.0	2	1	170
Paris	5.2	2	2	290
...



La maldición de la dimensionalidad

Se debe asegurar que existe una cantidad significativa de ejemplos por cada dimensión adicional que se maneja en nuestro problema.

De otra manera puede haber **sobreajuste (Overfitting)**. No se garantiza la generalización.

Por esto es importante identificar de que manera podemos eliminar características (**feature selection**) que poco aporten en nuestra búsqueda de nuestro modelo de reconocimiento de patrones.

Características (Dimensiones) con valores faltantes o Poca Varianza

En algunos casos debemos identificar estrategias para completar datos faltantes. (Usar el valor medio o por ejemplo (**mean**) - El Valor más frecuente (**mode**))

Para los valores que presenten poca varianza podemos eliminarlos. (Para esto es importante **normalizar la varianza** en todo el dataset)

También es muy útil identificar el valor de **correlación** en los datos.

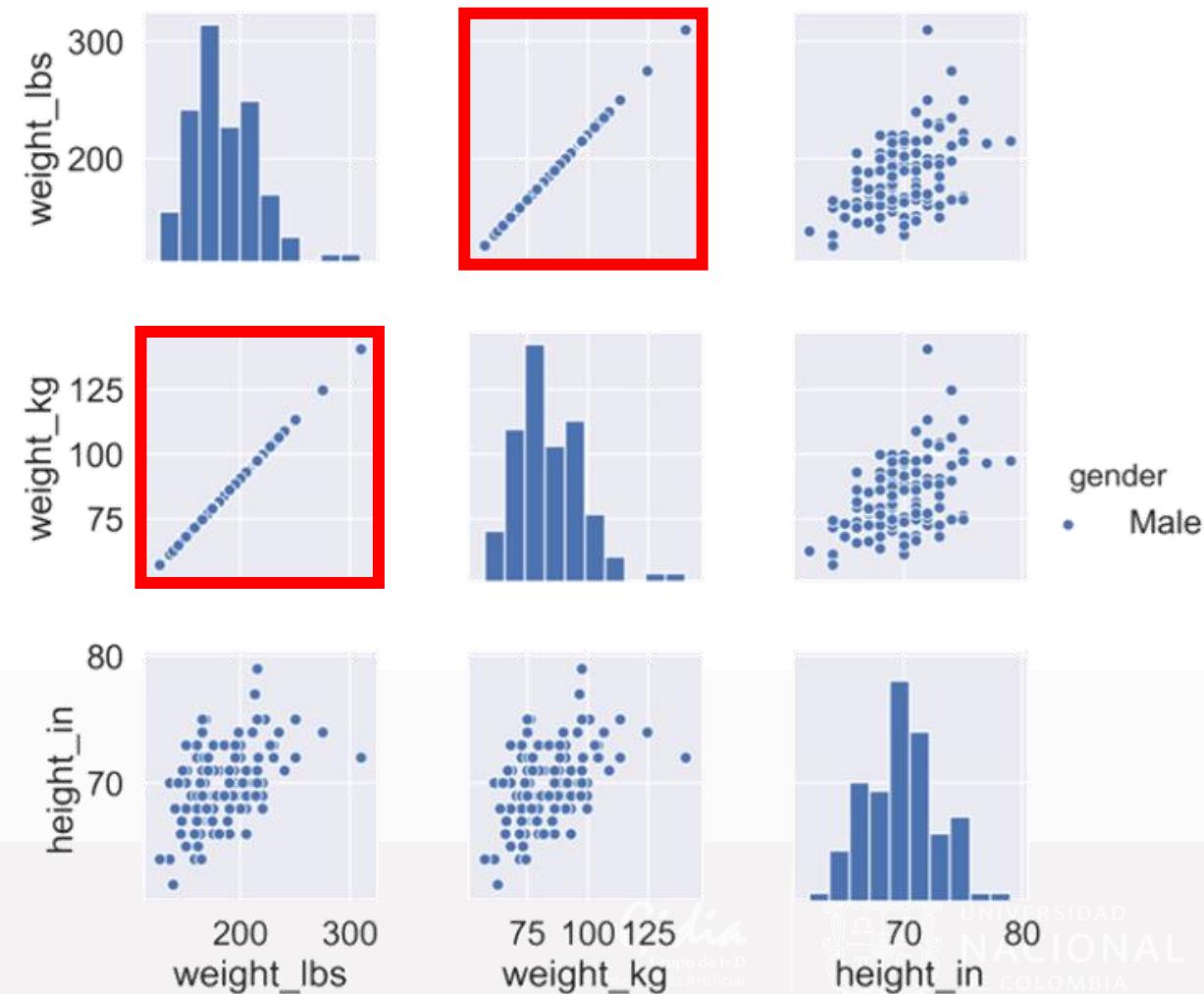
Correlación de Valores

Correlación en parejas

```
sns.pairplot(ansur, hue="gender")
```

$$Cov(X, Y) = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{n}$$

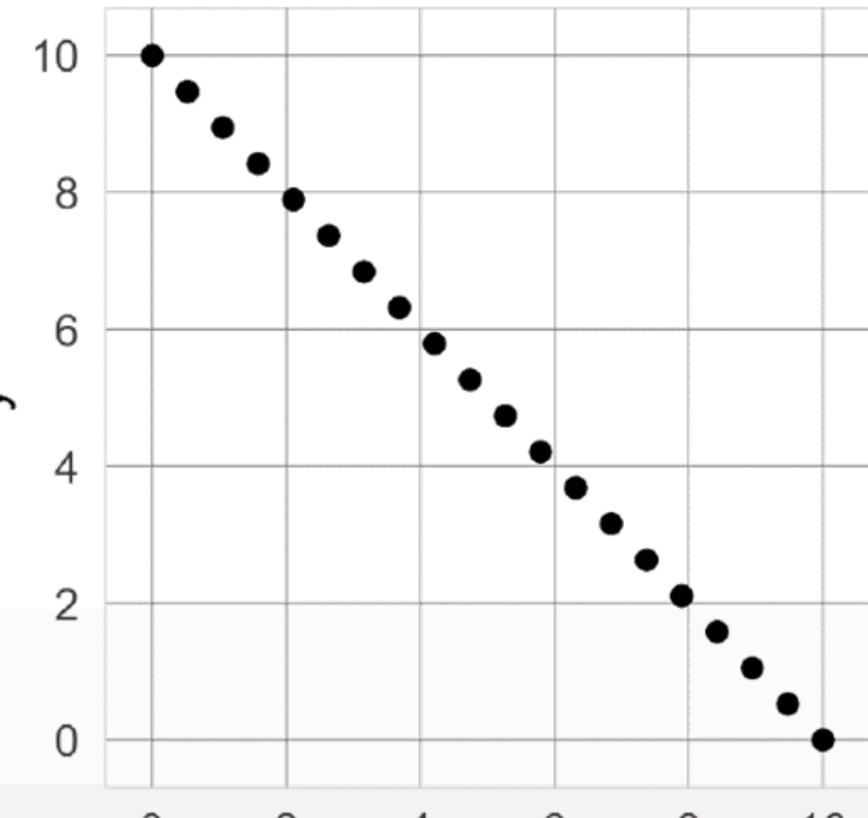
$$\rho_{xy} = \frac{Cov_{xy}}{\sigma_x \sigma_y}$$



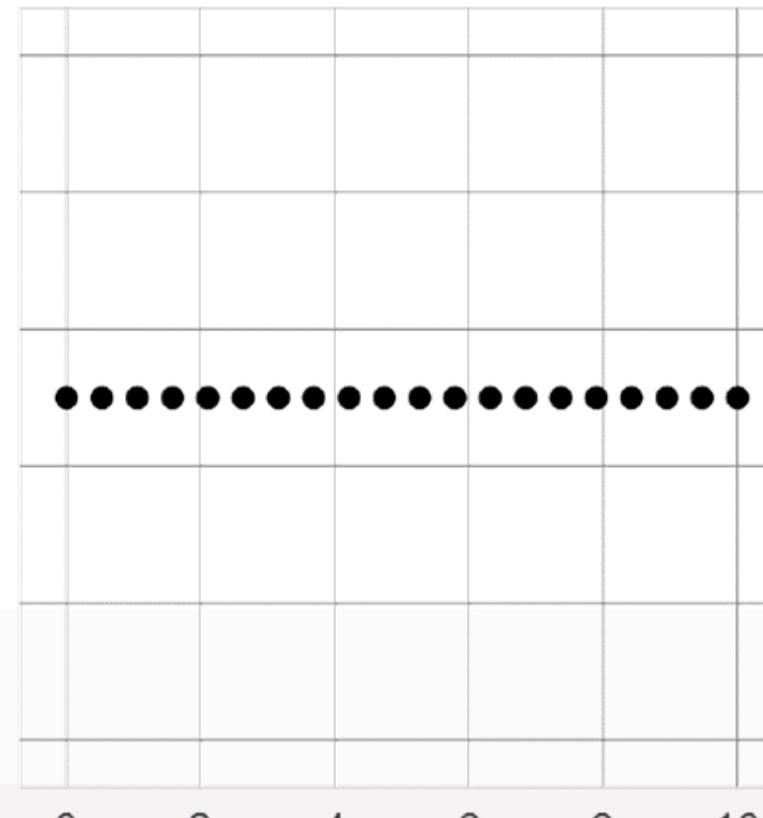
Correlación de Valores

Coeficiente de correlación

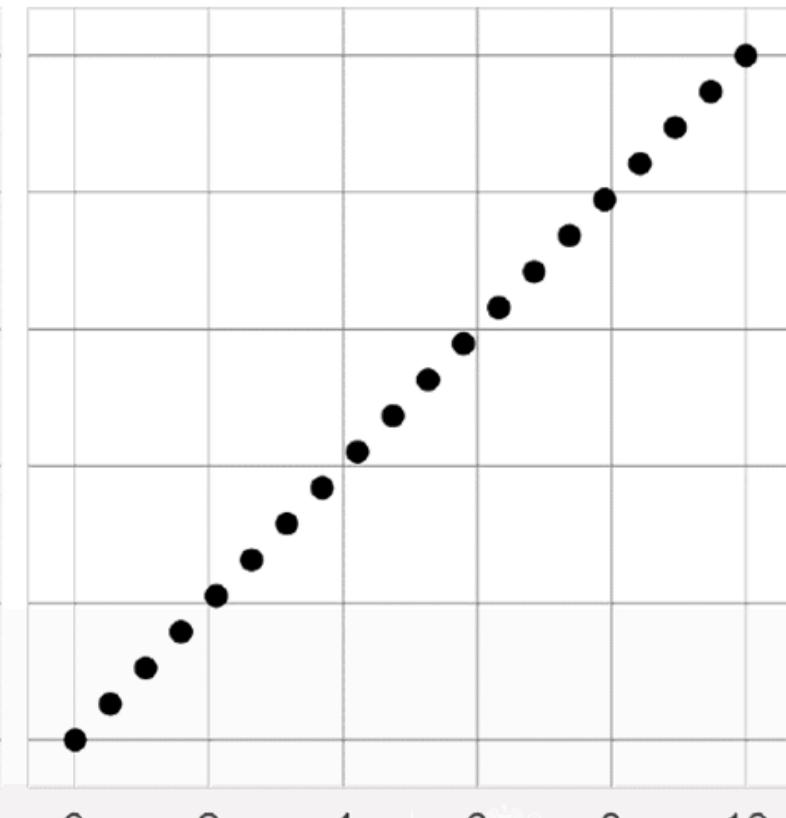
$$r = -1$$



$$r = 0$$



$$r = 1$$



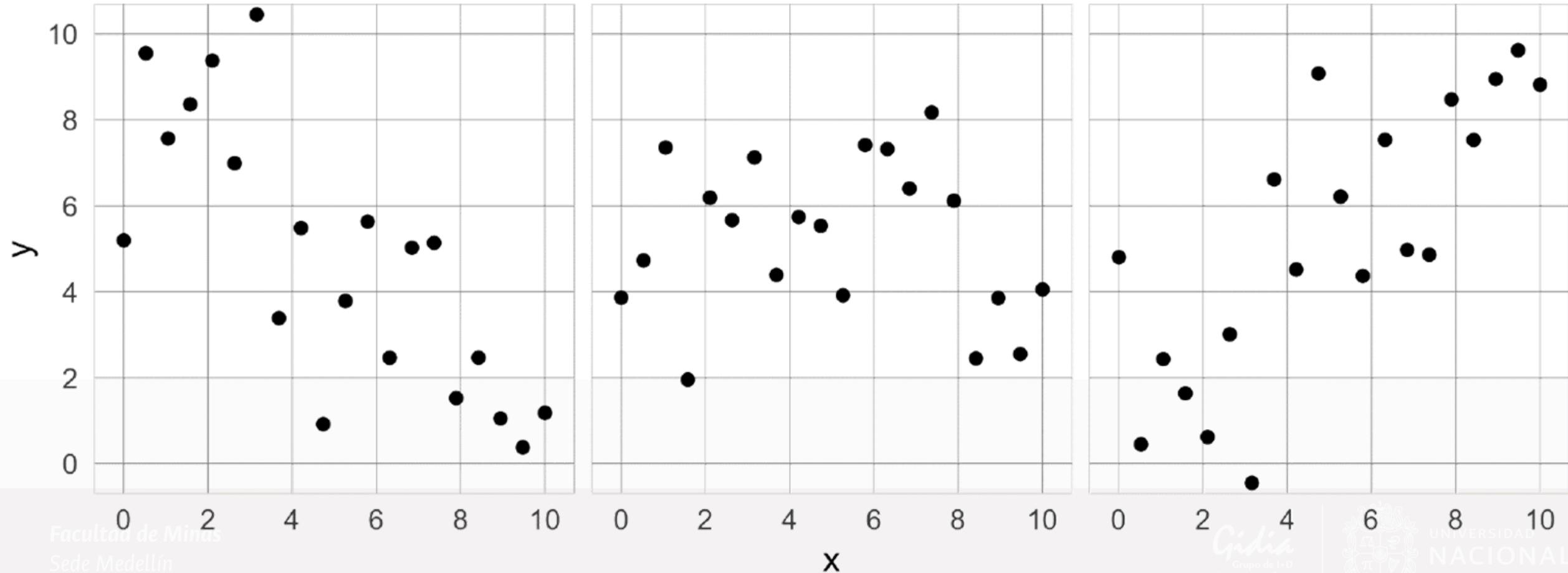
Correlación de Valores

Coeficiente de correlación

$$r = -0.88$$

$$r = 0.05$$

$$r = 0.88$$



Correlación de Valores

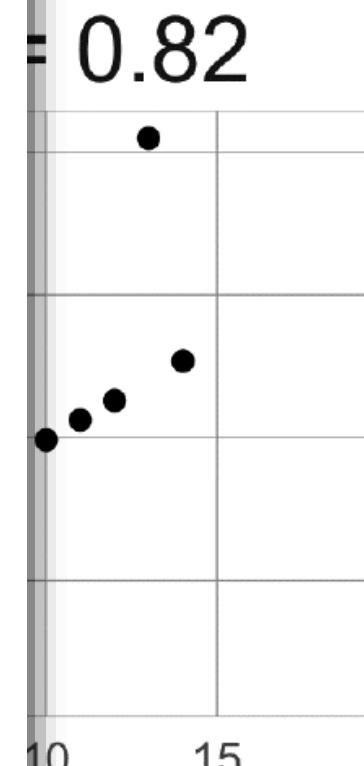
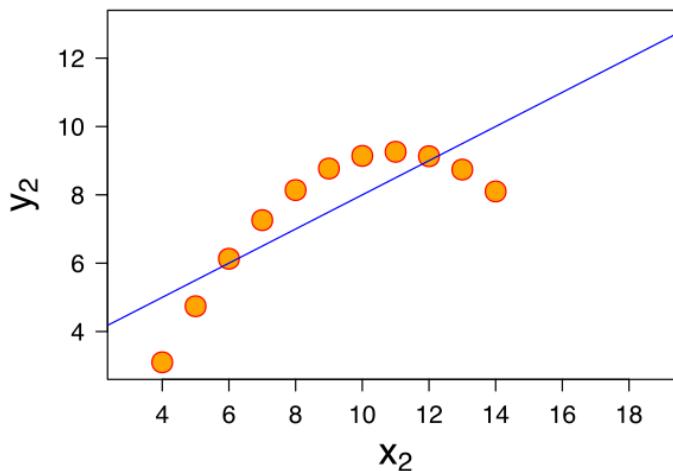
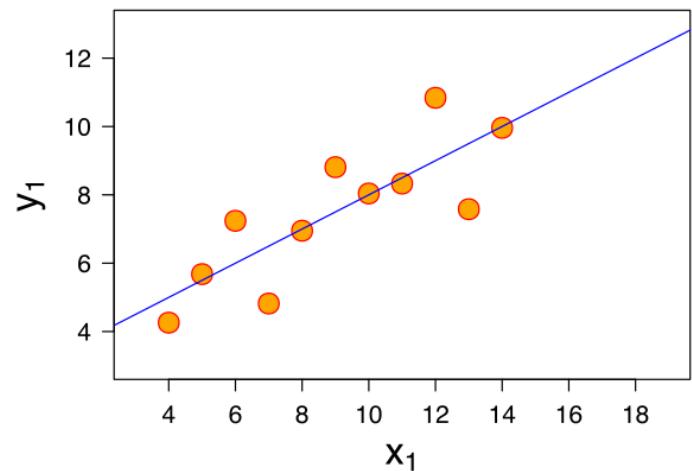
Matriz de Correlación

```
weights_df.corr()
```

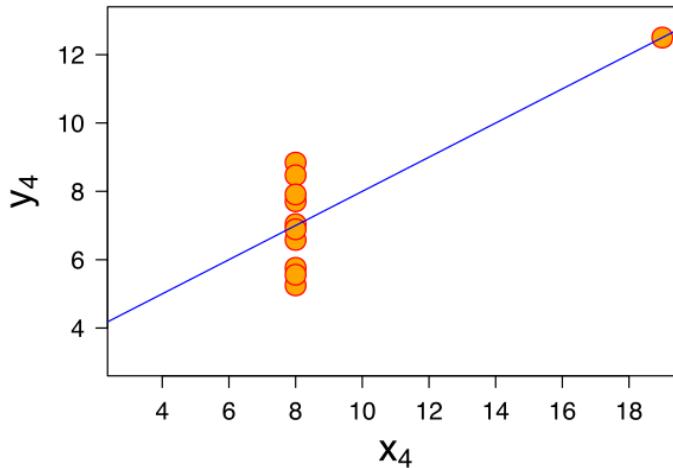
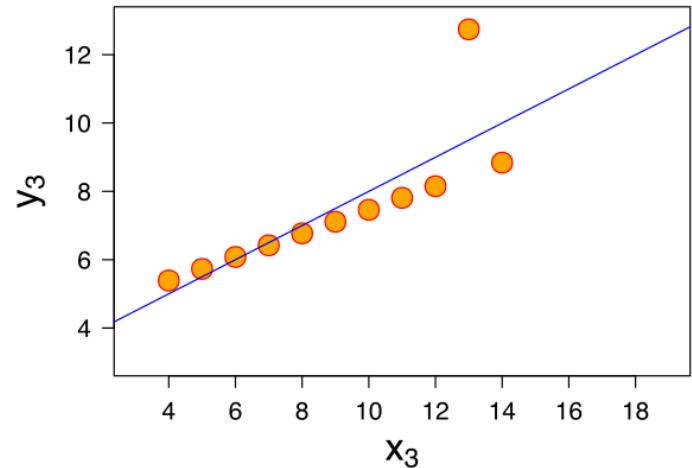
	weight_lbs	weight_kg	height_in
weight_lbs	1.00	1.00	0.47
weight_kg	1.00	1.00	0.47
height_in	0.47	0.47	1.00

Correlación de Valores

Problemas con la Correlación



$r = 0.82$

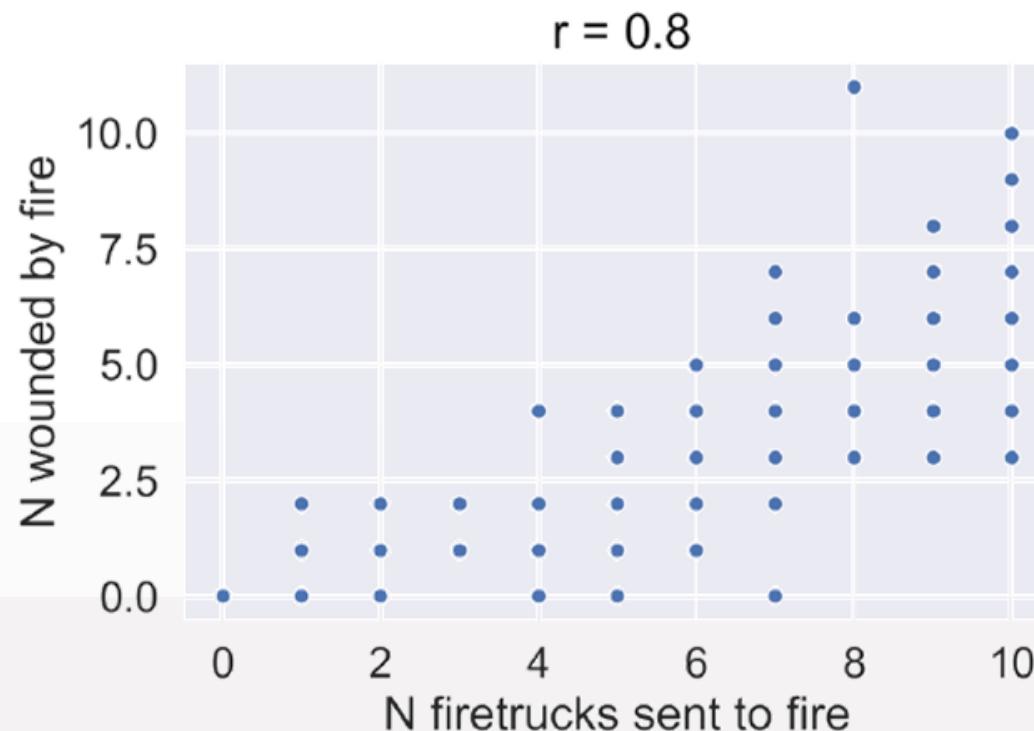


Cuarteto de Anscombe

Correlación de Valores

Problemas con la Correlación

```
sns.scatterplot(x="N firetrucks sent to fire",  
y="N wounded by fire", data=fire_df)
```



Causación

Laboratorios Restantes

<https://github.com/srobles05/CRP-2019S2/>

Reemplazo de valores faltantes y Filtrado de Baja Varianza

Considere una variable en nuestro conjunto de datos donde todas las observaciones tienen el mismo valor, digamos 1. Si usamos esta variable, ¿cree que puede mejorar el modelo que construiremos? La respuesta es no, porque esta variable tendrá cero varianza.

Entonces, necesitamos calcular la varianza de cada variable que se nos da. Luego, eliminar las variables que tienen una varianza baja en comparación con otras variables en nuestro conjunto de datos. La razón para hacerlo, como mencionamos anteriormente, es que las variables con una varianza baja no afectarán la variable objetivo.

```
# import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
train=pd.read_csv("Train_data.csv")
# To show rows and columns
train.shape
```

(8523, 12)

Primero, identificamos los datos faltantes en cada dimensión

```
train.isna().sum()
```

Item_Identifier	0
Item_Weight	1463
Item_Fat_Content	0
Item_Visibility	0
Item_Type	0
Item_MRP	0
Outlet_Identifier	0
Outlet_Establishment_Year	0
Outlet_Size	2410
Outlet_Location_Type	0
Outlet_Type	0
Item_Outlet_Sales	0
dtype:	int64

Basic_labs



Reemplazaremos los valores faltantes en la columna Item_Weight usando el valor medio de las observaciones conocidas de Item_Weight. Para la columna Outlet_Size, usaremos el modo de los valores Outlet_Size conocidos para reemplazar los valores faltantes:

```
train['Outlet_Size'].mode()[0]
'Medium'
```

Gracias !!!



© Man Bouncing Question Mark Towards Doctor - Artist: [Art Glazer](#)