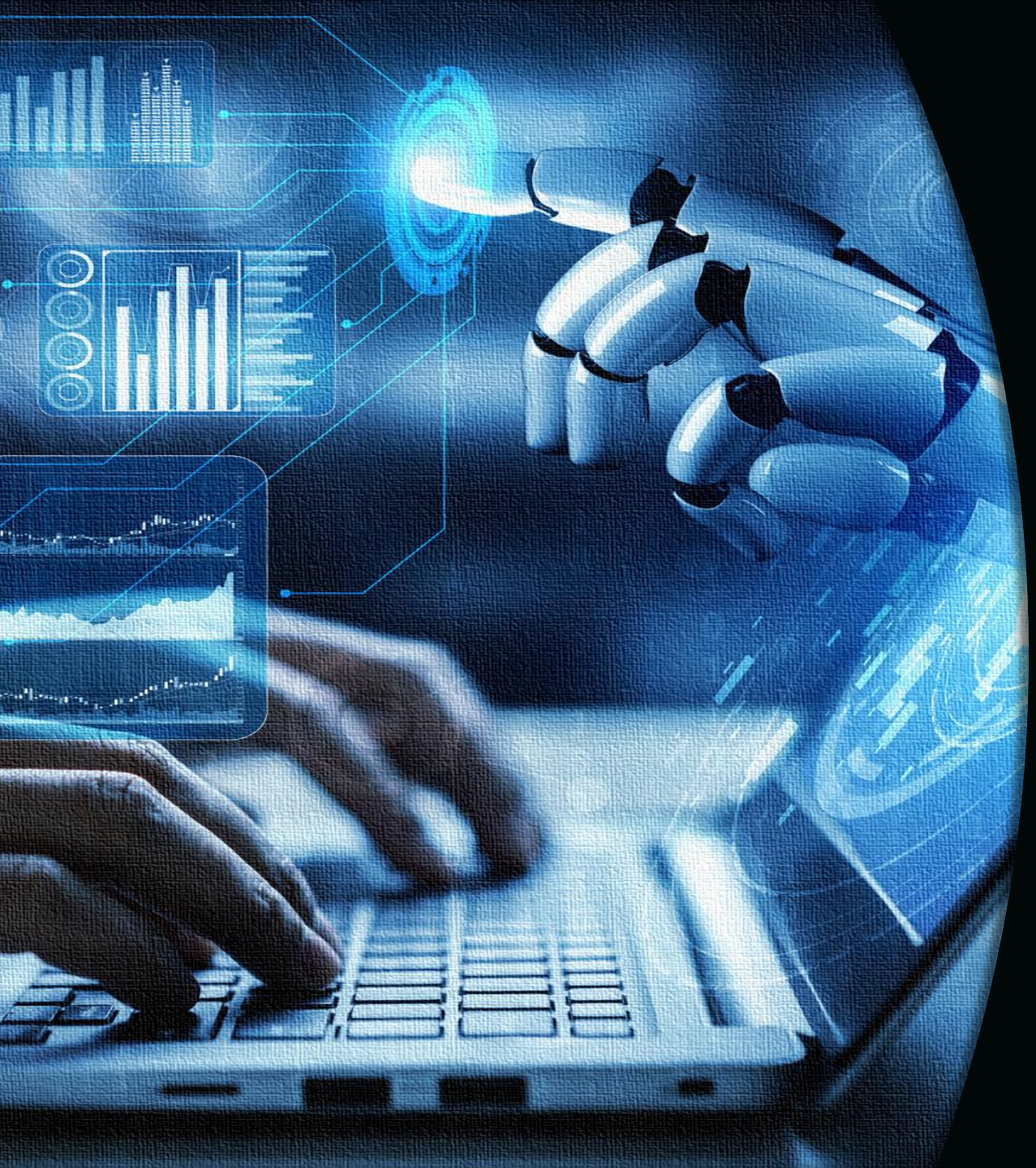


UTEC
Universidad Tecnológica

Hackathon Interfaces Cerebro Computadora
2022/2023
Taller N° 2

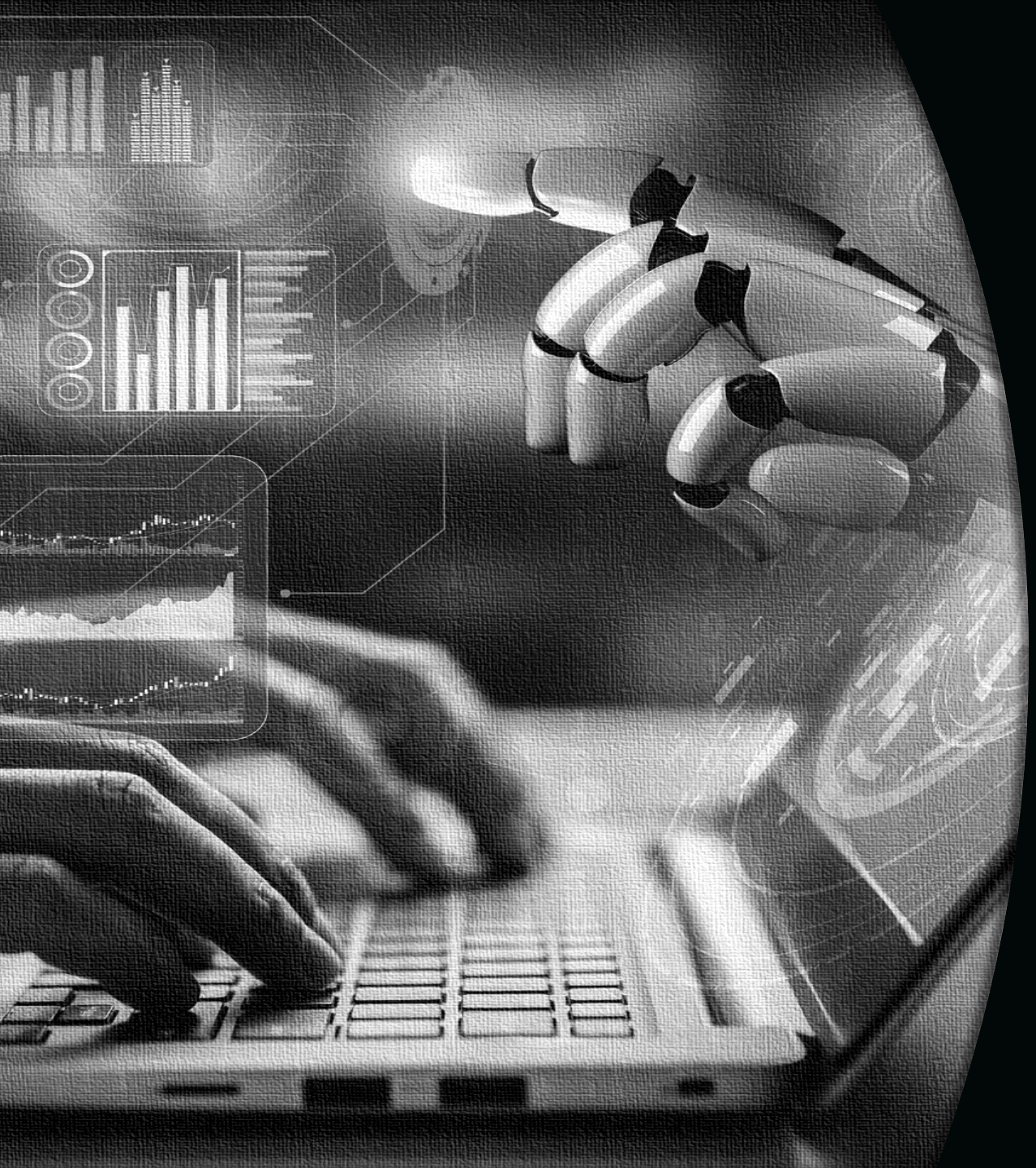
Introducción al Machine Learning

MSc. Bioing. BALDEZZARI Lucas

Profesor Adjunto
Ingeniería Biomédica

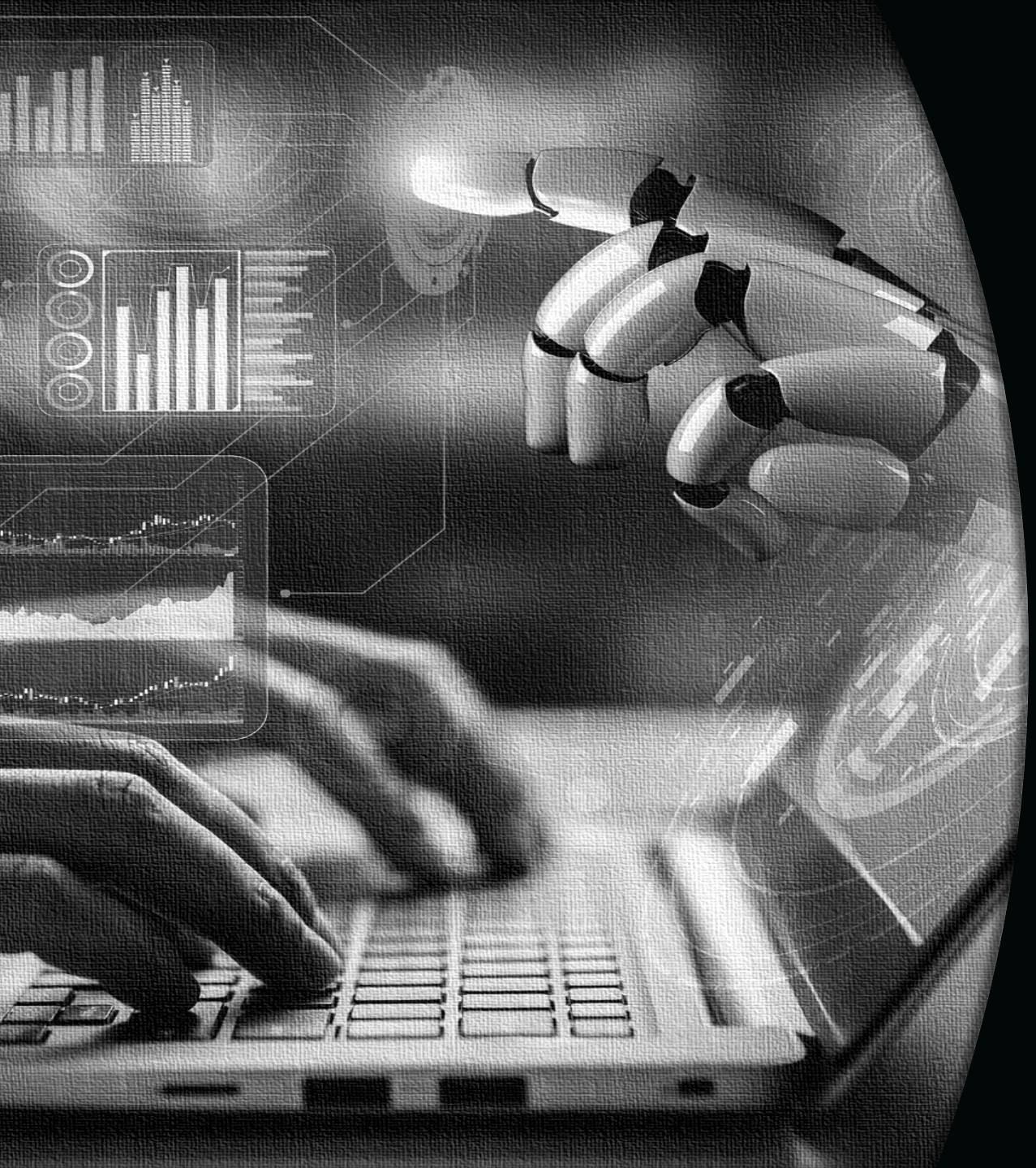
Fray Bentos, Uruguay

5 de octubre de 2022



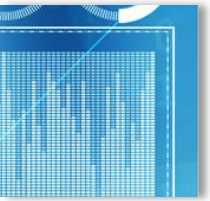
Temas a tratar

- Intro: ¿Qué es el Machine Learning? objetivos, usos.
- Tipos de ML: Aprendizaje Supervisado y Aprendizaje No Supervisado. Ejemplos.
- Introducción al uso de Scikit-Learn.
- Resolución de un problema real.



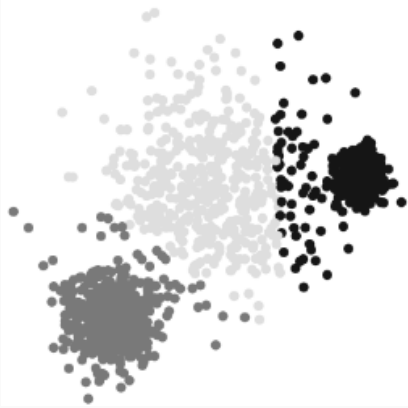
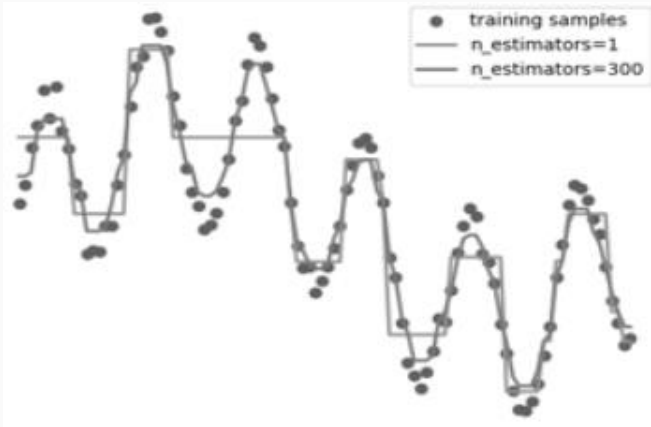
Temas a tratar

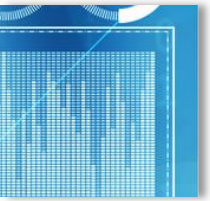
- Intro: ¿Qué es el Machine Learning? objetivos, usos.
- Tipos de ML: Aprendizaje Supervisado y Aprendizaje No Supervisado. Ejemplos.
- Introducción al uso de Scikit-Learn.
- Resolución de un problema real.



¿Qué entienden por Machine Learning?

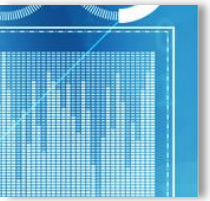





¿Cómo definimos al Machine Learning.

Hagamos algunas preguntas…

- ¿Qué significa que una máquina *aprenda* sobre algo?
- Si descargamos una enciclopedia, ¿la computadora aprende algo? ¿Se hace repentinamente inteligente?
- ¿Qué se necesita para que una máquina aprenda?
- ¿Dónde empieza y donde termina el Machine Learning?



¿Cómo definimos al Machine Learning?

El Machine Learning es la *ciencia* que se encarga de crear programas de computadora para que *aprendan de los datos*, utilizando *modelos matemáticos y estadísticos*.

¿Qué *aprende* exactamente la máquina/computadora?

La palabra Learning hace referencia a que *tuneamos* o modificamos ciertos *parámetros* del modelo, los cuales se *adaptan* a los datos observados.

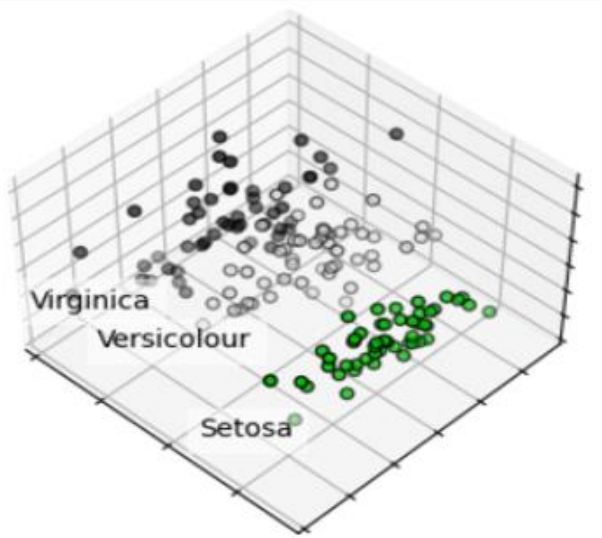
Podemos decir que el programa *aprende* de los datos.

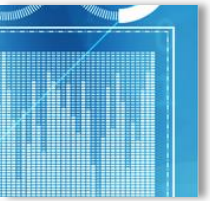
¿Cómo definimos al Machine Learning?

Otra definición más ingenieril puede ser,

"…A computer program *is said to learn* from *experience* E with respect to some *task* T and some *performance measure* P, if its performance on T, as measured by P, improves with experience E (Tom Mitchell, 1997)…"

¿Algún ejemplo?





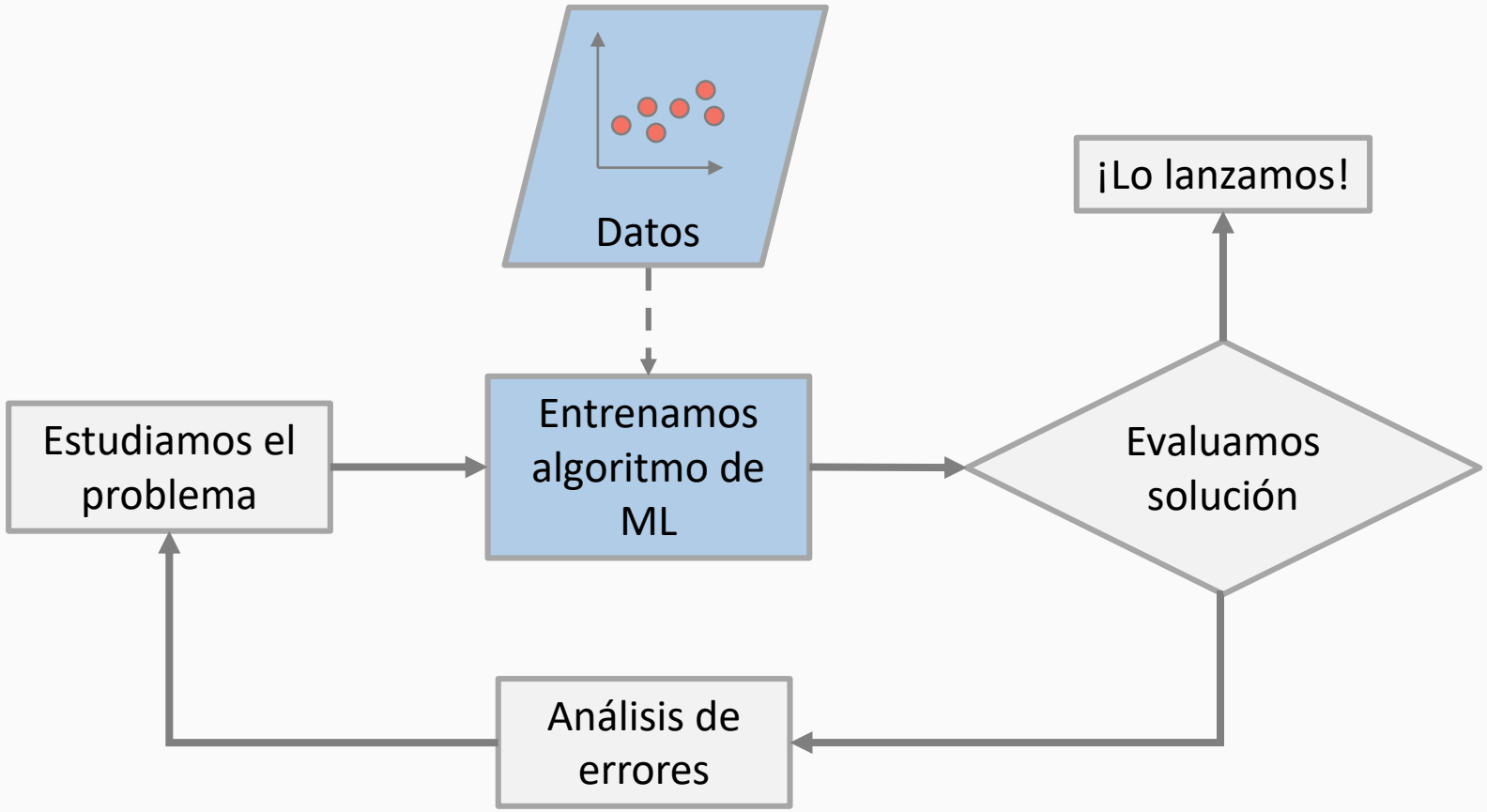
¿En qué podemos utilizar ML?

- Detección de latidos cardiacos sanos vs patológicos (clustering).
- Detección de spam (SVM).
- Analizar imágenes de productos en una línea de producción para clasificarlos automáticamente (CNN).
- Detección de tumores cerebrales en tomografías (CNN).
- Clasificación de nuevos artículos (NLP).
- Proyectar las ganancias de una empresa para el año próximo, basado en la performance de muchas métricas (Regresión).
- Detección de fraudes en tarjetas de créditos (detección de anomalías).
- Segmentación de clientes en función de sus compras para poder diseñar una estrategia de marketing diferente para cada segmento (Clustering).
- Representar complejos set de datos -dimensionalmente altos- en un diagrama claro e intuitivo.
- Recomendación de cierto producto a un cliente, en base a compras pasadas (sistemas de recomendación).

…Y muchísimo más…

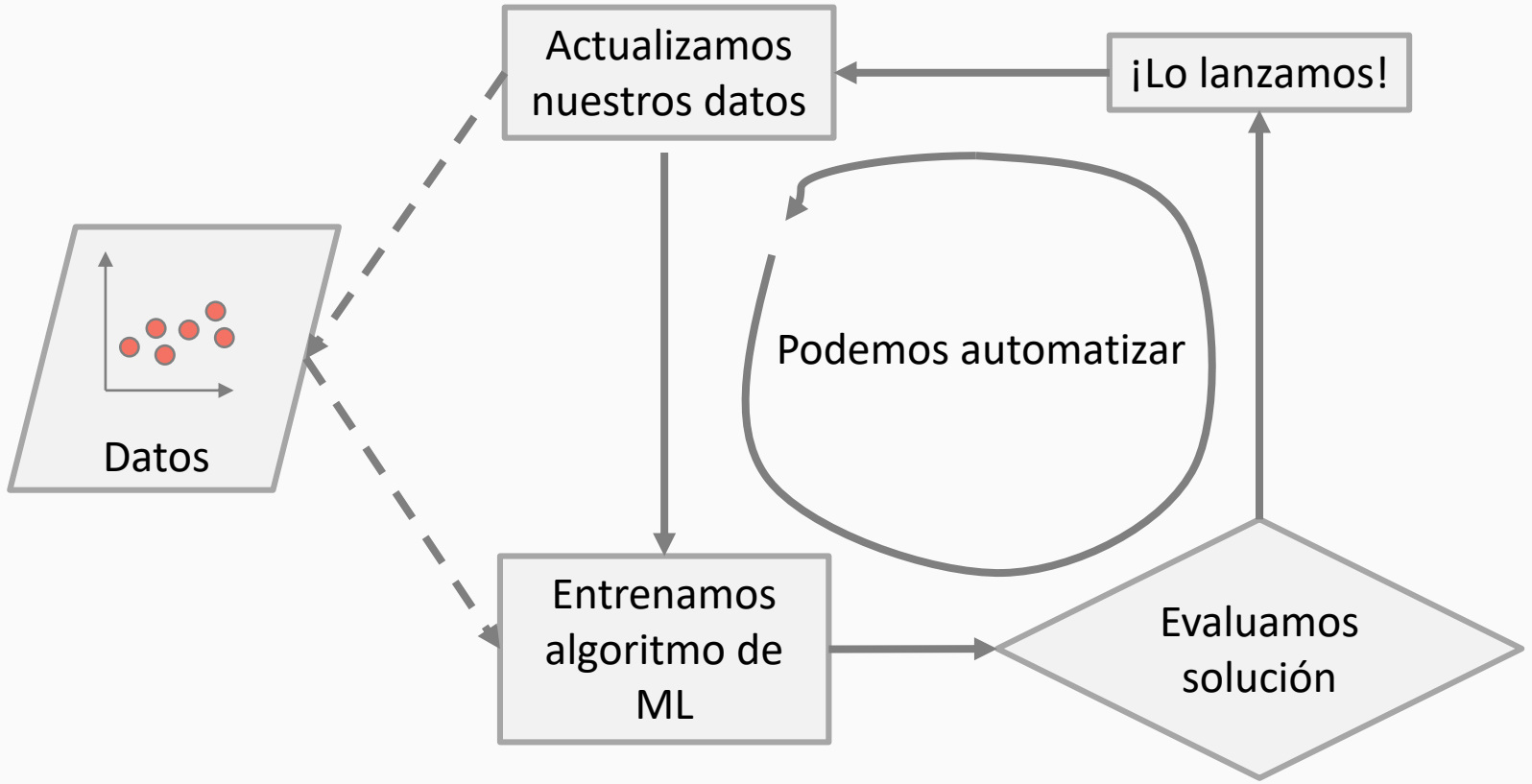
Enfoque del ML

La metodología básica para trabajar en ML es el siguiente.



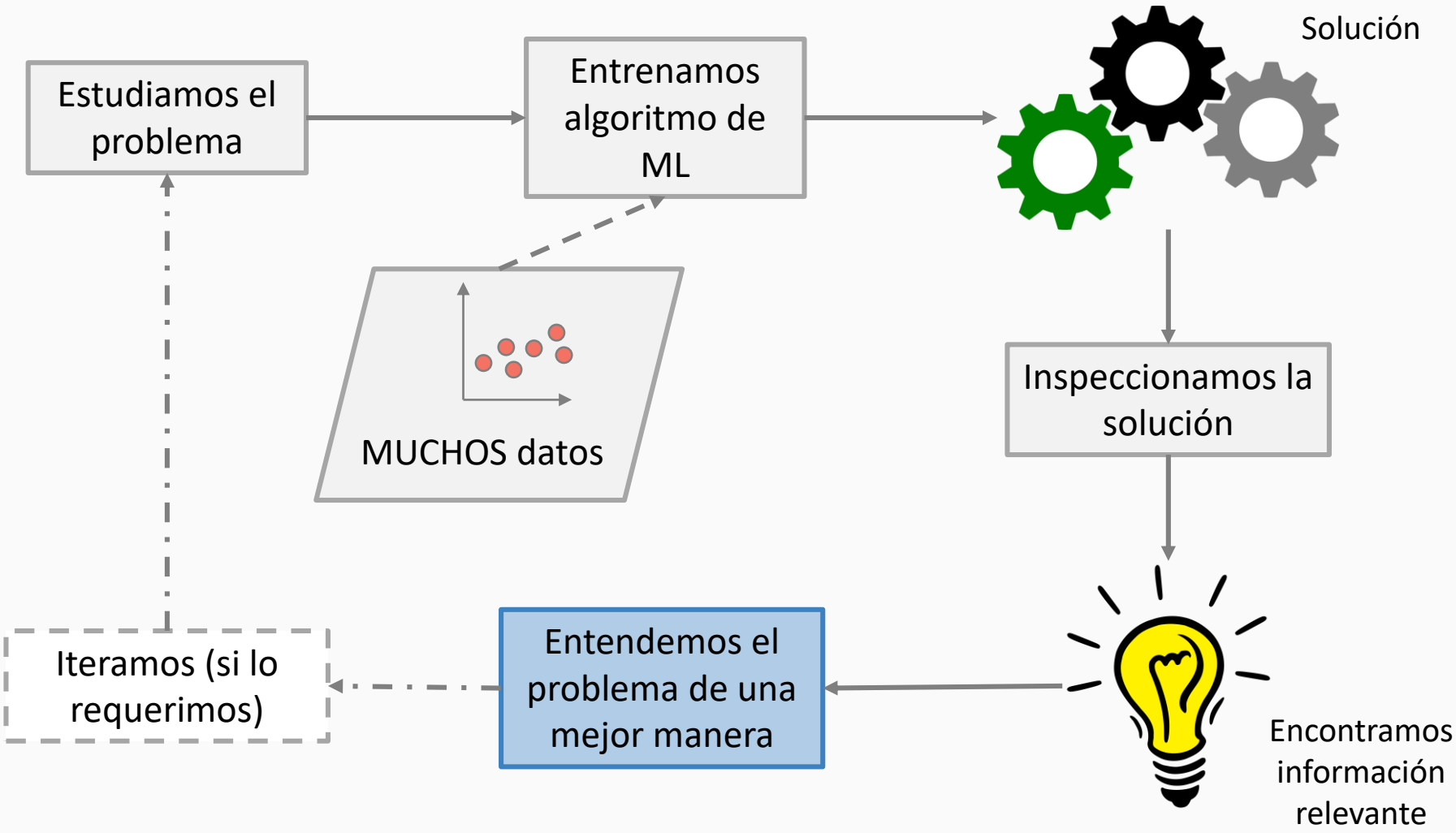
Enfoque del ML

Aunque podemos automatizar ciertos procesos.



Enfoque del ML

Podemos usar ML para aprender, esto se llama *Data Mining*.



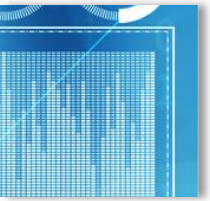
Temas a tratar

- Intro: ¿Qué es el Machine Learning? objetivos, usos.
- Tipos de ML: Aprendizaje Supervisado y Aprendizaje No Supervisado. Ejemplos.
- Introducción al uso de Scikit-Learn.
- Resolución de un problema real.

Tipos de Machine Learning

De manera general podemos dividir tres tipos de ML.

Supervised Learning	• Datos con labels • Feedback directo • *Predicción*
Unsupervised Learning	• Datos *sin* labels • Sin feedback o feedback indirecto • Trata de encontrar *estructuras* escondidas en los datos.
Reinforcemente Learning	• Procesos de decisión • Sistemas de recompensa. • Aprende una serie de acciones.



Supervised Learning y Unsupervised Learning

Supervised Learning: Trata acerca de modelar la relación entre las características medibles dentro de mi set de datos y ciertas *labels* o *etiquetas* asociadas al set.

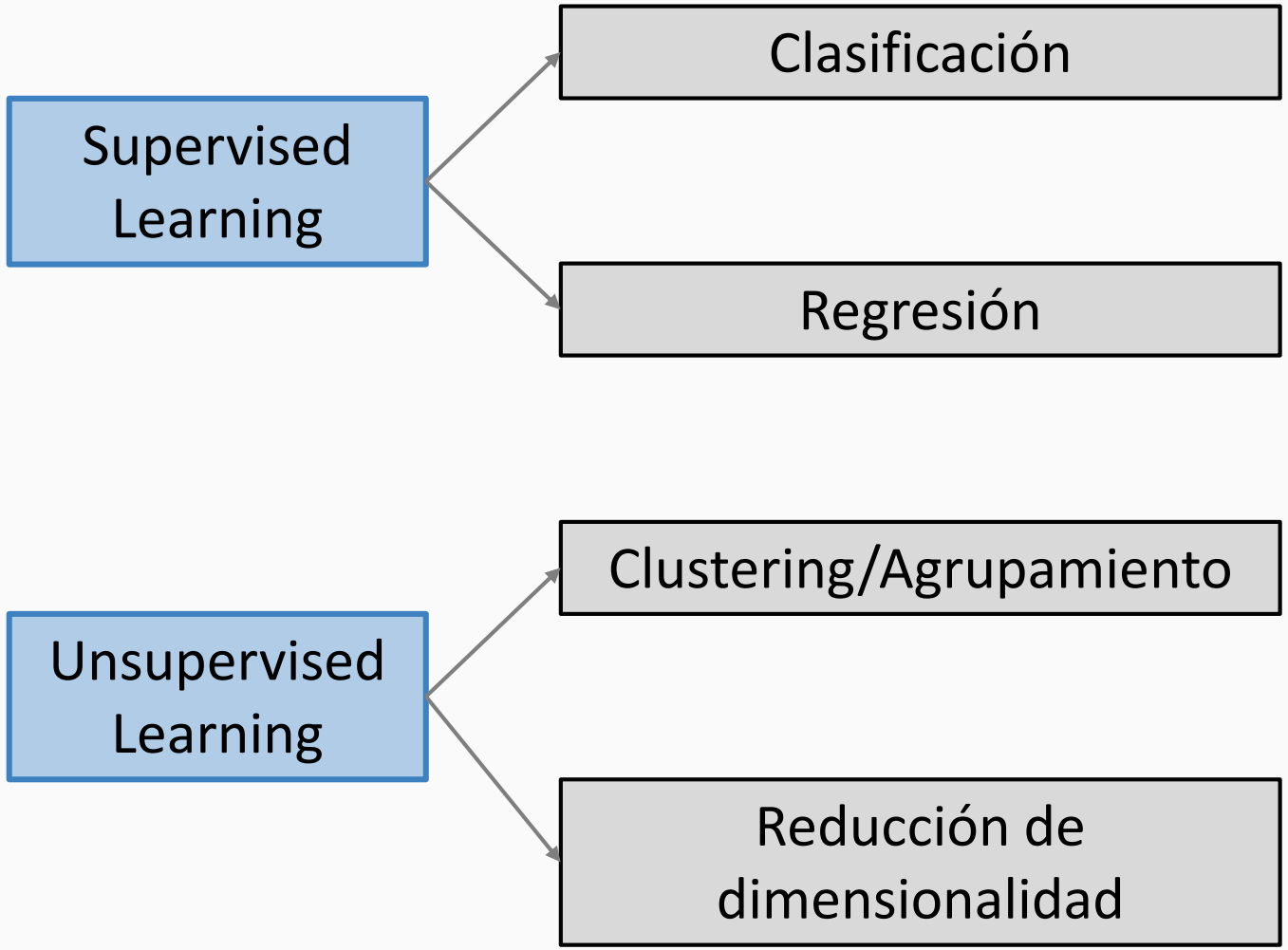
Una vez entrenado el modelo, puede ser usado para aplicar *labels* a datos nuevos y desconocidos.

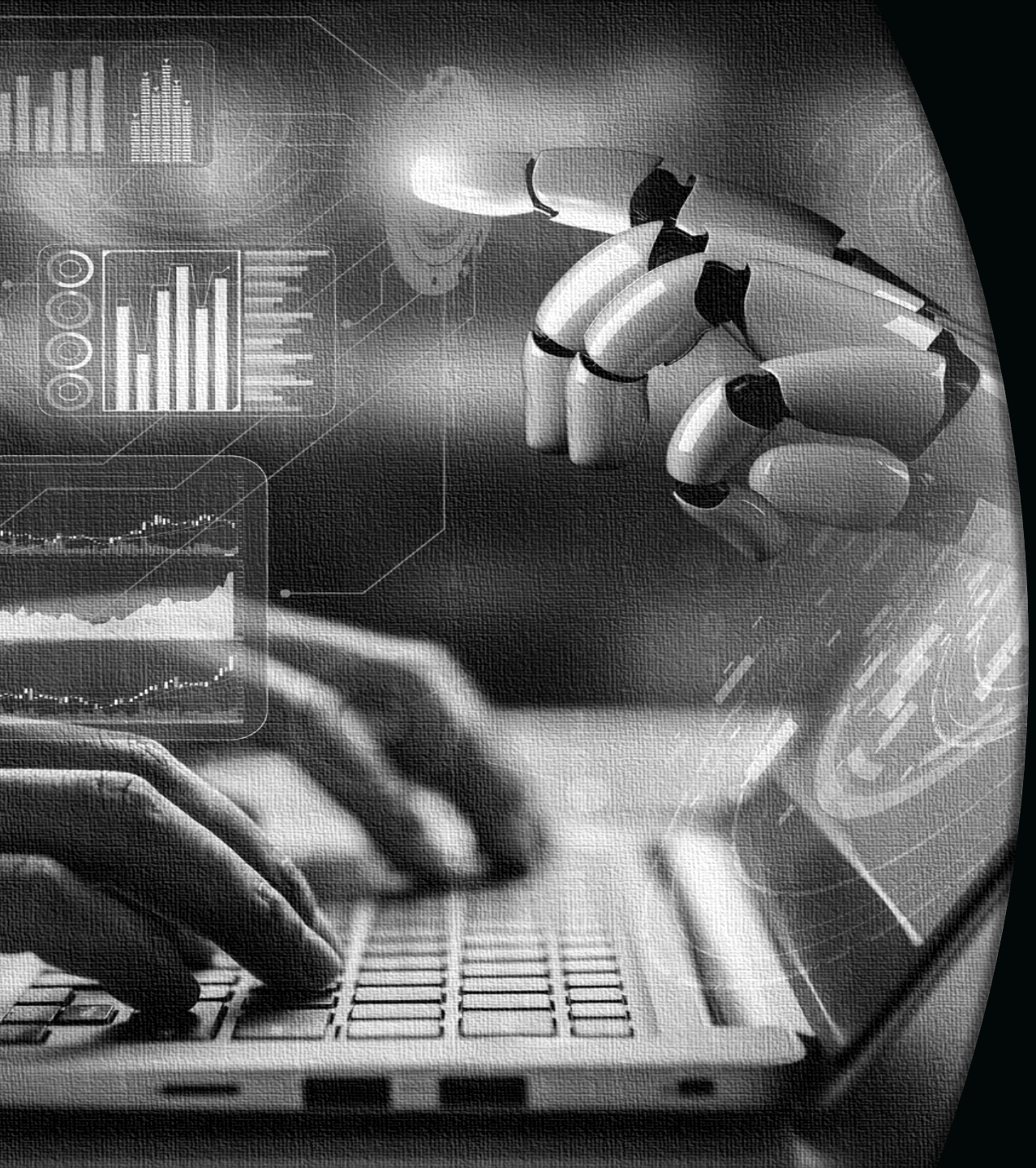
Unsupervised Learning: Involucra la modelización entre las características del set de datos sin ninguna referencia o etiquetas, se dice, *"dejen que el set de datos hable por sí sólo".*

Sirve, entre otras cosas, para reconocer patrones dentro del set de datos.

Supervised Learning y Unsupervised Learning

Dentro de estos tipos de ML pueden ser subdivididos.

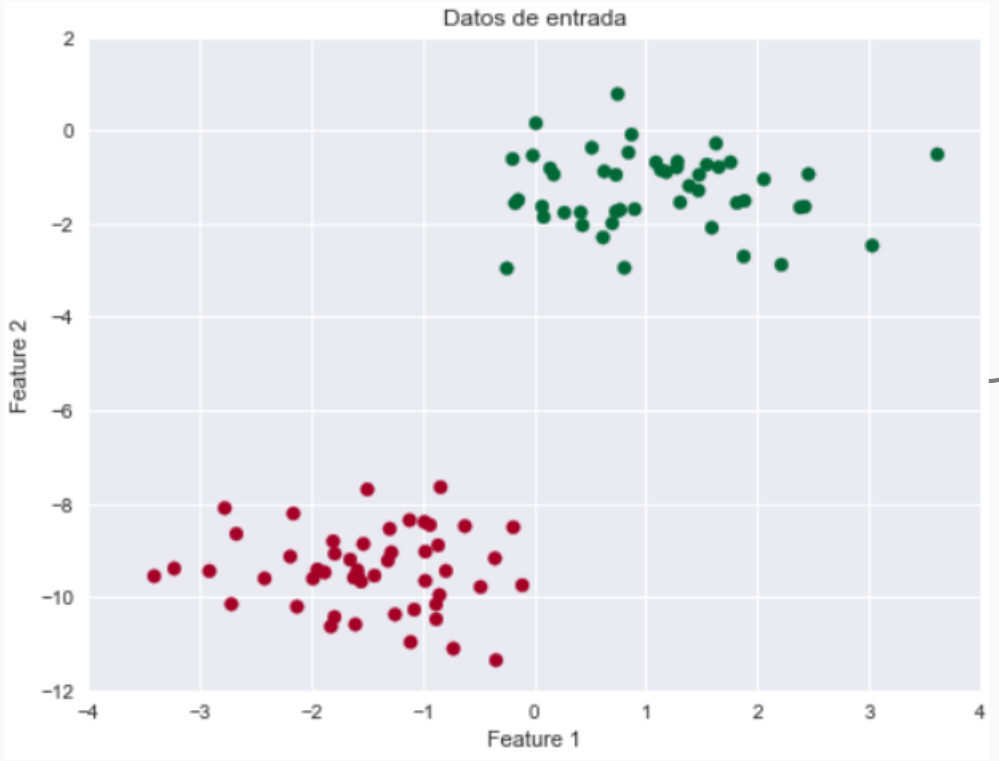




Ejemplo de Clasificación

Clasificación: Prediciendo labels discretas

Clasificando con Support Vector Machine.

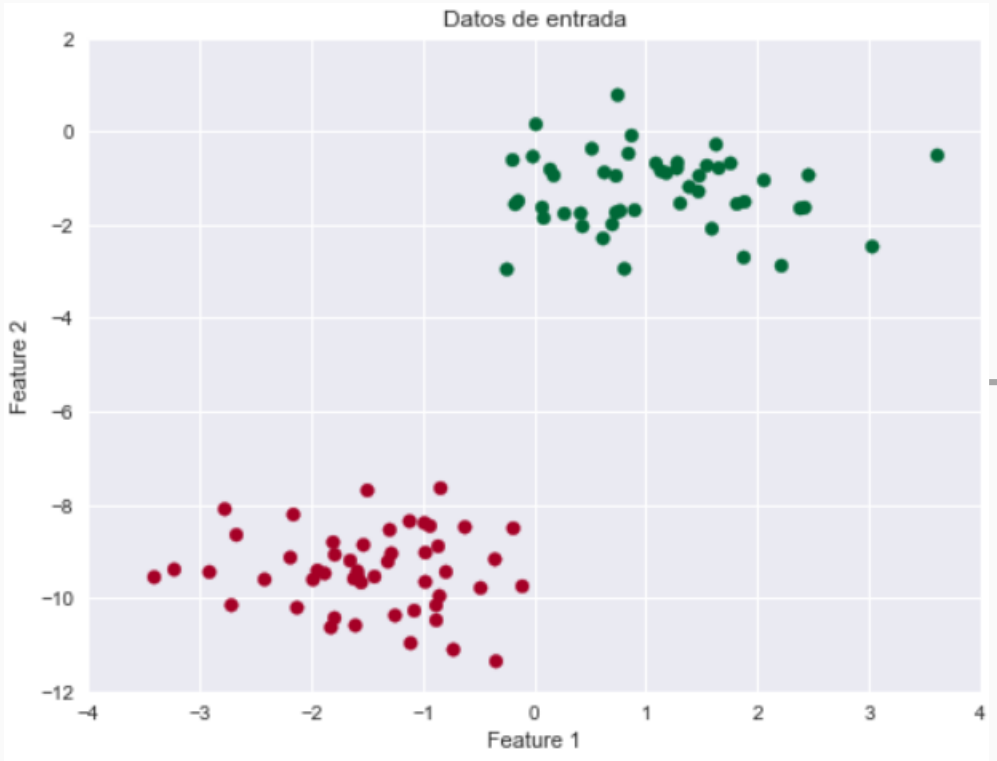


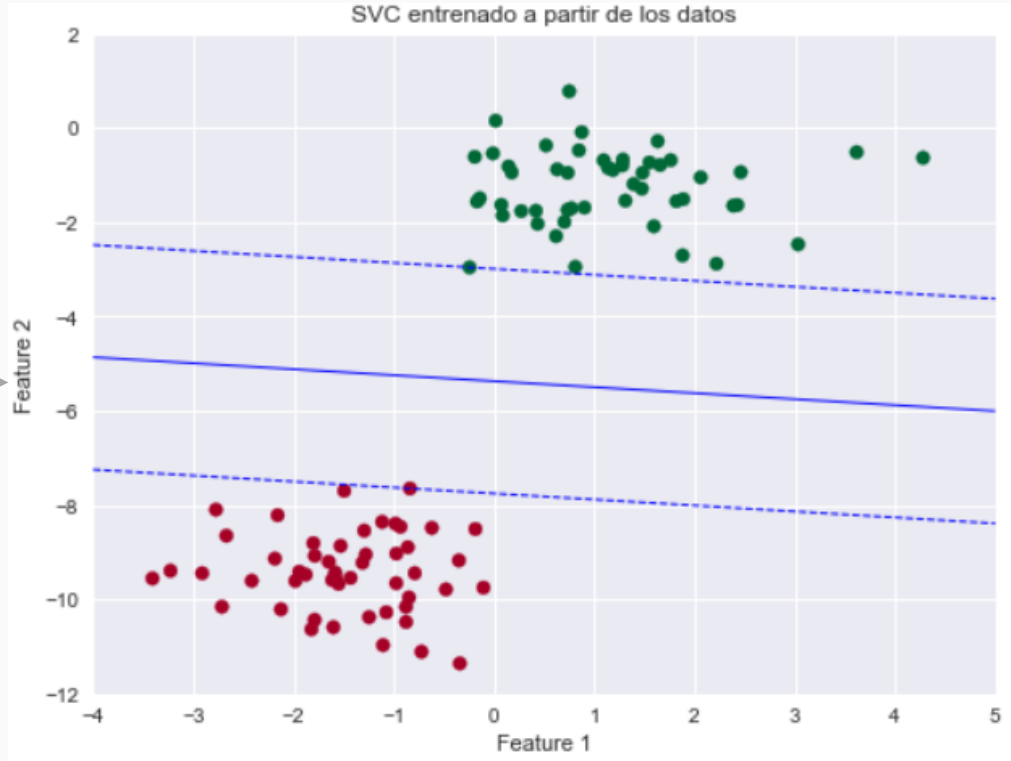

```
1  print(X[:10])
2  print()
3  print("Etiquetas")
4  print(y[:10])
```

```
[[ -0.10595955  -9.75001723]
 [ -3.40766152  -9.55463746]
 [ -0.83893872  -7.64770895]
 [  0.41784648  -1.76028382]
 [  1.28055622  -0.79577582]
 [ -0.98374633  -8.39376827]
 [ -0.87779682 -10.47770583]
 [ -0.19272282  -0.61650263]
 [ -1.80270217  -8.80751034]
 [ -0.18631362  -8.50716686]]

Etiquetas
[0 0 0 1 1 0 0 1 0 0]
```

Clasificación: Prediciendo labels discretas

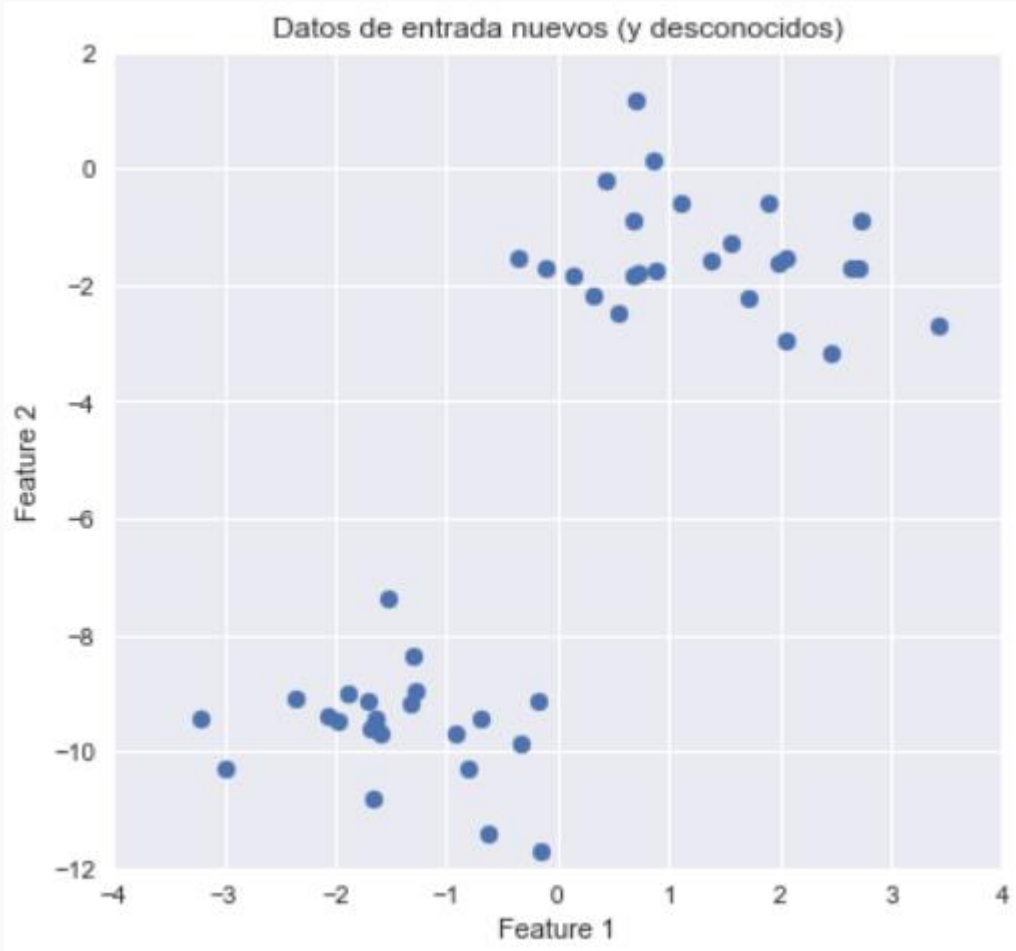
Modelo entrenado a partir de nuestro set de datos.

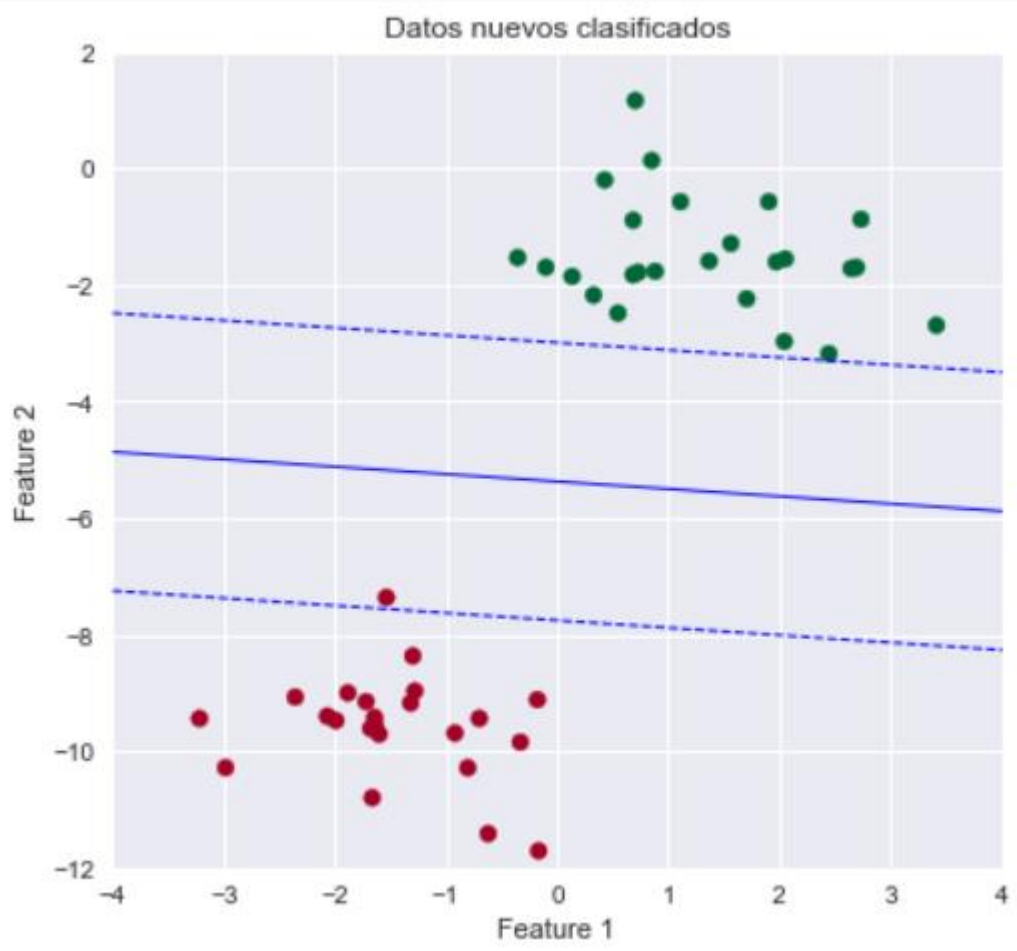




Clasificación: Prediciendo labels discretas

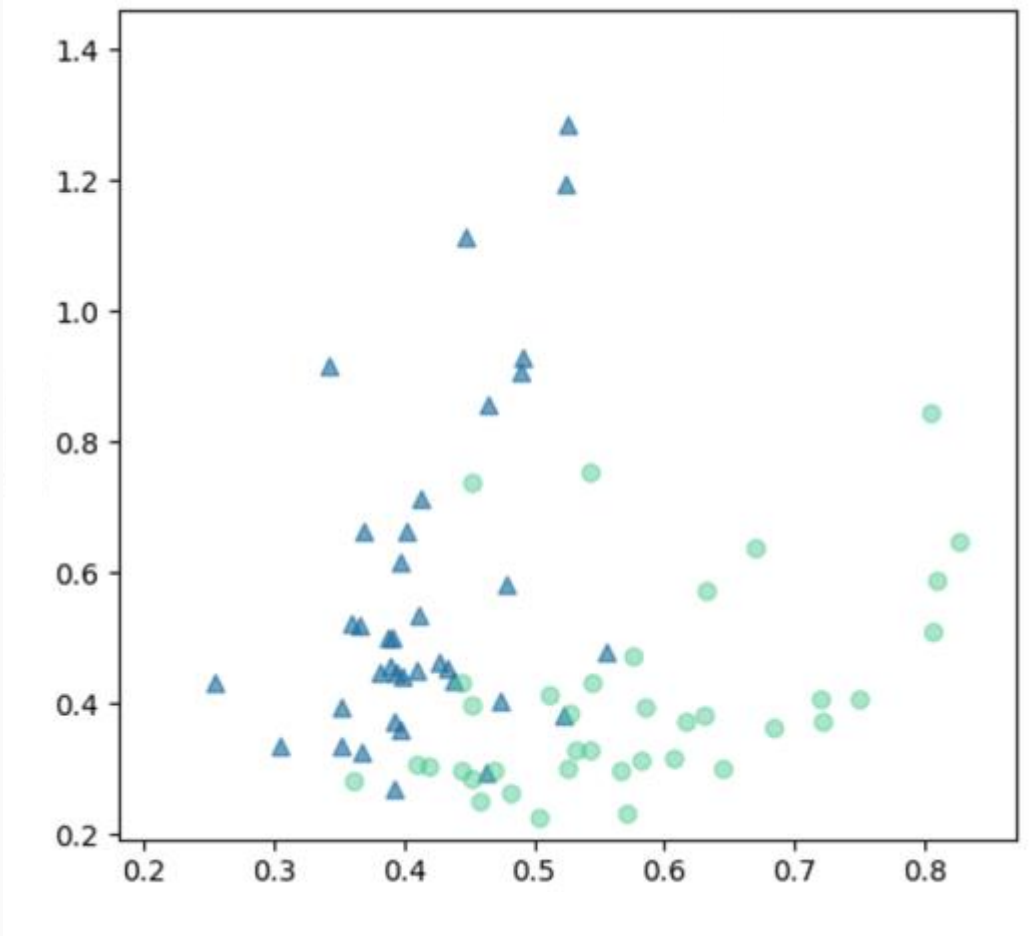
Utilizando el modelo para clasificar nuevos datos.





Clasificación: Prediciendo labels discretas

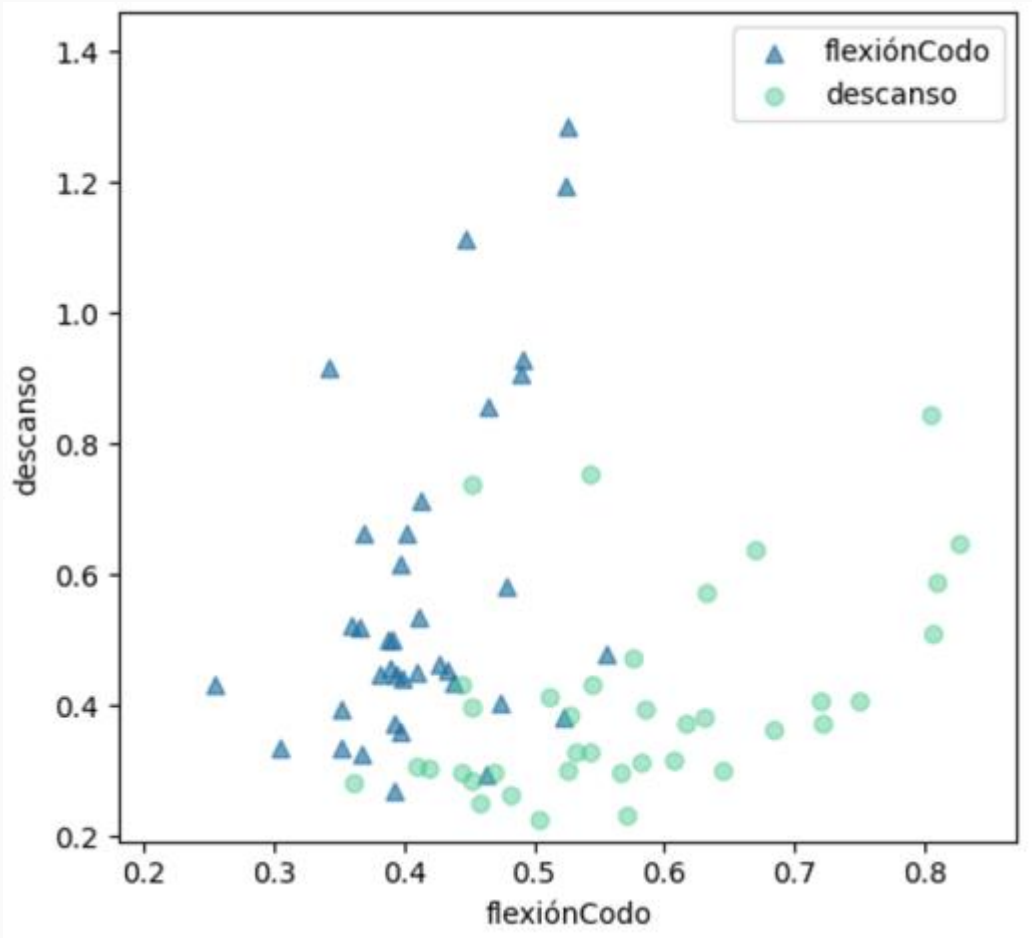
Otro ejemplo

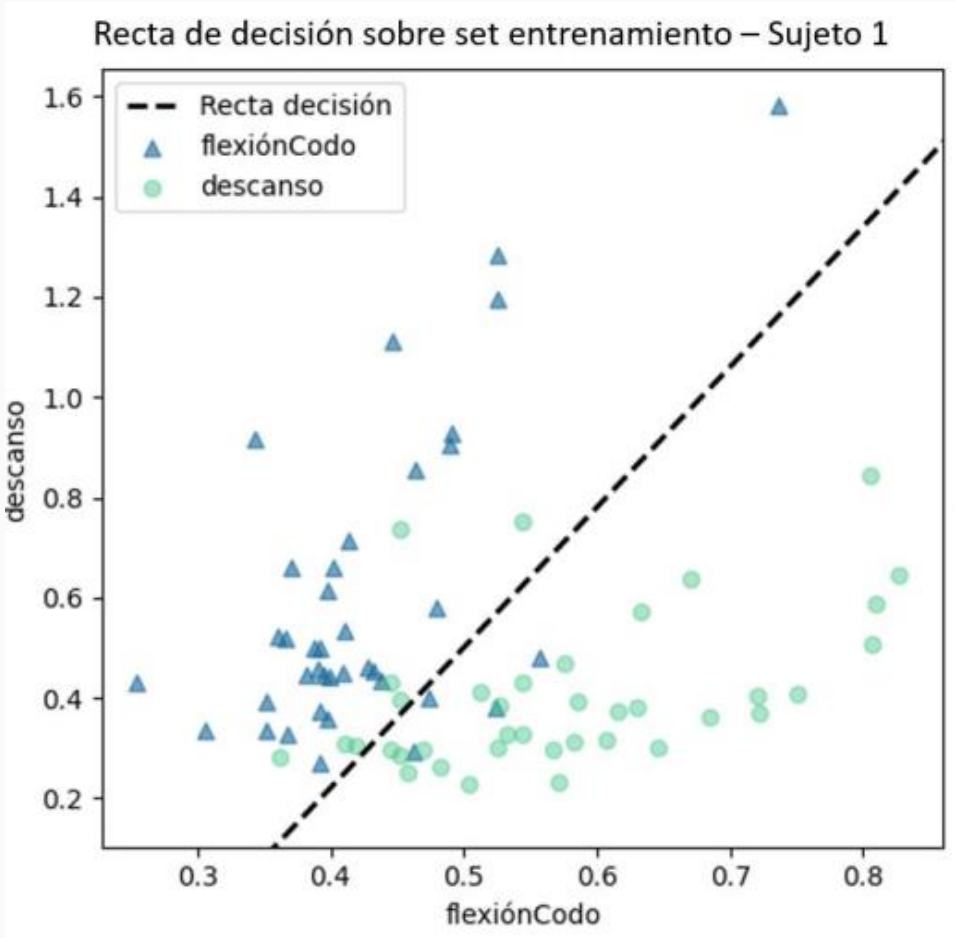


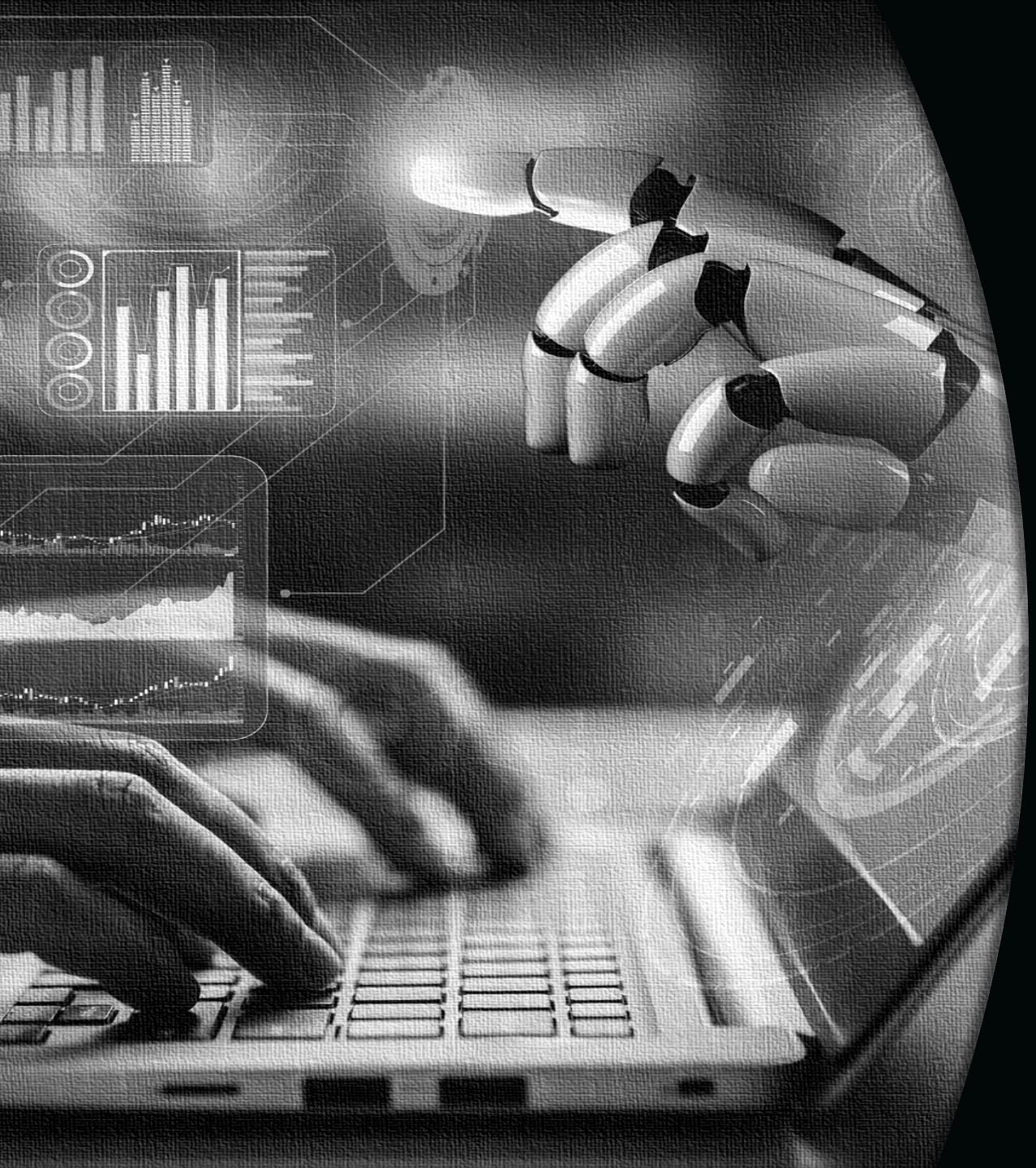
¿Estos puntos fueron generados aleatoriamente?

Clasificación: Prediciendo labels discretas

Clasificación de flexión de codo vs estado de reposo durante tarea de imaginería motora.



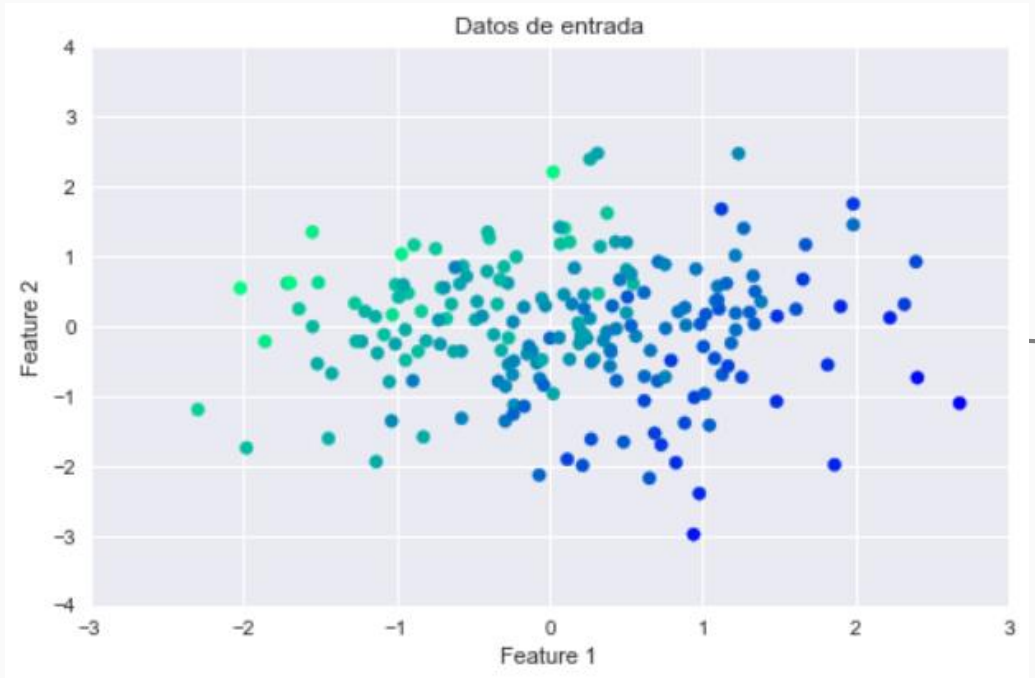




Ejemplo de Regresión

Regresión: Prediciendo valores continuos

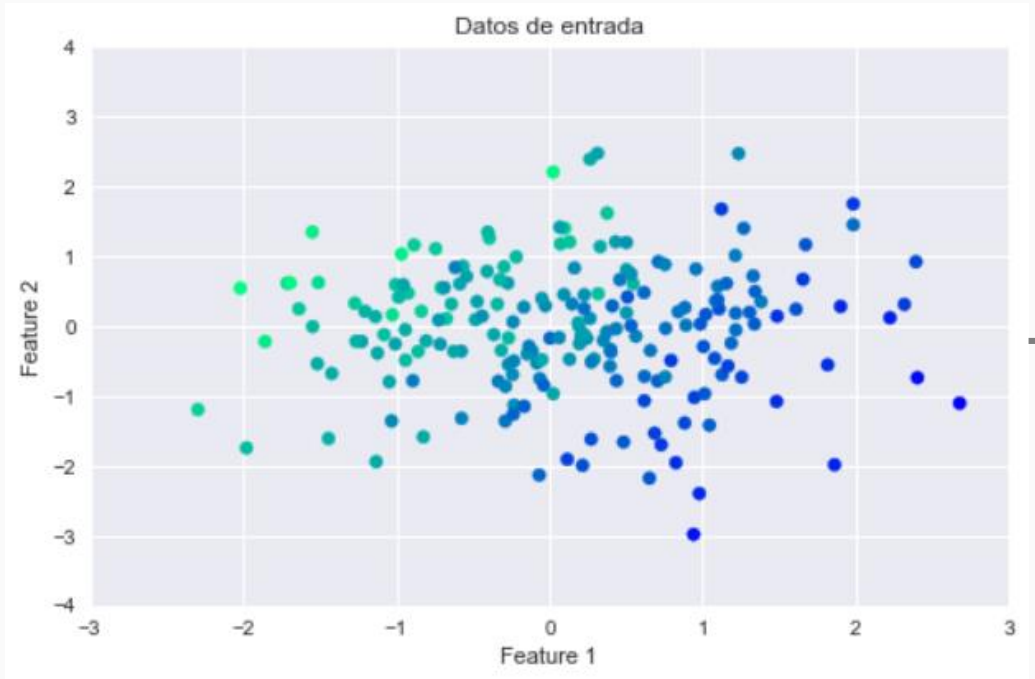
A diferencia de la tarea de clasificación donde tenemos labels *discretas*, en regresión tenemos labels *continuas.*

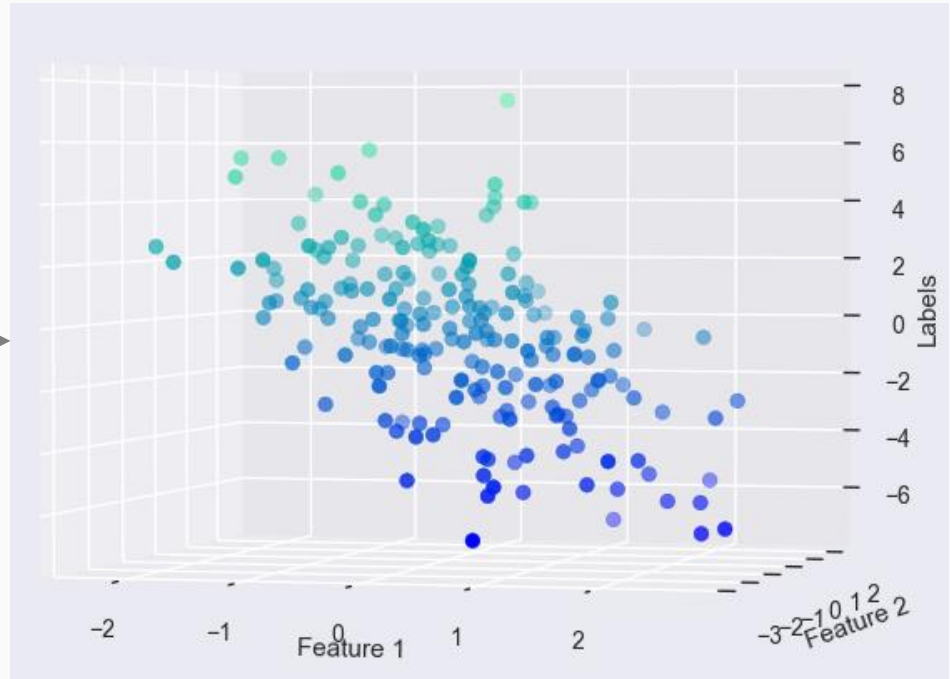



```
Posición de los puntos
[[ 1.3315865   0.71527897]
 [-1.54540029 -0.00838385]
 [ 0.62133597 -0.72008556]
 [ 0.26551159  0.10854853]
 [ 0.00429143 -0.17460021]
 [ 0.43302619  1.20303737]
 [-0.96506567  1.02827408]
 [ 0.22863013  0.44513761]
 [-1.13660221  0.13513688]
 [ 1.484537   -1.07980489]]

Etiquetas
[-1.48550848  1.8625612  -1.77018467  0.01297574 -1.81812721  0.77192227
  4.3562248   0.11254413  1.5168899  -4.62577223]
```

Regresión: Prediciendo valores continuos

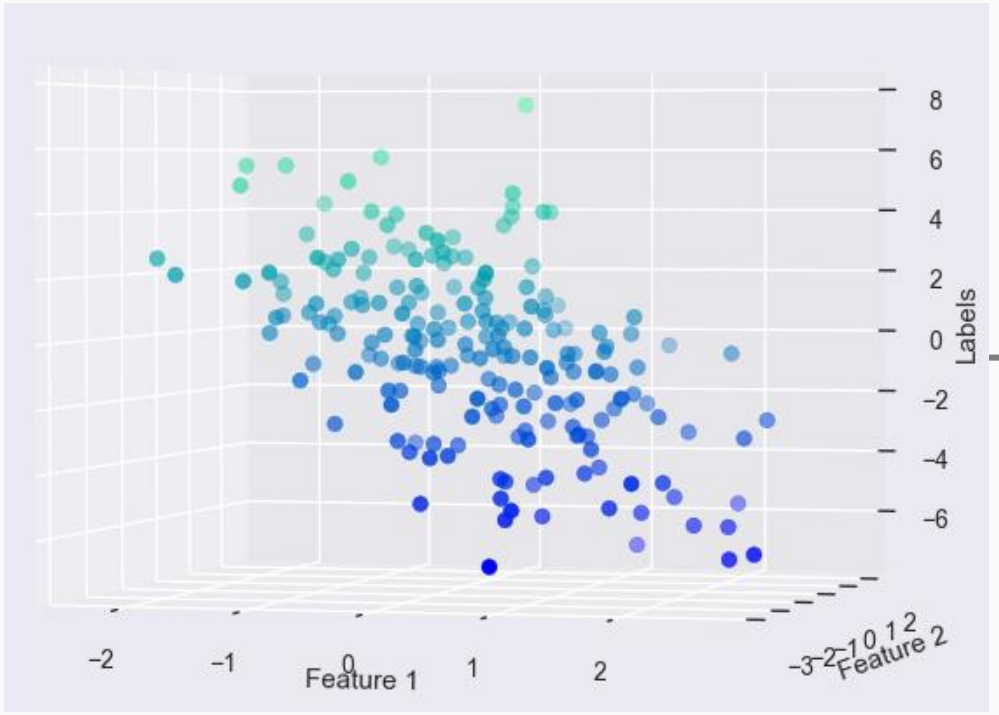
Podríamos pensar que cada punto tiene una *altura* diferente y dicha altura se corresponde con la etiqueta. Esto lo podemos representar en un gráfico de tres dimensiones.

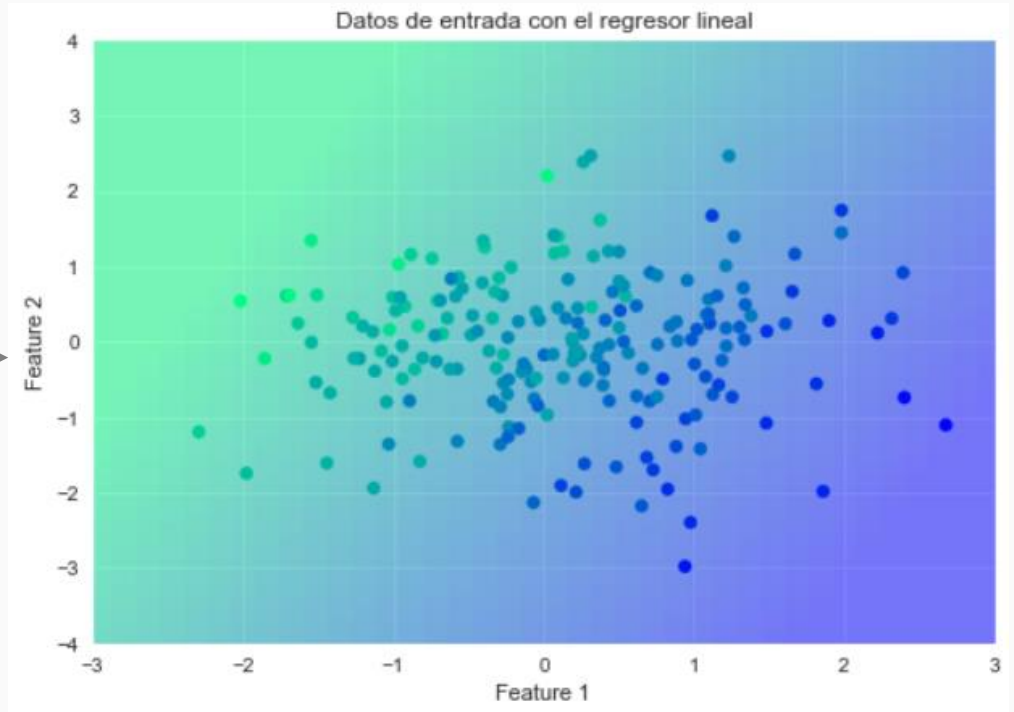




Regresión: Prediciendo valores continuos

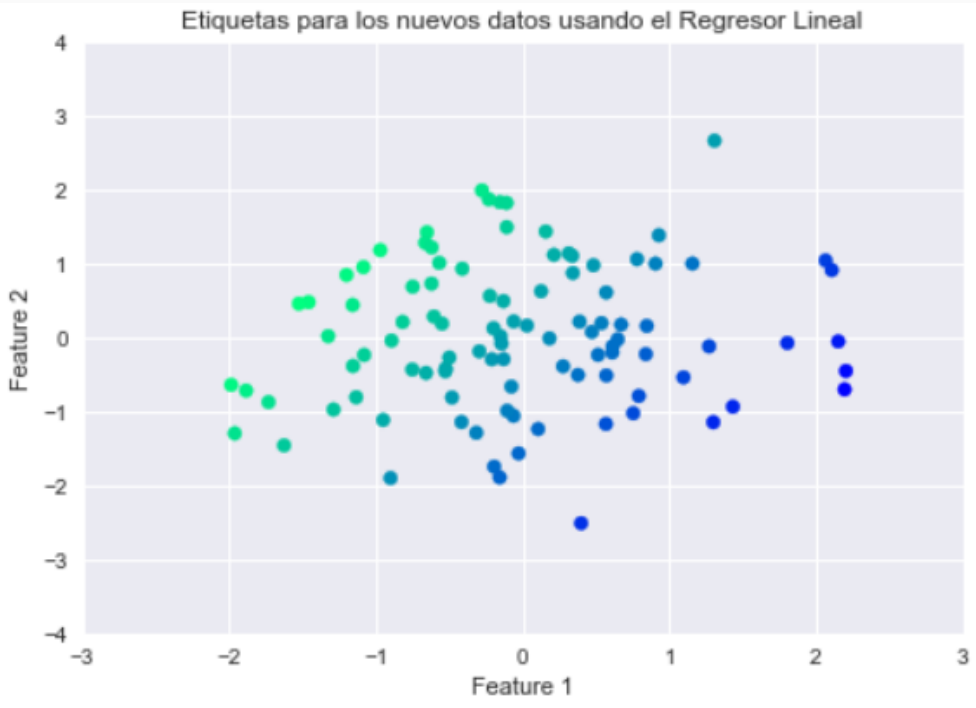
Podemos entrenar un LinearRegression() para obtener un plano que nos etiquete los datos.

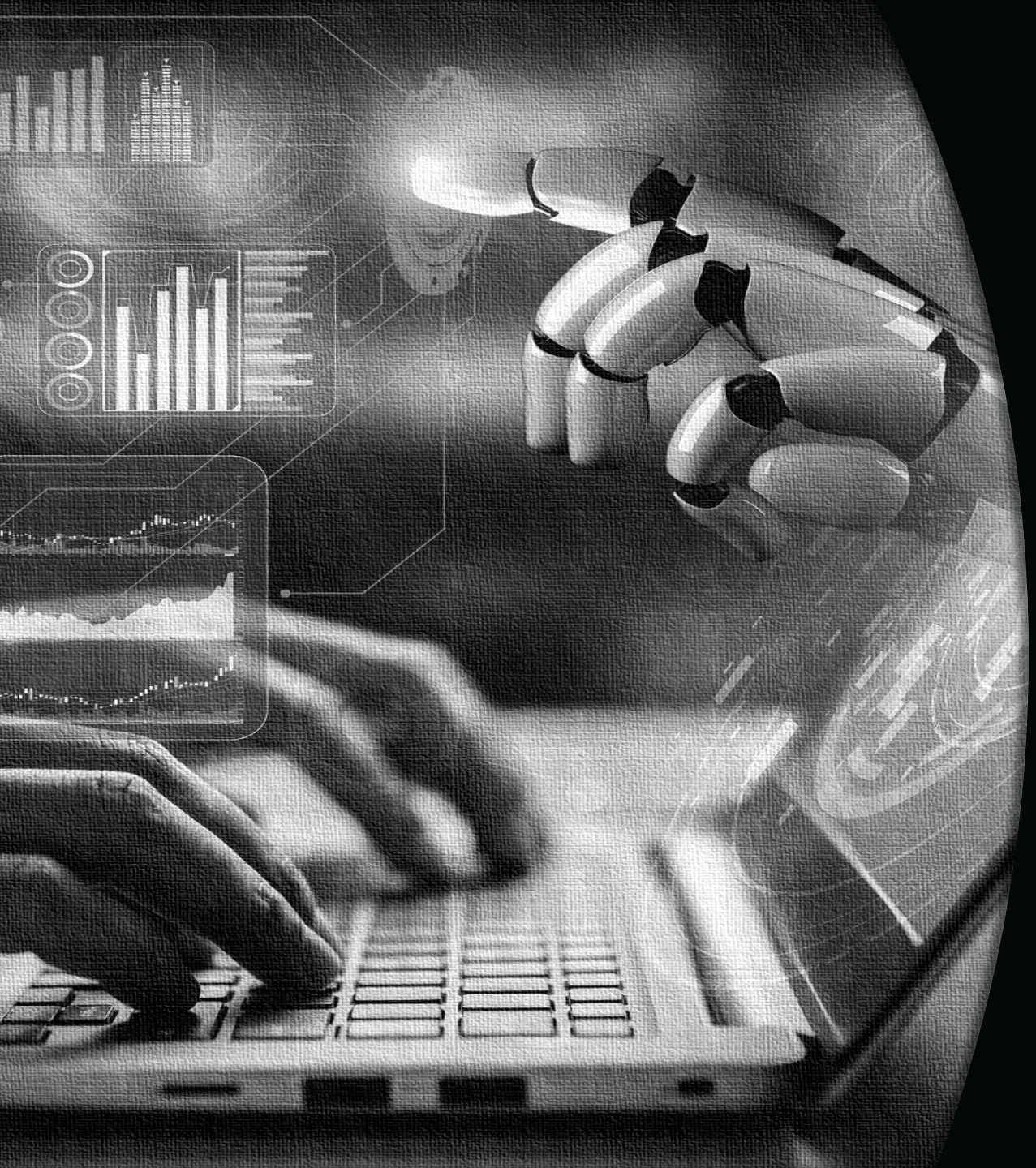




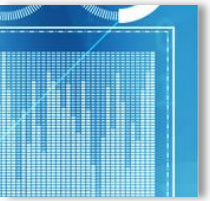
Regresión: Prediciendo valores continuos

Utilizando un Regresor Lineal podemos crear un plano para etiquetar nuevos datos a partir de los datos de entrada que ya tenemos.





Ejemplo de Clustering



Clustering: Obteniendo grupos con *K-means*

Hemos visto dos sencillos modelos de aprendizaje supervisado. Los mismos aprenden de datos etiquetados y una vez entrenados, sirven para etiquetar datos nuevos.

En el caso del aprendizaje no supervisado, el modelo *aprende* sin ninguna referencia a una etiqueta.

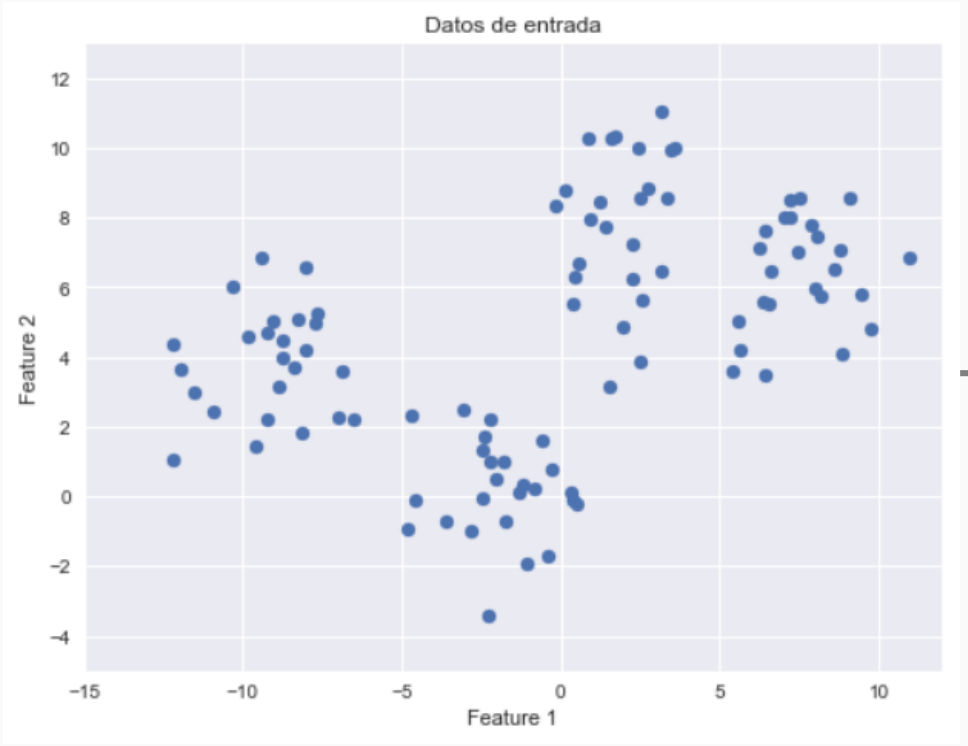
Un método muy conocido de aprendizaje no supervisado es Clustering, en donde los datos son asignados a un número n de grupos discretos.

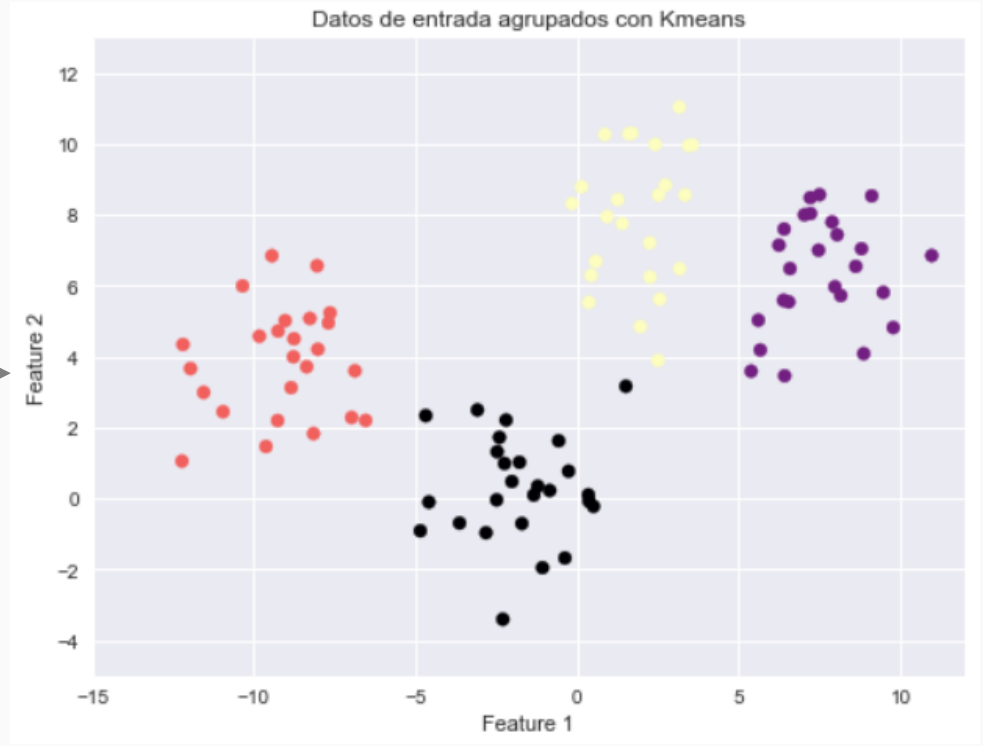
Clustering: Obteniendo grupos con *K-means*

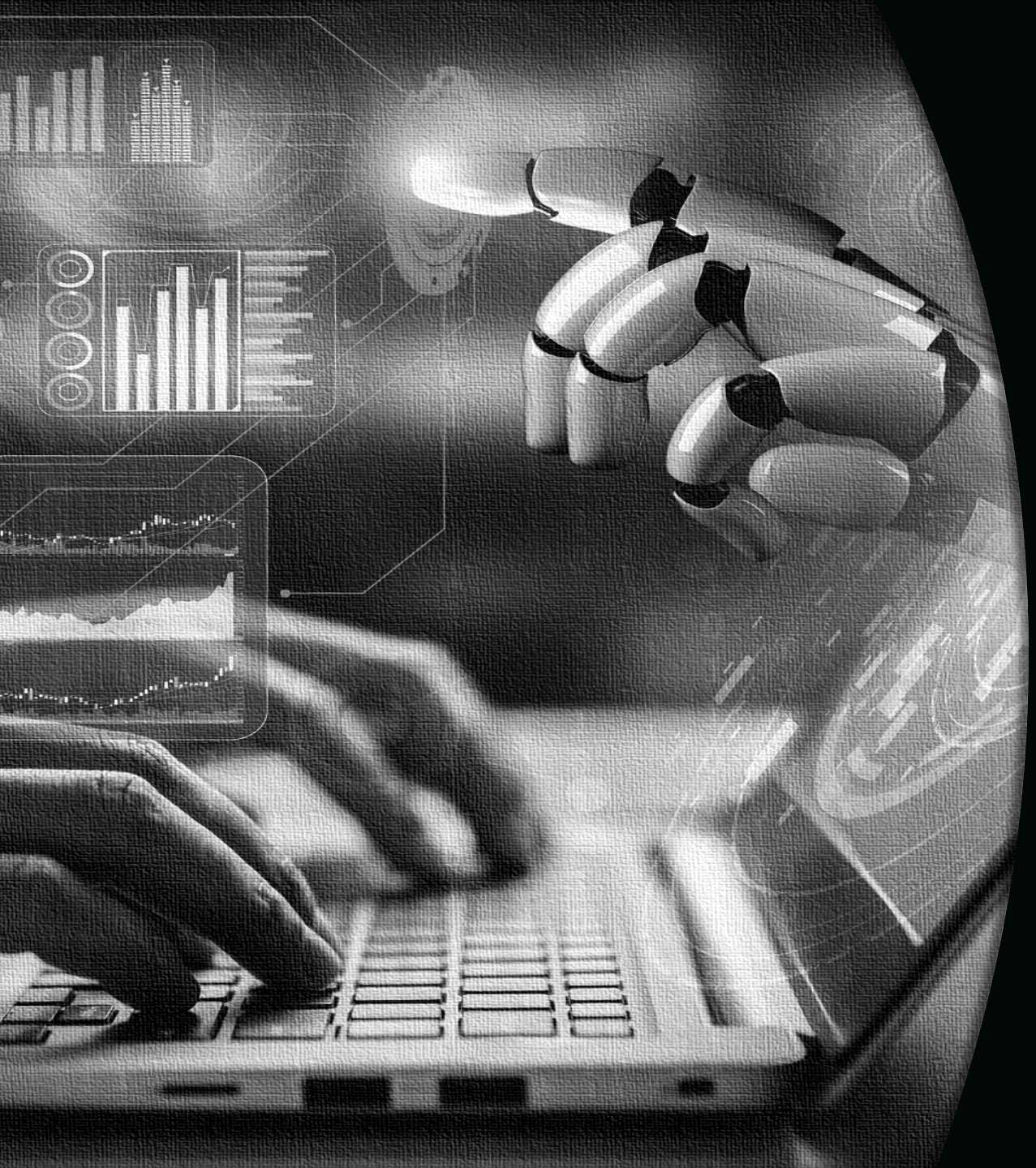
Supongamos el siguiente set de datos. Lo separaremos usando *Kmeans*.

Clustering: Obteniendo grupos con *K-means*

Supongamos el siguiente set de datos. Lo separaremos usando *Kmeans*.



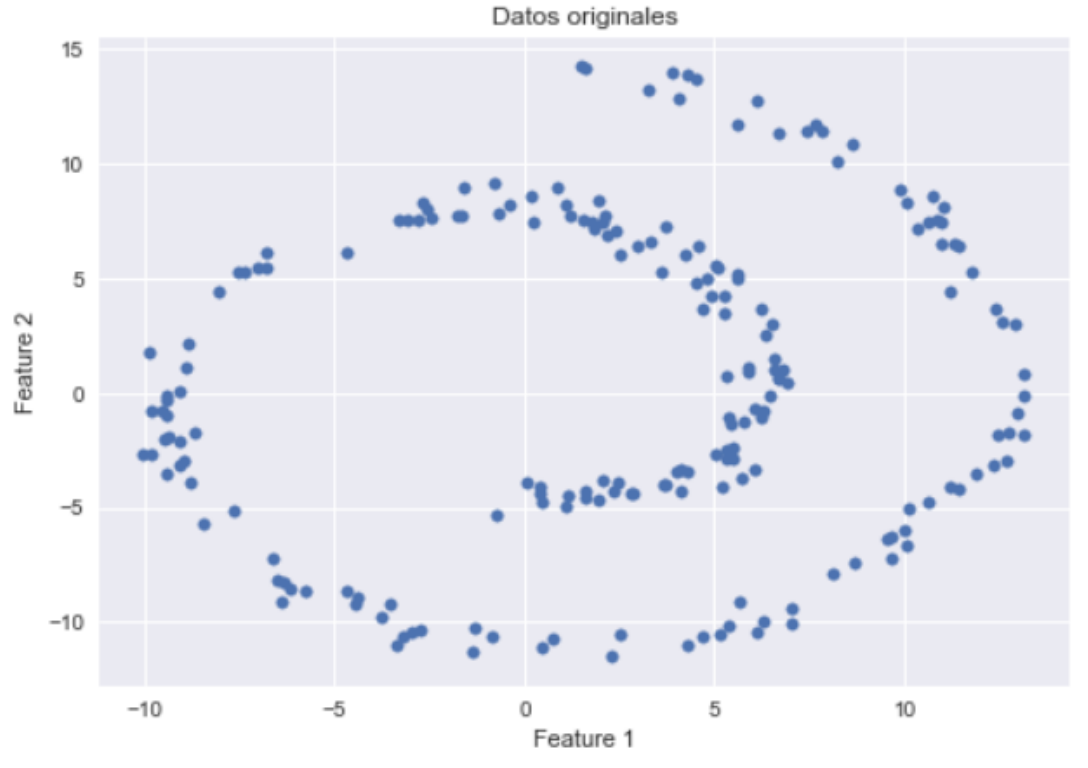




Reducción de dimensionalidad: infiriendo la estructura de mis datos.

Reduciendo la dimensiones de mis datos – Algoritmos Manifold

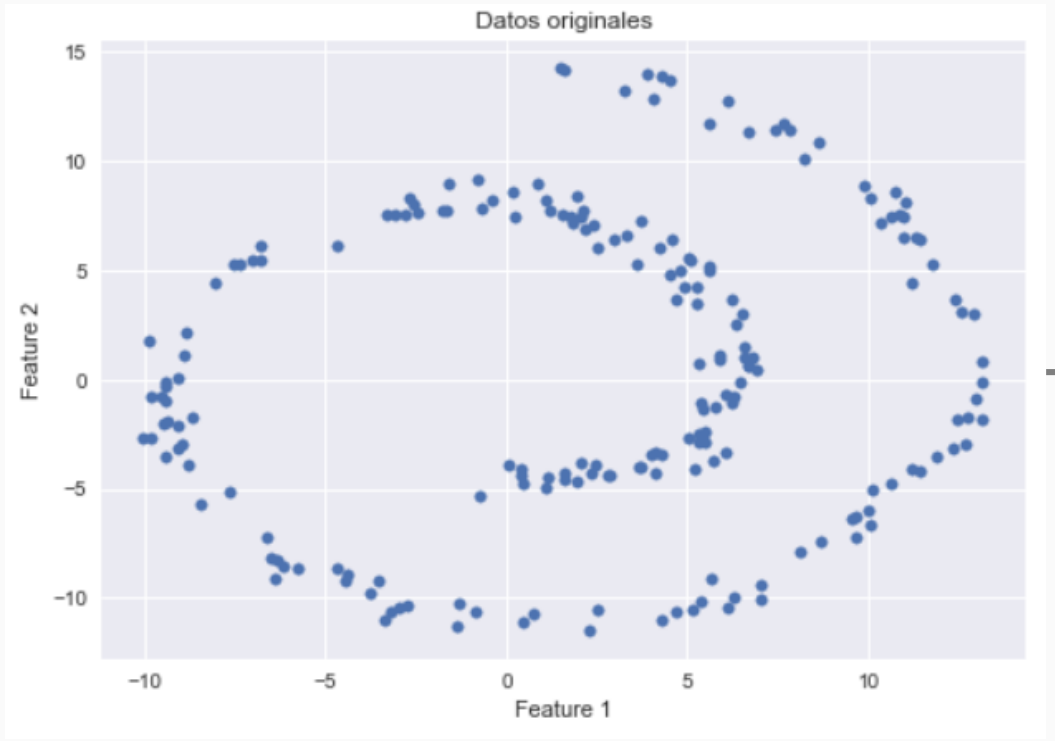
En muchos casos, los datos son N-dimensionales, con $N > 3$. Podría ser útil reducir la dimensión de nuestros datos, para analizarlos, para procesarlos, etc. Estos algoritmos son útiles cuando tenemos datos en dimensión 1000 y los hacemos de dimensión 3 o 2 para buscar patrones.

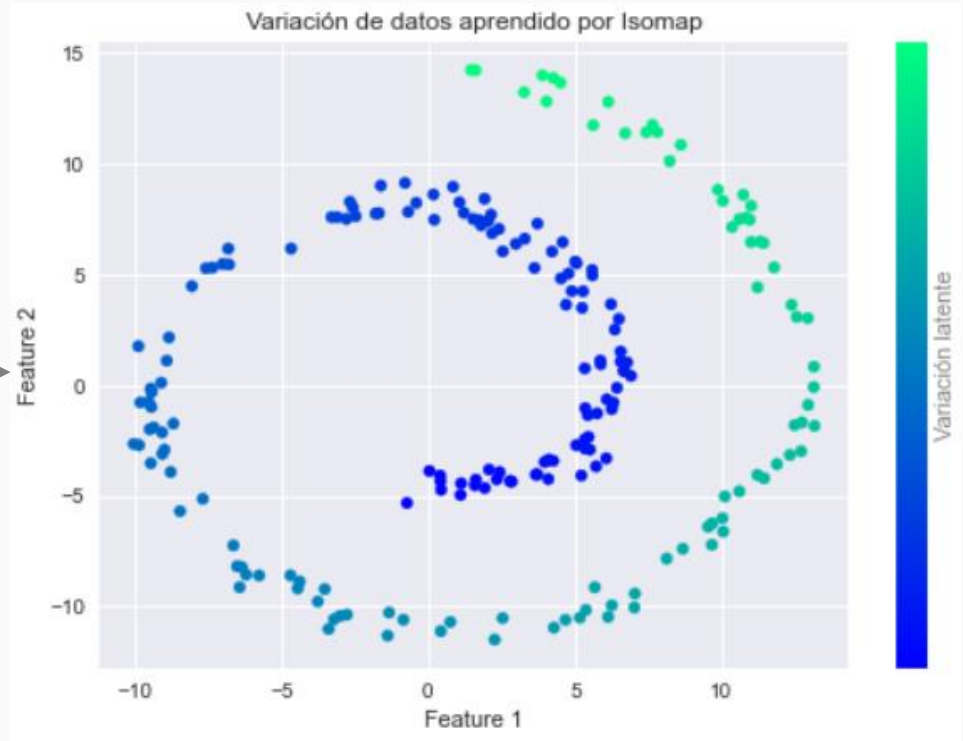


- ¿Cómo se distribuyen estos datos? ¿Linealmente o no?
- ¿Cómo se originaron?

Reduciendo la dimensiones de mis datos – Algoritmos Manifold

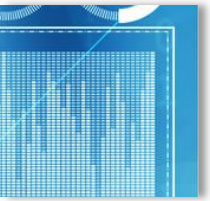
Con el algoritmo Isometric Mapping, reducimos la dimensión de los datos y por ende la complejidad del mismo, luego intentamos encontrar cómo estos varían o se generan.





Temas a tratar

- Intro: ¿Qué es el Machine Learning? objetivos, usos.
- Tipos de ML: Aprendizaje Supervisado y Aprendizaje No Supervisado. Ejemplos.
- Introducción al uso de Scikit-Learn.
- Resolución de un problema real.



Introducción a Scikit-Learn

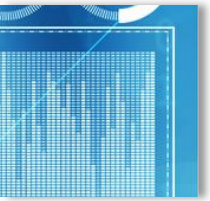
Sabemos que el ML trata de entrenar modelos a partir de datos para hacer algo con ellos.

Ahora bien, ¿qué estructura deben de tener los datos para poder entrenar algoritmos?

Datos como tablas

La forma básica es trabajar los datos como tablas.

Las columnas representan las *características* o *features.* El número de columnas me define la cantidad de *features,* es decir, $n\_features$.
Las filas las *observaciones,* también llamadas *muestras.* La cantidad de filas me define la cantidad de muestras, es decir, $n\_samples$.



Introducción a Scikit-Learn

Datos como tablas.

Set de datos *Iris*.

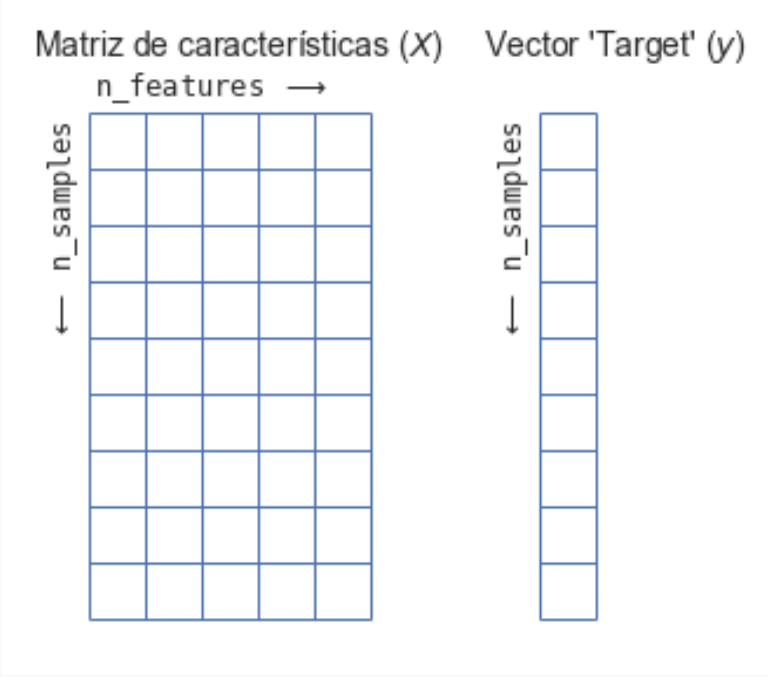
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Introducción a Scikit-Learn

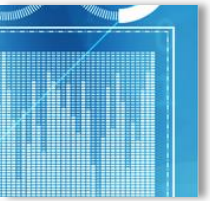
Matriz de Características: De forma $[n_{samples} \times n_{features}]$. En general se almacena en una variable llamada X. Suelen ser arreglos *numpy* o *DataFrames* de pandas.

Cada fila es una muestra, por ejemplo, una flor, una persona, una imagen, un video, etc. Cada columna representa una característica. En general contiene números reales, pero podrían ser valores discretos o booleanos.

Target vector: Es un arreglo generalmente unidimensional, aunque podría ser mayor ($[n_{samples} \times n_{targets}]$), que contiene $labels$. Suele almacenarse en una variable llamada y. En general contienen valores reales continuos o valores discretos.



Los valores en el vector de blancos son los que queremos predecir a partir de los datos, en términos estadísticos es la *variable dependiente.*



Introducción a Scikit-Learn

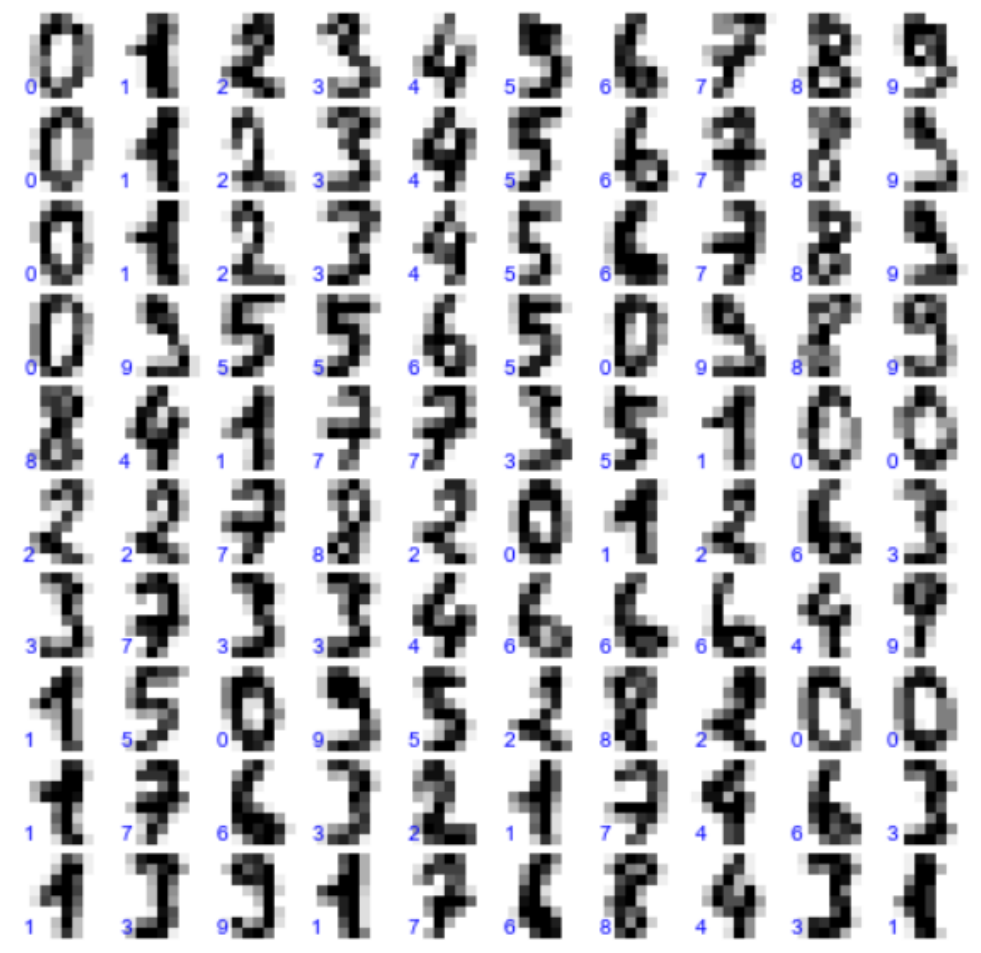
Pasos ***básicos*** para la implementación de un modelo.

1. Elegir un modelo adecuado al problema que queremos resolver/analizar.
2. Elegir los hiperparámetros del modelo instanciando esta clase con los valores deseados
3. Organizar los datos en una matriz de características y un vector blanco/target de la forma vista anteriormente.
4. Entrenar el modelo invocando al método *fit().*
5. Aplicar el modelo a nuevos datos:
   - Para aprendizaje supervisado, a menudo predecimos etiquetas para datos desconocidos usando el método de *prediction()*.
   - Para aprendizaje no supervisado, a menudo *transformamos* o *inferimos* propiedades de los datos utilizando el método *transform()* o *predict().*

Veamos un ejemplo…

Clasificando dígitos escritos a mano

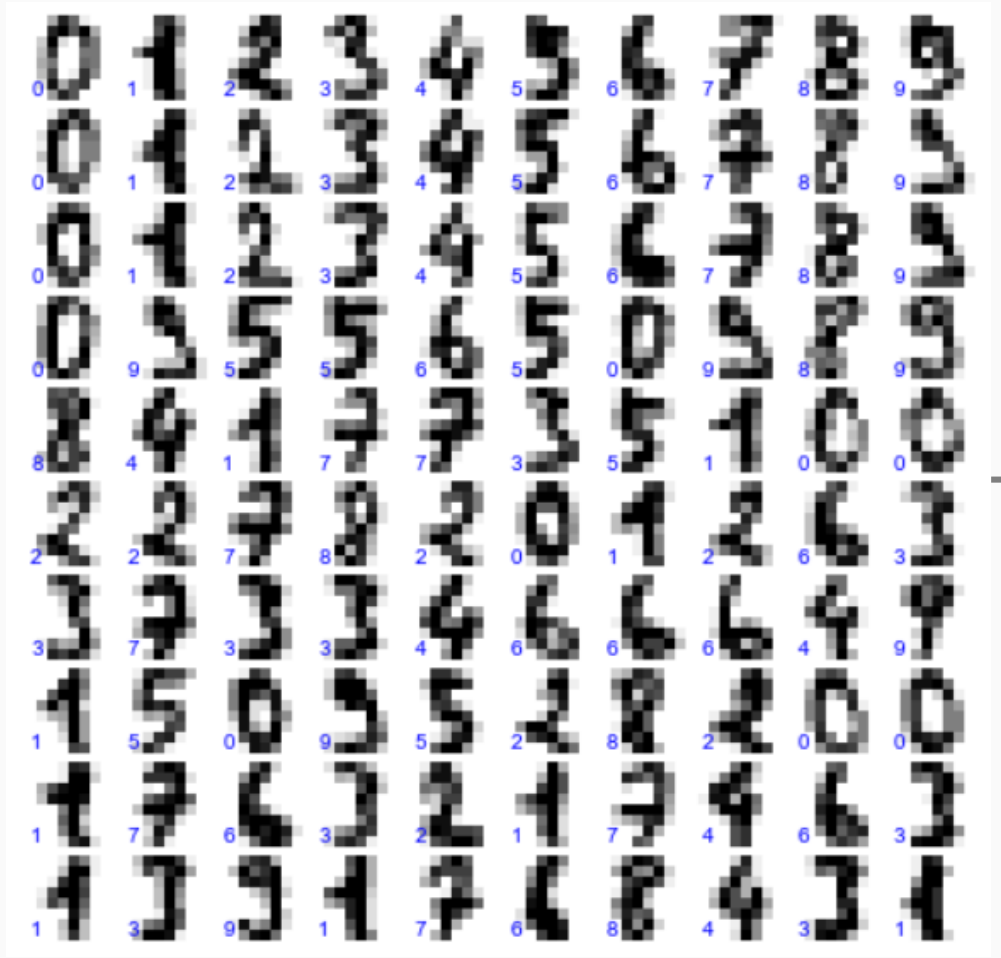
Intentaremos clasificar dígitos escritos a mano. El set de datos que utilizaremos esta conformado por 1797 dígitos de 8×8 pixéles.

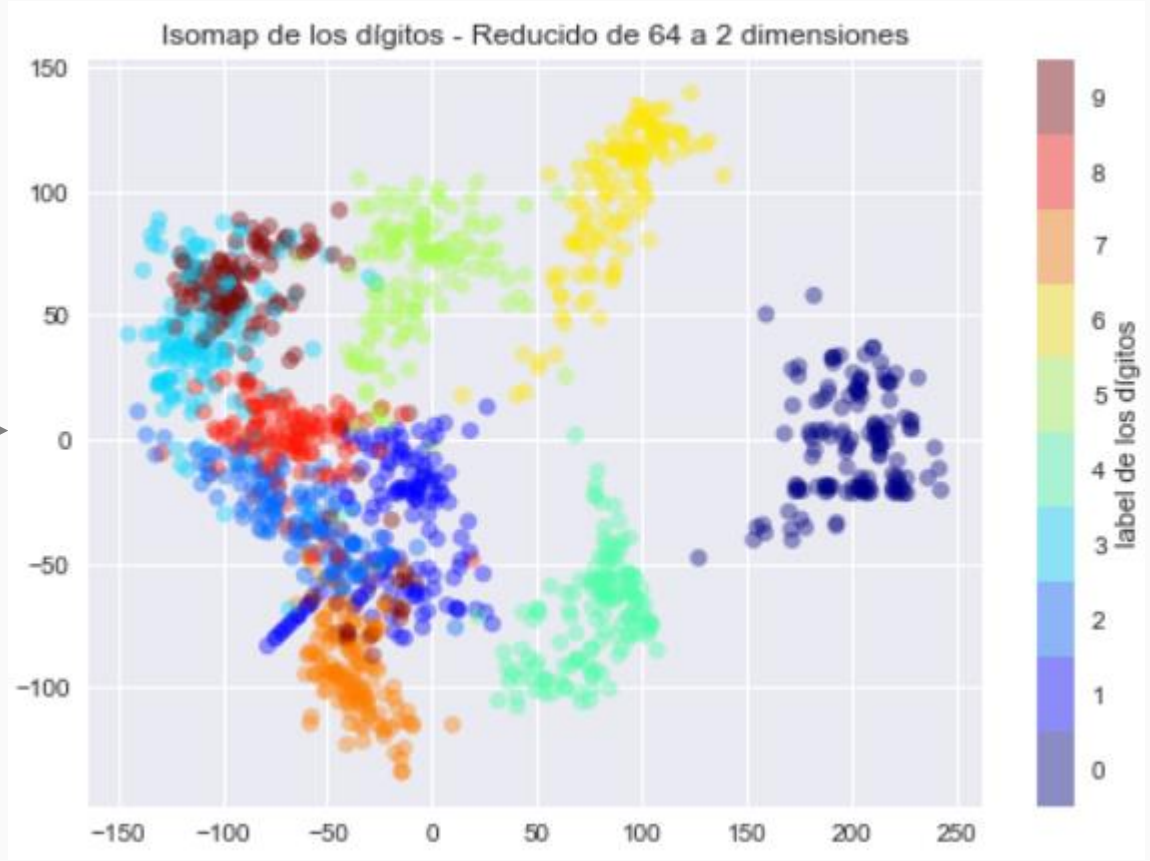


```
In [33]:  1 X = digits.data
          2 X.shape

Out[33]: (1797, 64)

In [108]: 1 print(X[:2])

[[ 0.  0.  5. 13.  9.  1.  0.  0.  0.  0. 13. 15. 10. 15.  5.  0.  0.  3.
  15.  2.  0. 11.  8.  0.  0.  4. 12.  0.  0.  8.  8.  0.  0.  5.  8.  0.
   0.  9.  8.  0.  0.  4. 11.  0.  1. 12.  7.  0.  0.  2. 14.  5. 10. 12.
   0.  0.  0.  0.  6. 13. 10.  0.  0.  0.]
 [ 0.  0.  0. 12. 13.  5.  0.  0.  0.  0.  0. 11. 16.  9.  0.  0.  0.  0.
   3. 15. 16.  6.  0.  0.  0.  7. 15. 16. 16.  2.  0.  0.  0.  0.  1. 16.
  16.  3.  0.  0.  0.  0.  1. 16. 16.  6.  0.  0.  0.  0.  1. 16. 16.  6.
   0.  0.  0.  0.  0. 11. 16. 10.  0.  0.]]

In [34]:  1 y = digits.target
          2 y.shape

Out[34]: (1797,)
```

Clasificando dígitos escritos a mano

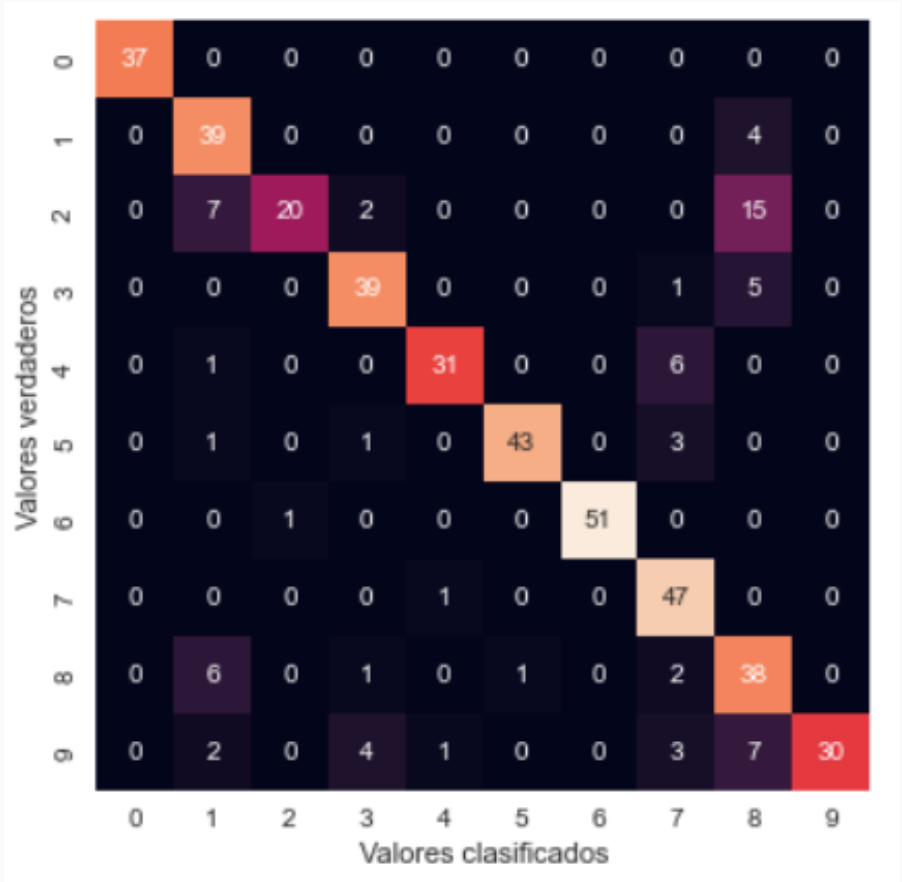
Utilizando aprendizaje no supervisado podemos reducir la dimensionalidad del set de datos original para buscar patrones dentro de los datos.



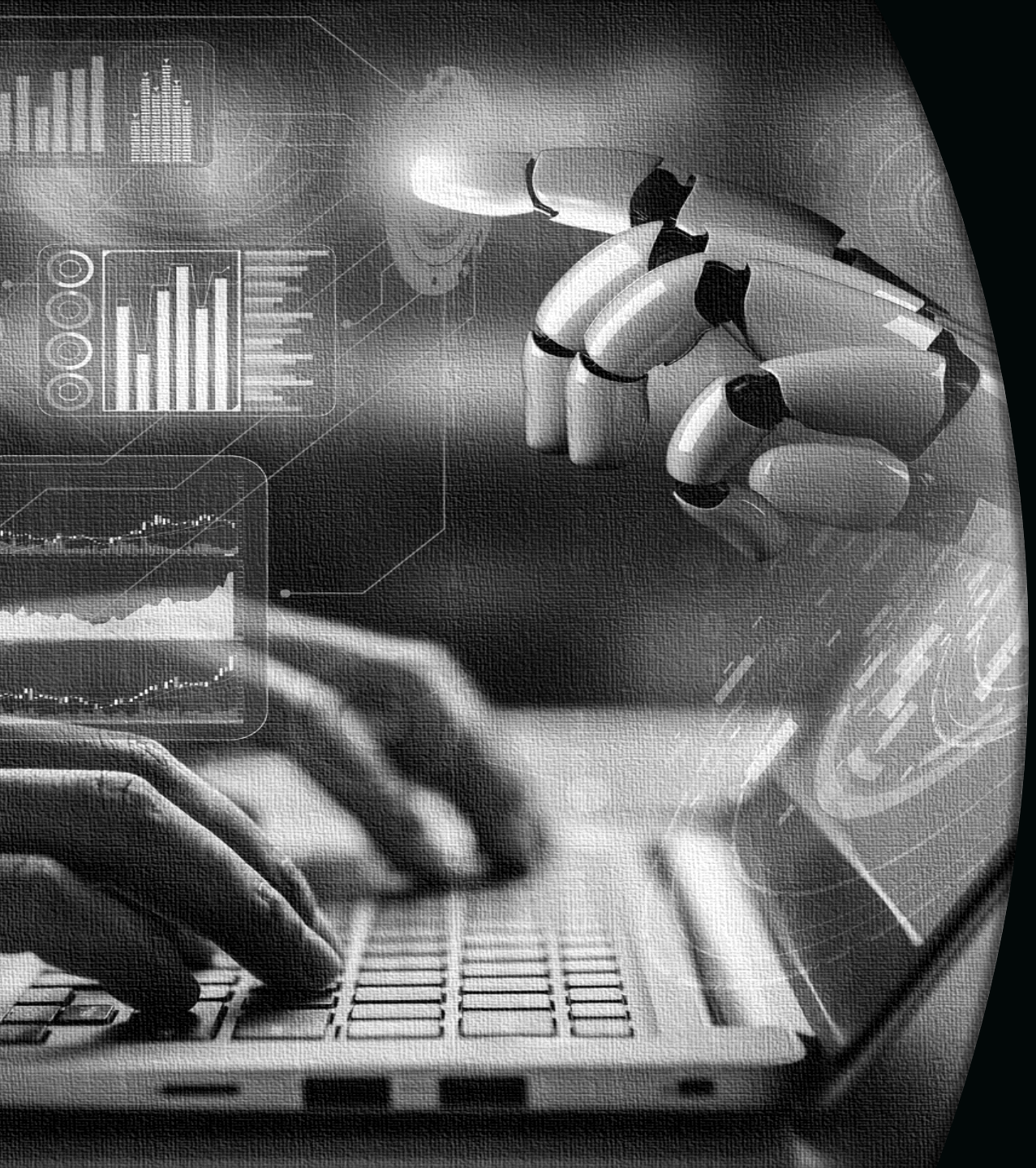


Clasificando dígitos escritos a mano

Utilizando un algoritmo conocido como *Gaussian Naive Bayes* podemos clasificar dígitos desconocidos, es decir, que no han sido vistos por el clasificador al entrenarse.



Matriz de confusión con los resultados del clasificador



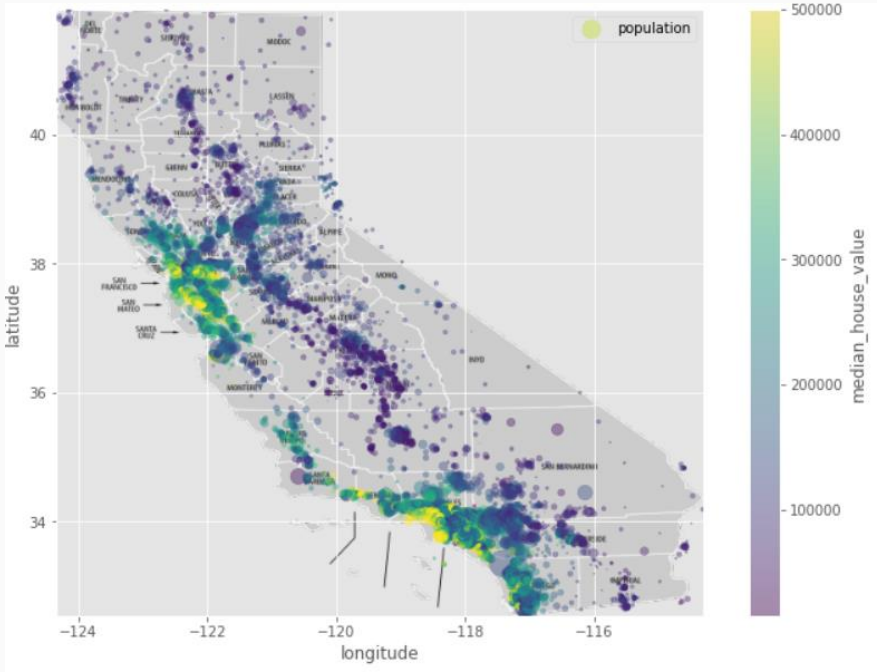
Temas a tratar

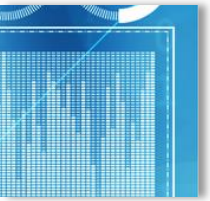
- Intro: ¿Qué es el Machine Learning? objetivos, usos.
- Tipos de ML: Aprendizaje Supervisado y Aprendizaje No Supervisado. Ejemplos.
- Introducción al uso de Scikit-Learn.
- Resolución de un problema real.

Prediciendo precios de casas

En este ejercicio intentaremos predecir el precio de casas ubicadas en la región de California (EEUU). El dataset a utilizar se llama *California Housing Prices dataset* del repositorio de StatLib.

La idea es entrenar un modelo que aprenda de los datos y sea capaz de predecir el precio medio de una casa de cualquier distrito en base a diferentes métricas y características dentro del set de datos.





Prediciendo precios de casas

La idea del ejercicio es,

1. Entender el problema.
2. Obtener los datos.
3. Analizar y obtener pistas, perspectivas e ideas de los datos que nos sean de utilidad para seleccionar y entrenar un modelo adecuado.
4. Preparar los datos para trabajar con algoritmos de ML.
5. Seleccionar y entrenar un modelo.

GRACIAS POR SU ATENCIÓN

¿Preguntas?

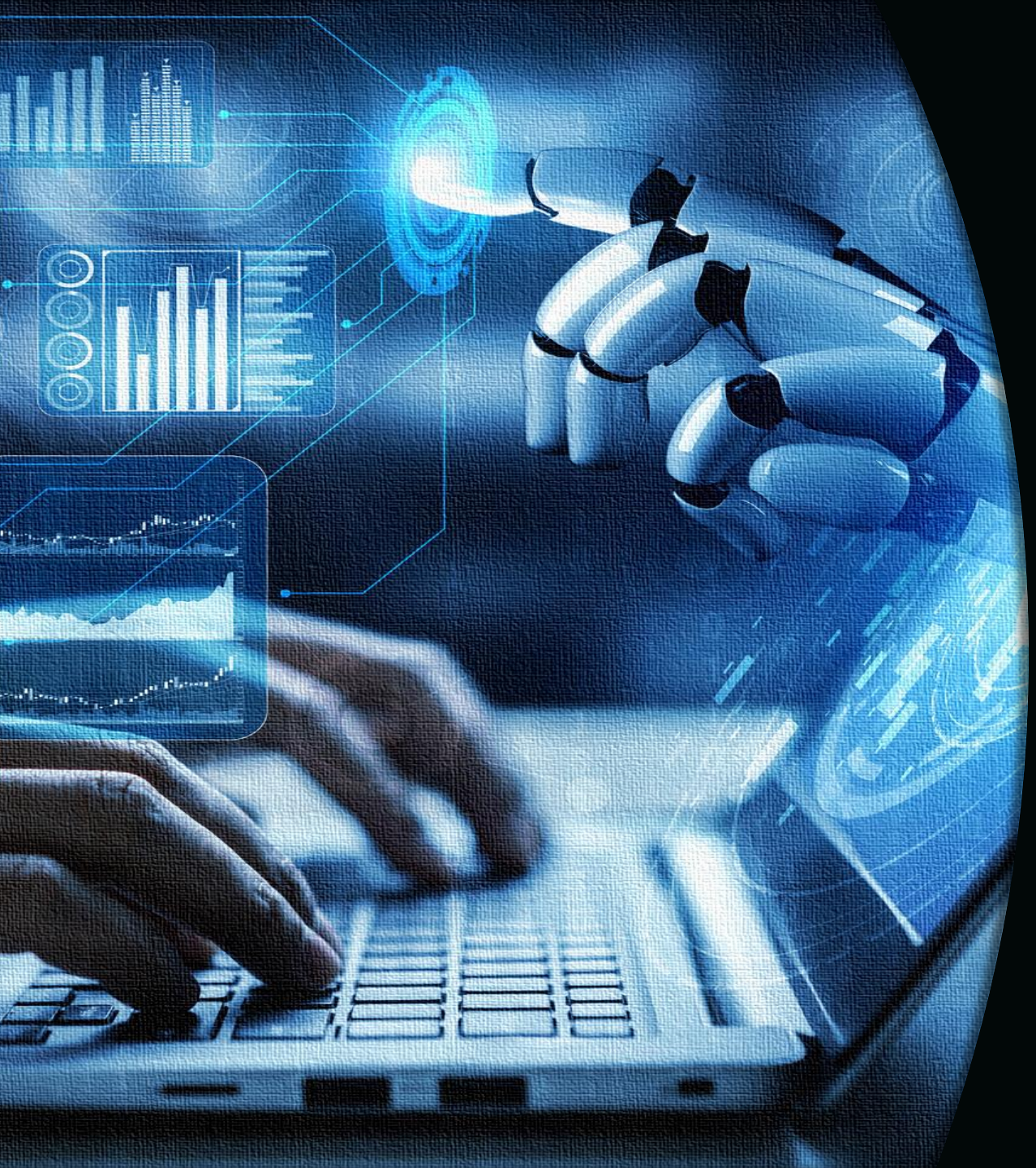
Bibliografía

[1] *“Python Data Science Handbook”* – Jake VanderPlas – Primera Edición 2016.

[2] *“Python Machine Learning”* – Sebastian Raschka y Vahid Mirjalili – Segunda edición 2017.

[3] *“Data Science from scratch”* – Joel Grus – Primera edición 2015.

[4] *“Documentación official del paquete ScikitLearn”* - https://scikit-learn.org/stable/index.html - Última visita 30/9/2022.



UTEC
Universidad Tecnológica

Hackathon Interfaces Cerebro Computadora
2022/2023
Taller N° 2

Introducción al Machine Learning

MSc. Bioing. BALDEZZARI Lucas

Profesor Adjunto
Ingeniería Biomédica

Fray Bentos, Uruguay

5 de octubre de 2022