Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2115 - Programación como herramienta para la ingeniería

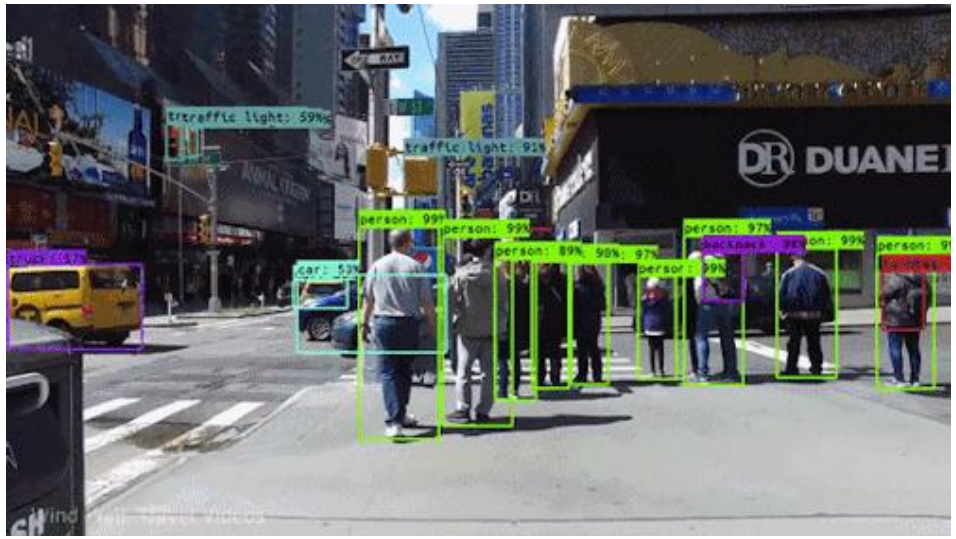
Modelos Predictivos basados en Machine Learning

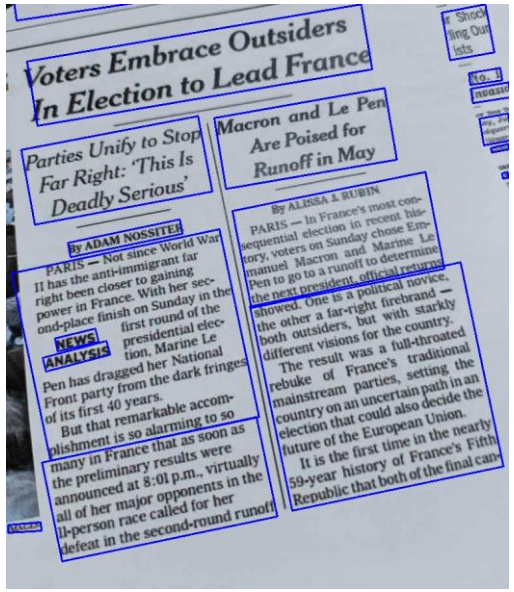
Profesor: Hans Löbel

¿Cómo puedo saber si en una foto hay un perro o un gato?

(usando un computador y sin preguntarle a un asistente, obviamente)

¿  ?




Voters Embrace Outsiders In Election to Lead France

Parties Unify to Stop Far Right: 'This Is Deadly Serious'

Macron and Le Pen Are Poised for Runoff in May

By ADAM NOSSITER

By ALISSA J. RUBIN

PARIS — Not since World War II has the anti-immigrant far right been closer to gaining power in France. With her second-place finish on Sunday in the first round of the presidential election, Marine Le Pen has dragged her National Front party from the dark fringes of its first 40 years.

But that remarkable accomplishment is so alarming to so many in France that as soon as the preliminary results were announced at 8:01 p.m., virtually all of her major opponents in the 11-person race called for her defeat in the second-round runoff

PARIS — In France's most consequential election in recent history, voters on Sunday chose Emmanuel Macron and Marine Le Pen to go to a runoff to determine the next president, official returns showed. One is a political novice, the other a far-right firebrand — both outsiders, but with starkly different visions for the country.

The result was a full-throated rebuke of France's traditional mainstream parties, setting the country on an uncertain path in an election that could also decide the future of the European Union.

It is the first time in the nearly 59-year history of France's Fifth Republic that both of the final can-

Generated Question	Answer
How many bears are laying on the ice?	“two”
What are the two animals laying on the ice?	“bears”
What are the bears doing?	“laying down”
Two bears are laying down on what?	“ice”
Where are the bears laying?	“on the ice”
Are the bears on the ice?	“yes”

A dog wearing a Superhero outfit with red cape flying through the sky

En simple, Machine Learning será para nosotros el estudio sobre cómo hacer predicciones a partir de ejemplos

- Adecuado para problemas donde la programación “tradicional” no es posible (reglas, algoritmos o procedimientos explícito).
- Éxito o fracaso depende del rendimiento en ejemplos no “vistos” previamente.
- Metodologías y protocolos para “entrenar” buenos modelos son el foco de este capítulo.

Con respecto a la herramienta específica para ML, en este capítulo nos centraremos en scikit-learn

- Implementa gran cantidad de algoritmos y modelos predictivos, basados en Machine Learning.
- Clasificación, regresión, *clustering*, reducción de dimensionalidad, selección de modelos y más.
- Permite una fácil integración con Pandas y librerías de visualización.

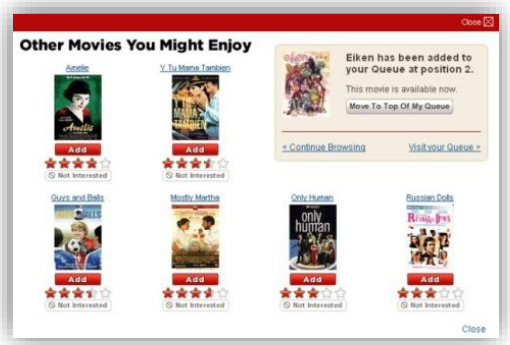
Antes de revisar *scikit-learn* y el tipo de problemas en este capítulo, necesitamos una breve introducción a los elementos centrales de Machine Learning

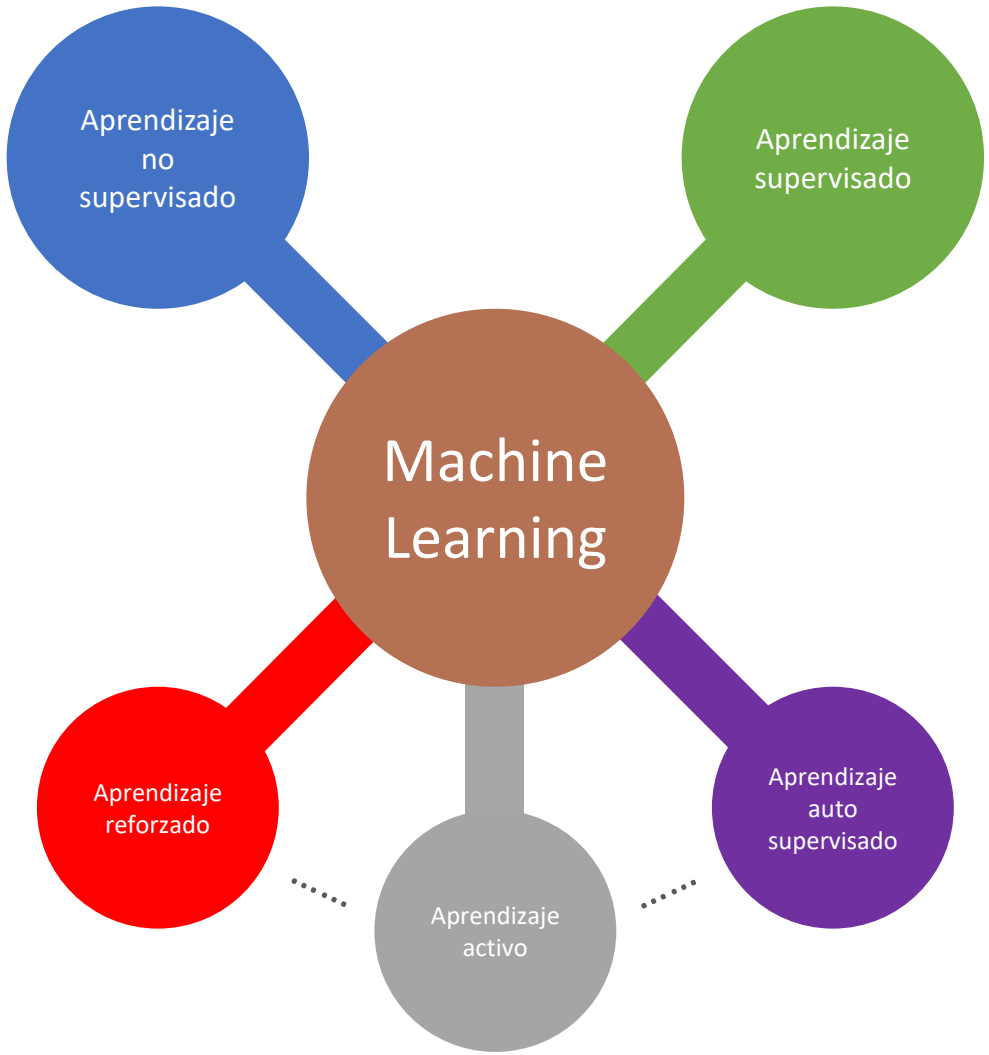
Ahora con algo más de detalle,
¿Qué es Machine Learning?

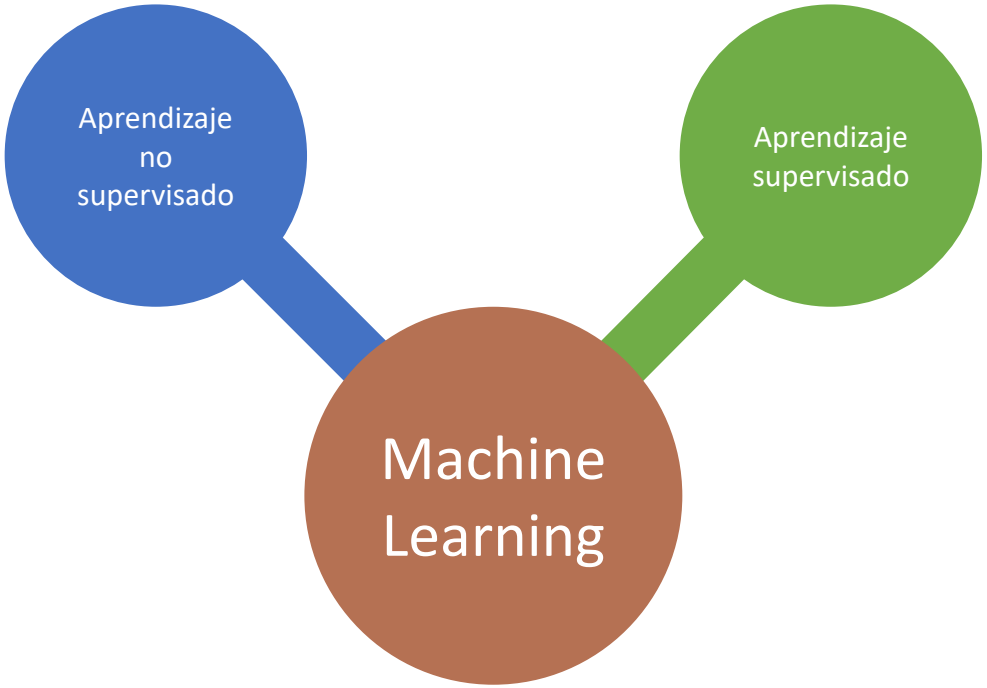
- Técnicas computacionales que usan datos para realizar tareas predictivas (clasificación, regresión, reducción de dimensionalidad, clustering, etc.)

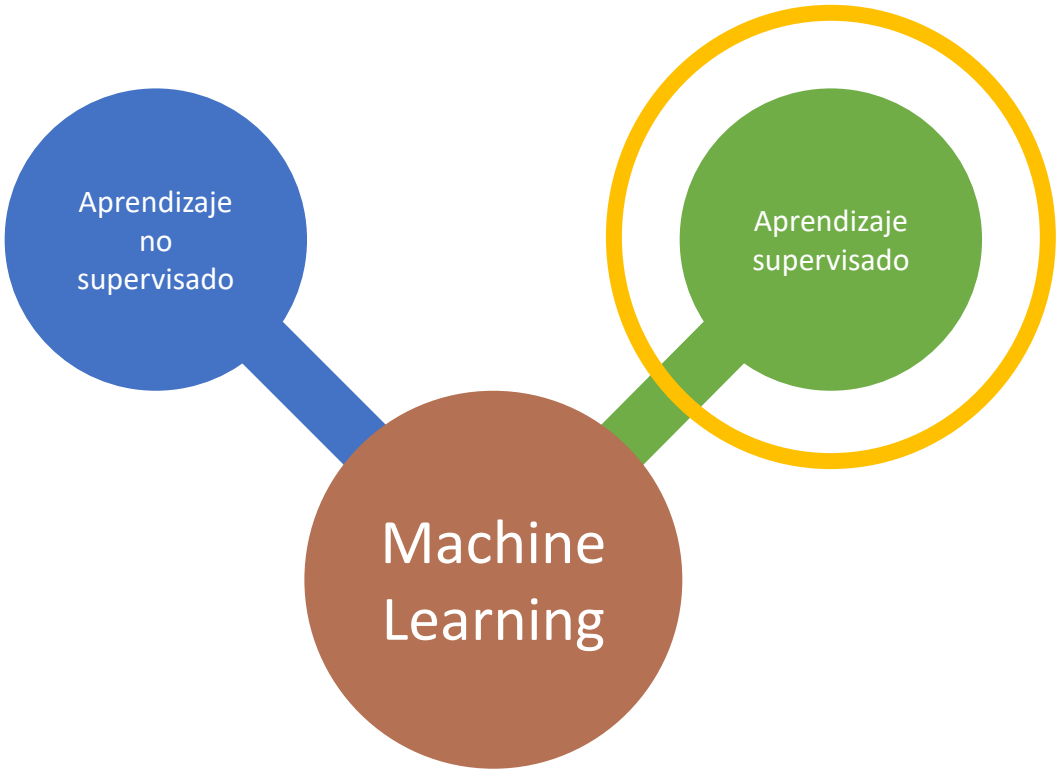
- Típicamente, estas técnicas (modelos, algoritmos, etc.) mejoran su rendimiento predictivo, a través de la experiencia → aprendizaje desde los datos.

- Foco es distinto a la estadística: técnicas de ML buscan realizar tareas/predicciones con la mayor precisión posible, más que entender el fenómeno subyacente.




Aprendizaje no supervisado
Aprendizaje supervisado
Machine Learning
Aprendizaje reforzado
Aprendizaje activo
Aprendizaje auto supervisado


Aprendizaje no supervisado
Aprendizaje supervisado
Machine Learning


Aprendizaje
no
supervisado
Aprendizaje
supervisado
Machine
Learning

Cat

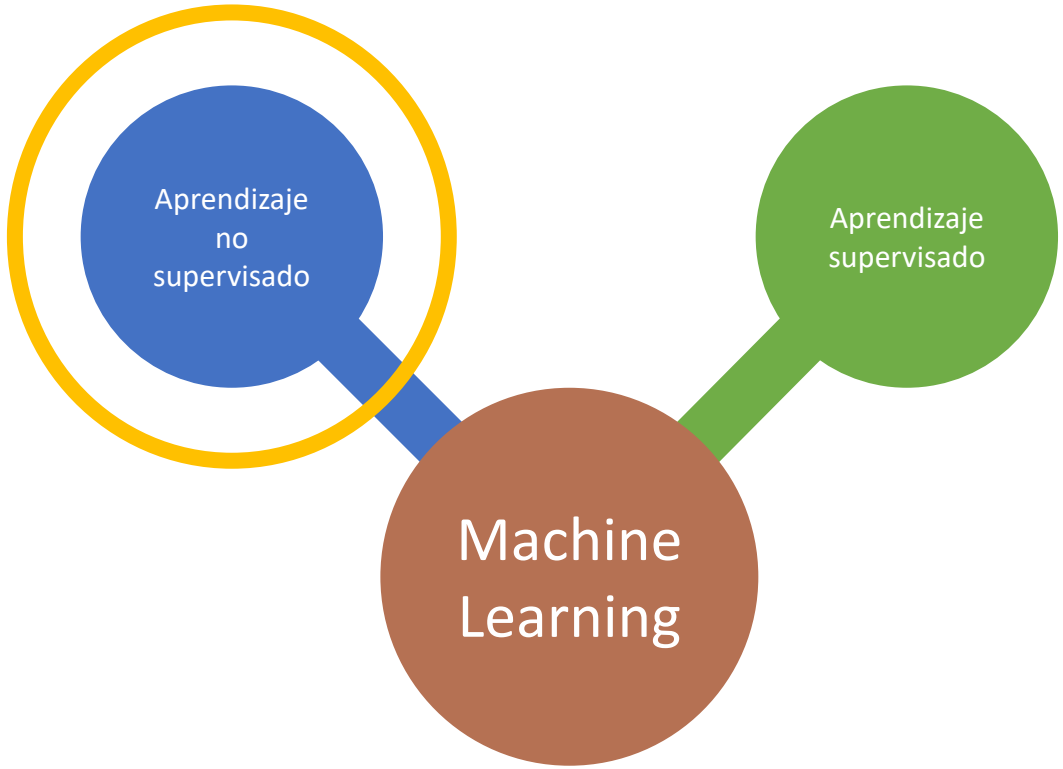
Dog

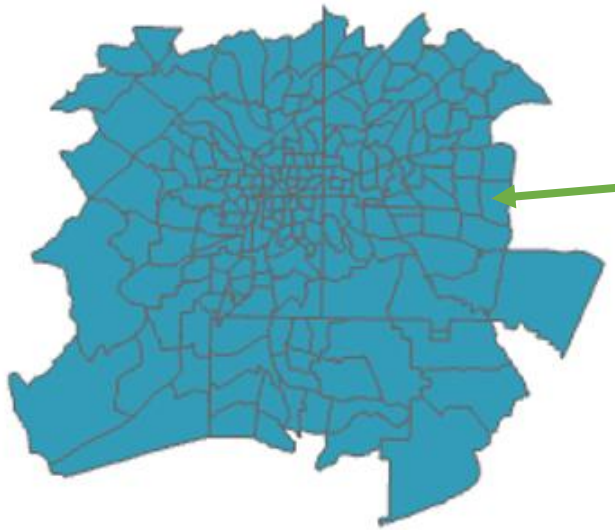
¿  ?

Cat

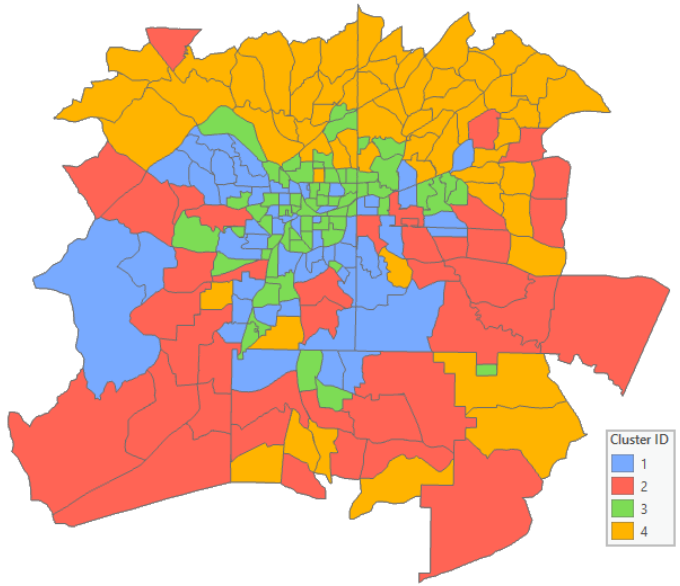
¿  ?

Dog


Aprendizaje
no
supervisado
Aprendizaje
supervisado
Machine
Learning



Type	Level	Description
Location	I/B[†]	Geocoded position of the unit
SES	I B	Socioeconomic status
Aesthetic	I/B	Aesthetic perception index of the nearest geocoded image (individual) or averaged at the urban block level
Political	I/B	Proportion of a political choice in the Constitutional plebiscite or in the Chilean Presidential Elections (first round)
Land use	I/B	Proportion/M^{2}[‡] of land use of the urban block according to a land use typology*
Demographic	I B	Sex, age, proportion of immigrants Age, proportion of immigrants/women



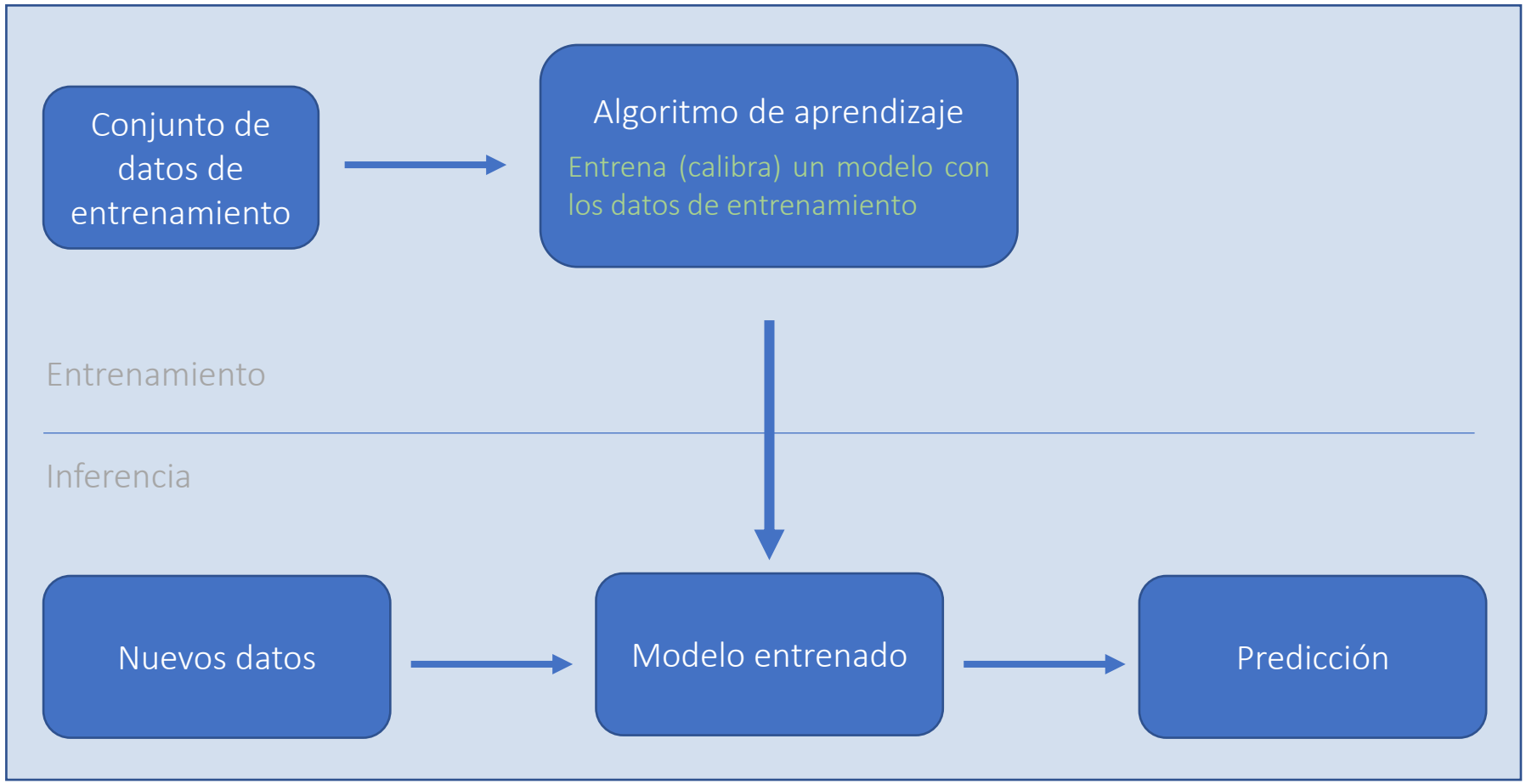
Como problema guía para lo que viene, estudiaremos la siguiente pregunta:

¿Cómo puedo estimar el ingreso anual promedio per cápita para los habitantes de una ciudad?

Como insumo para el estudio, tenemos un archivo `.csv` con datos diversos para múltiples ciudades

Ciudad	Población	Superficie (km^2)	Densidad poblacional por km^2	Ingreso anual promedio per cápita en dólares
Metropolis	3.500.000	600	5.833	$45.000
Rivertown	1.200.000	150	8.000	$35.000
Coastline	2.800.000	300	9.333	$40.000
Highland	900.000	350	2.571	$30.000
Greenfield	600.000	400	1.500	$28.000
Sunnyside	750.000	500	1.500	$32.000
Frostville	300.000	200	1.500	$27.000
Eastbank	1.100.000	250	4.400	$36.000
Westwood	2.300.000	450	5.111	$42.000
Clearwater	500.000	180	2.777	$29.000
Ironforge	800.000	190	4.211	$40.000
Lakeview	950.000	220	4.318	$47.000
Cedarwood	410.000	300	1.367	$25.000
Newhaven	1.500.000	280	5.357	$70.000
Pinecrest	670.000	330	2.030	$33.000
Stonebridge	390.000	210	1.857	$11.000

(casi) Todas las técnicas de ML usan el mismo esquema de procesamiento



Técnicas de ML trabajan sobre datos multidimensionales

- Cada dato esta caracterizado por una serie de características = *features* = atributos = variables.
- Siempre buscaremos predecir algo a partir de estas características.
- Muchas veces, los datos poseen además una etiqueta = *label* = rótulo, que típicamente buscaremos predecir.

Datos de Entrenamiento

Características			Etiqueta
Población	Superficie (km^2)	Densidad poblacional por km^2	Ingreso anual promedio per cápita en dólares
3.500.000	600	5.833	$45.000
1.200.000	150	8.000	$35.000
2.800.000	300	9.333	$40.000
900.000	350	2.571	$30.000
600.000	400	1.500	$28.000
750.000	500	1.500	$32.000
300.000	200	1.500	$27.000
1.100.000	250	4.400	$36.000

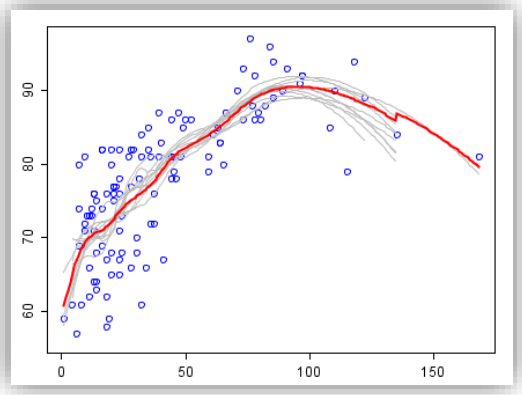
Técnicas de ML trabajan sobre datos multidimensionales

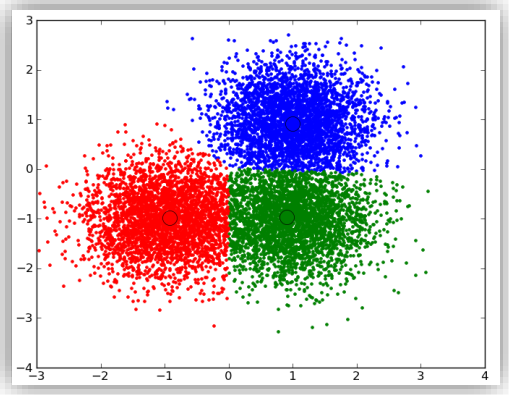
- Cada dato del conjunto de entrenamiento puede considerarse como un vector.
- La cantidad de características define la dimensionalidad del vector.
- El espacio donde viven los vectores se conoce como espacio de características.

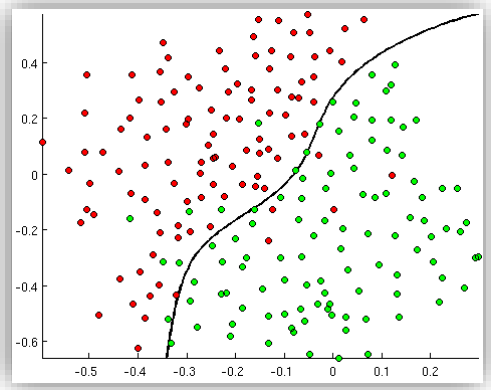
Datos de Entrenamiento

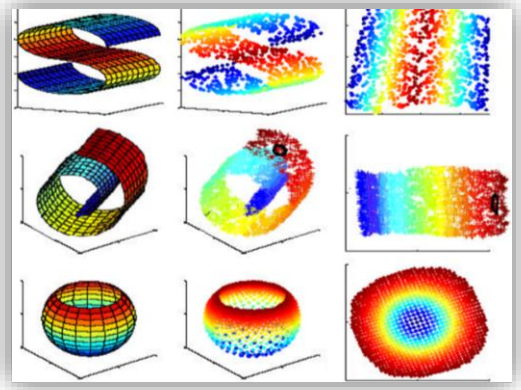
Características			Etiqueta
Población	Superficie (km^2)	Densidad poblacional por km^2	Ingreso anual promedio per cápita en dólares
3.500.000	600	5.833	$45.000
1.200.000	150	8.000	$35.000
2.800.000	300	9.333	$40.000
900.000	350	2.571	$30.000
600.000	400	1.500	$28.000
750.000	500	1.500	$32.000
300.000	200	1.500	$27.000
1.100.000	250	4.400	$36.000

Cada dato puede verse como un vector/punto en el espacio de características









Esto hace que datos y tareas puedan interpretarse de manera más intuitiva desde un punto de vista geométrico

Teniendo las bases cubiertas, podemos revisar como usar *scikit-learn para resolver problemas predictivos*

Scikit-learn será nuestra herramienta principal para este capítulo

- scikit-learn es el módulo para ML más conocido y utilizado en Python.
- Su principal atractivo es una interfaz limpia, uniforme y simple, que facilita la exploración y permite la integración con otros paquetes, como Pandas.
- Posee además de una completa documentación en línea (https://scikit-learn.org/).

Esquema de datos de scikit-learn es similar a Pandas

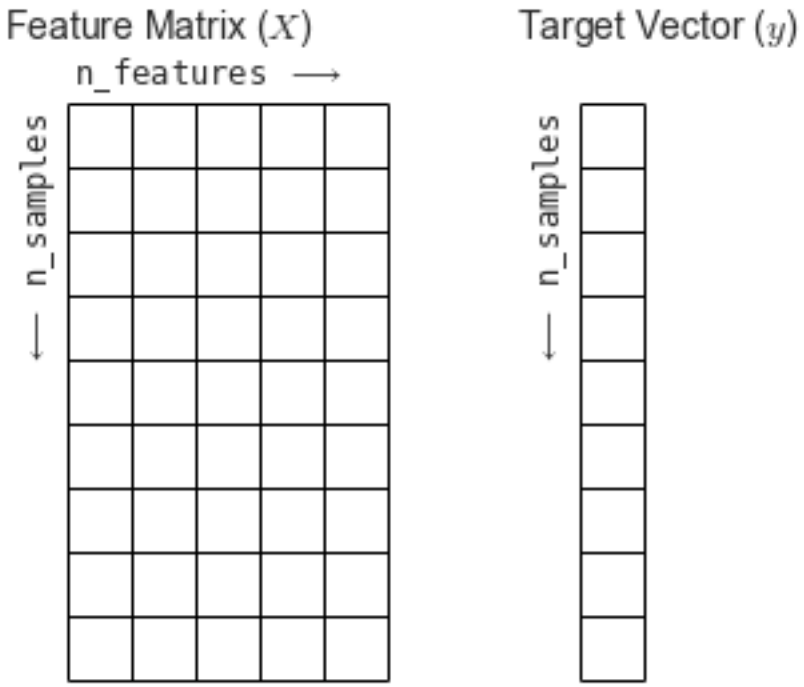
- Datos son representados por una matriz de *características* y, si la tarea a resolver es supervisada, un vector objetivo de etiquetas.

- Las características de los ejemplos se almacenan en una matriz de característcas (X), de tamaño [n_samples, n_features] (esta matriz típicamente será un DataFrame, pero puede ser de otro tipo).

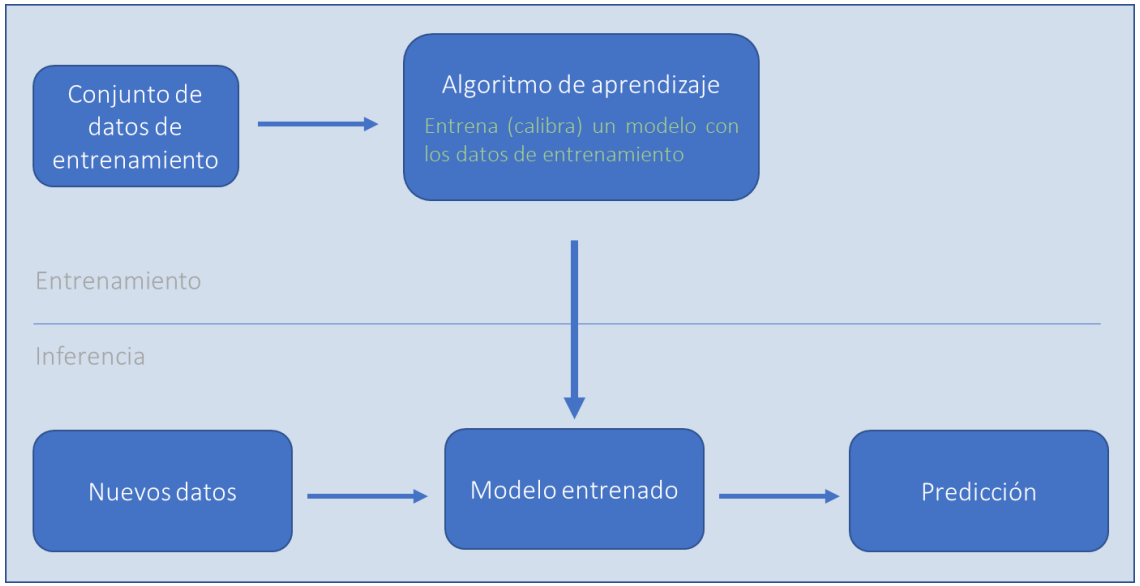
- El vector objetivo (y) contiene la etiqueta para cada ejemplo y tiene tamaño [n_samples, 1] (este vector típicamente será un Series, pero puede ser de otro tipo).

- Y eso es todo…

Esquema de datos es similar a Pandas



Población	Superficie (km²)	Densidad poblacional por km²	Ingreso anual promedio per cápita en dólares
3.500.000	600	5.833	$45.000
1.200.000	150	8.000	$35.000
2.800.000	300	9.333	$40.000
900.000	350	2.571	$30.000
600.000	400	1.500	$28.000
750.000	500	1.500	$32.000
300.000	200	1.500	$27.000
1.100.000	250	4.400	$36.000



En base a los datos y al funcionamiento de ML, ¿cómo podríamos plantear el problema de predicción usando scikit-learn?

Uso de los modelos es simple y directo

En general, un caso de uso típico en scikit-learn es como el siguiente:

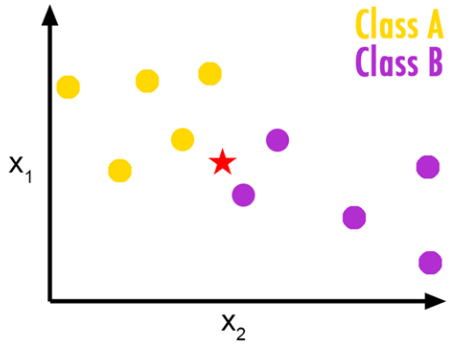
1. Obtener o generar matriz X y vector y → Preparar los datos
2. Elegir el modelo adecuado, importando la clase correspondiente desde *sklearn.*
3. Entrenar el modelo llamando al fit(X, y).
4. Aplicar el modelo entrenado en nuevos datos, usando el método predict().

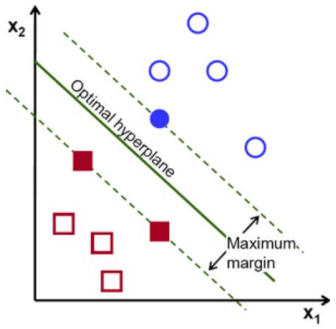
Preparación de los datos tiene un foco distinto

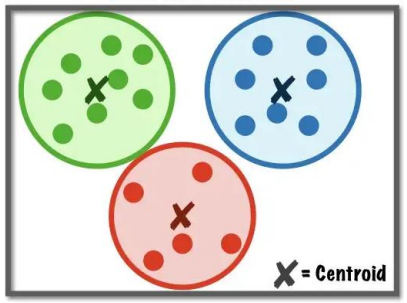
- El objetivo es que los datos de entrenamiento sean lo más representativos posibles del problema a resolver.
- Así, queremos incorporar la menor cantidad posible de supuestos sobre ellos, de modo que el modelo aprenda de los datos y no de nosotros.
- ¿Qué implica esto en la práctica? Que son más importantes las cosas que están que las que no lo están o que presentan fallas.
- Esto último no implica que vamos a descartar los datos faltantes o outliers, sino que pueden ser corregidos posteriormente mediante modelos entrenados sobre los datos “correctos”.
- Para asegurar esto, es necesario seguir un protocolo bien definido que considera etapas de codificación, división de conjuntos y transformación, entre otros (ver la materia).

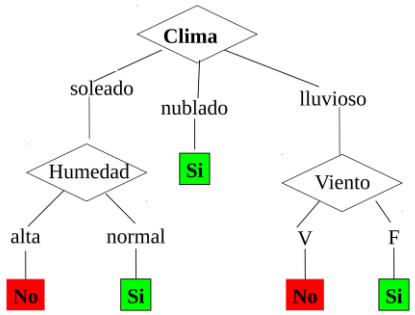
Una vez listos los datos, deberán elegir entre múltiples modelos predictivos, dependiendo de la tarea a resolver

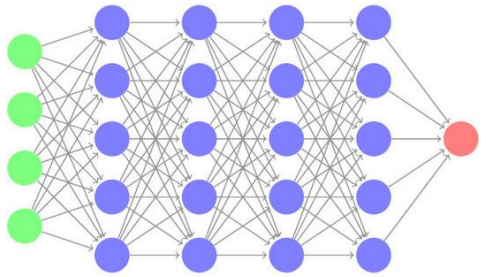
- k-NN
- Regresiones SVM
- Árboles de decisión
- Ensambles
- Redes neuronales
- K-Means Clustering
- PCA, tSNE
- y más…

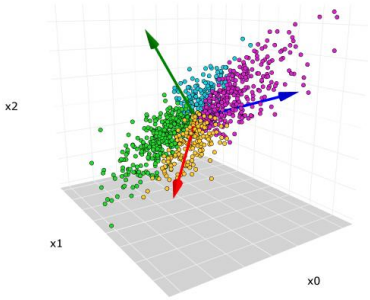








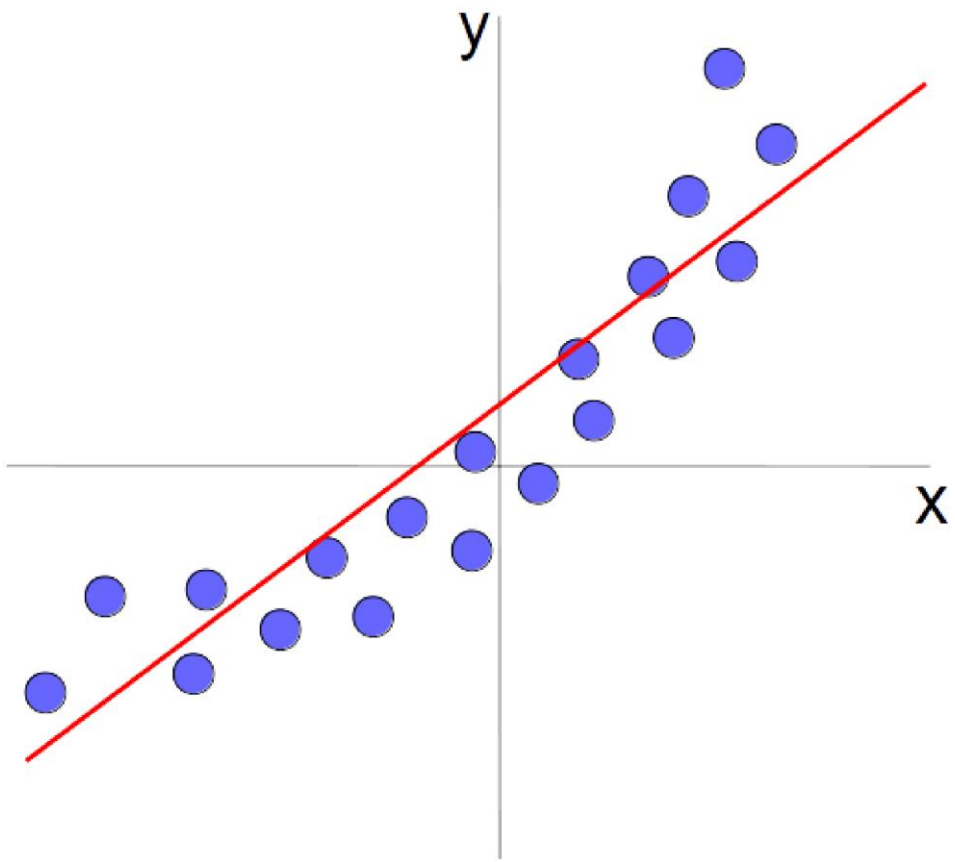




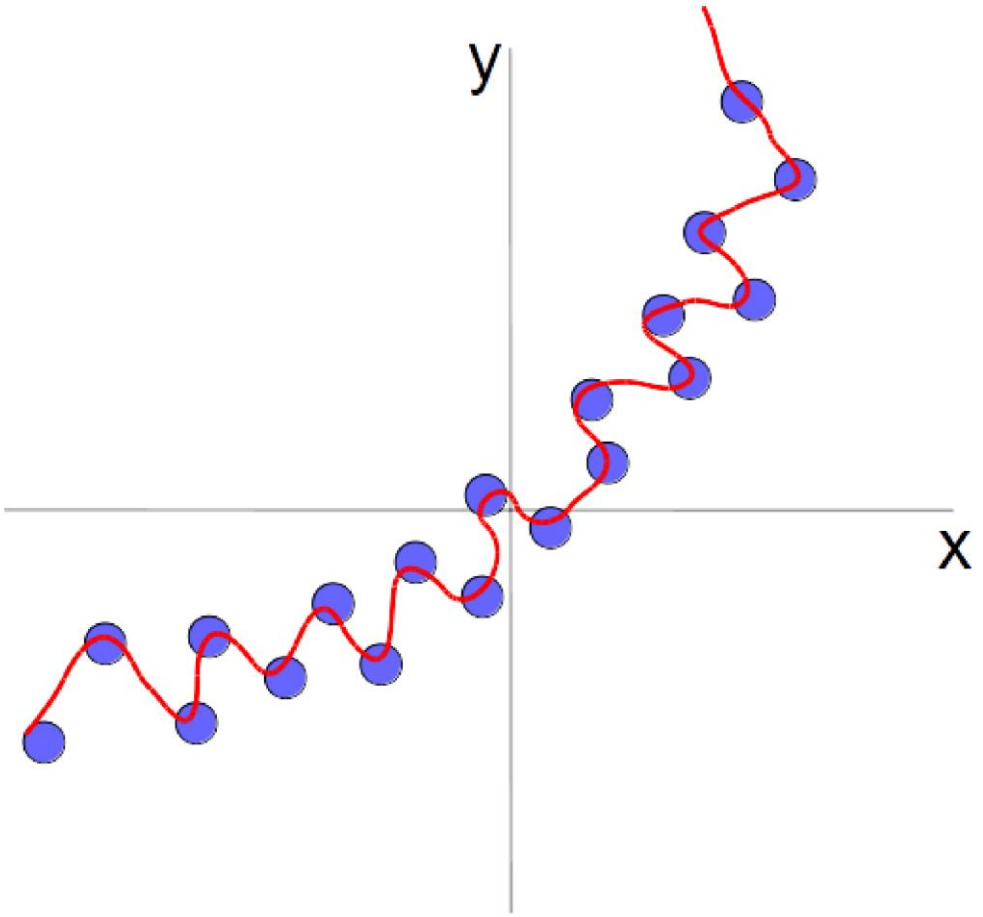
A pesar de la simplicidad en el uso de sklearn, el procedimiento para elegir el modelo adecuado no es trivial

- En general, los algoritmos de aprendizaje viven y mueren por los datos de entrenamiento.
- Lamentablemente, tener un buen conjunto de entrenamiento no asegura tener buen rendimiento con datos no vistos previamente (buena generalización).
- La complejidad del modelo pasa a ser un tema central.
- Típicamente la complejidad se asocia con la cantidad de factores que pueden considerar el modelo, o cuánto puede aprender.

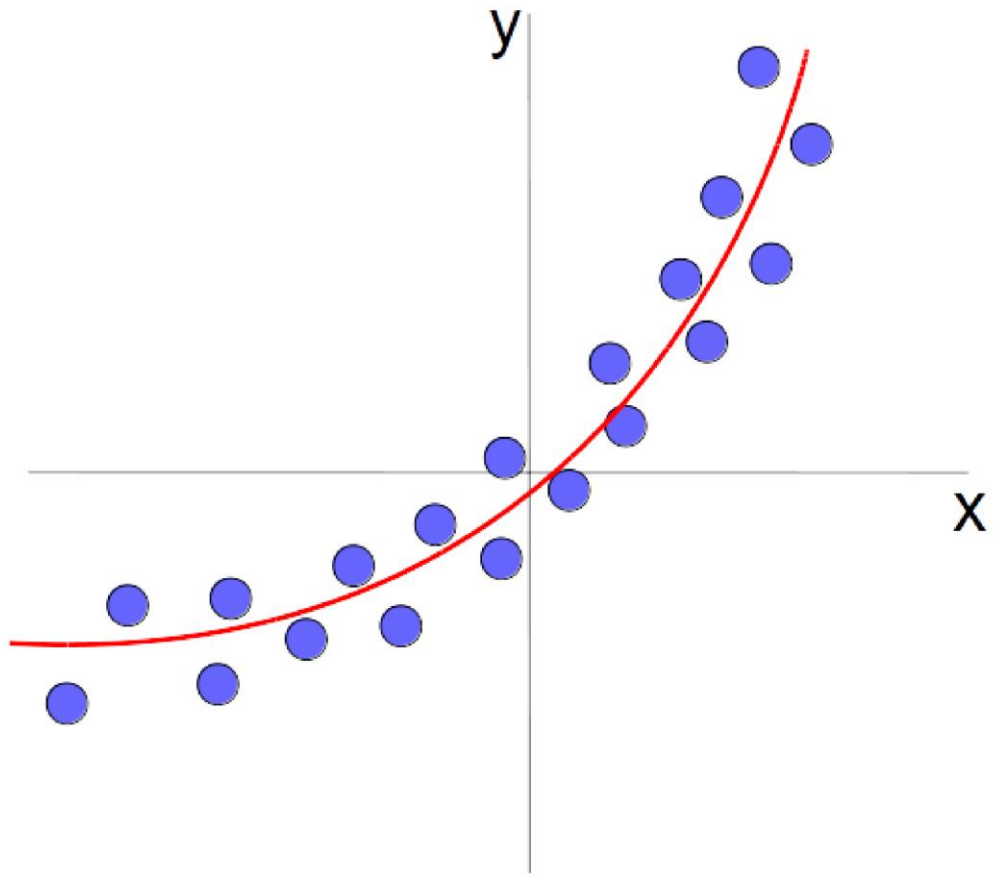
Subajuste (*underfitting*) → modelo menos complejo que los datos



Sobreajuste (overfitting) → modelo más complejo que los datos



Ajuste correcto del modelo

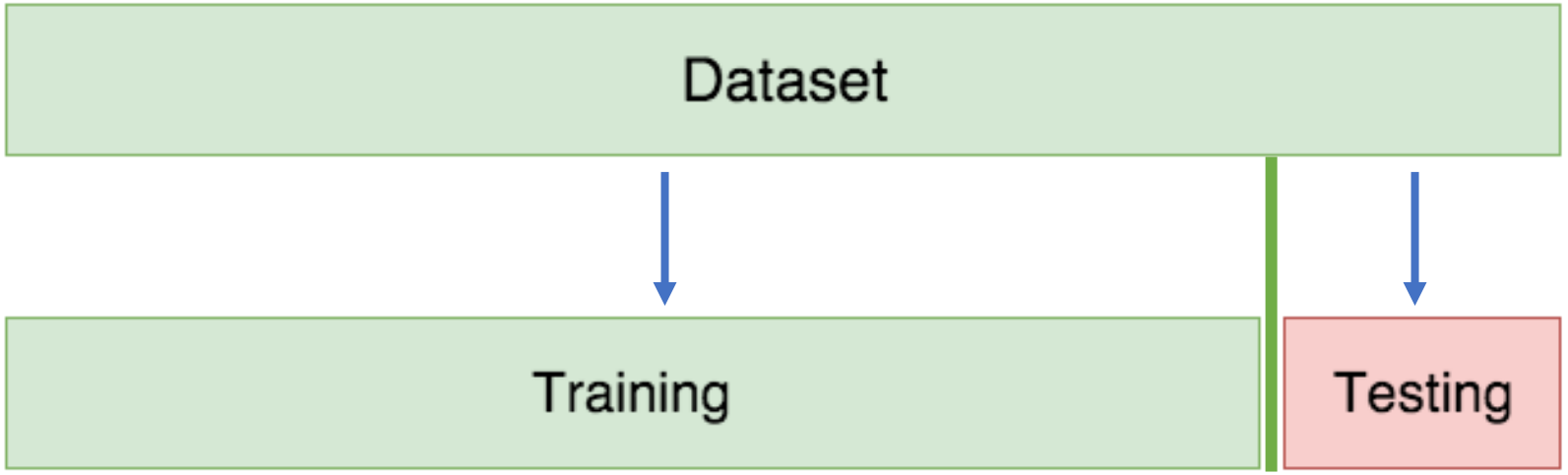


¿Cómo puedo entonces elegir el modelo adecuado?

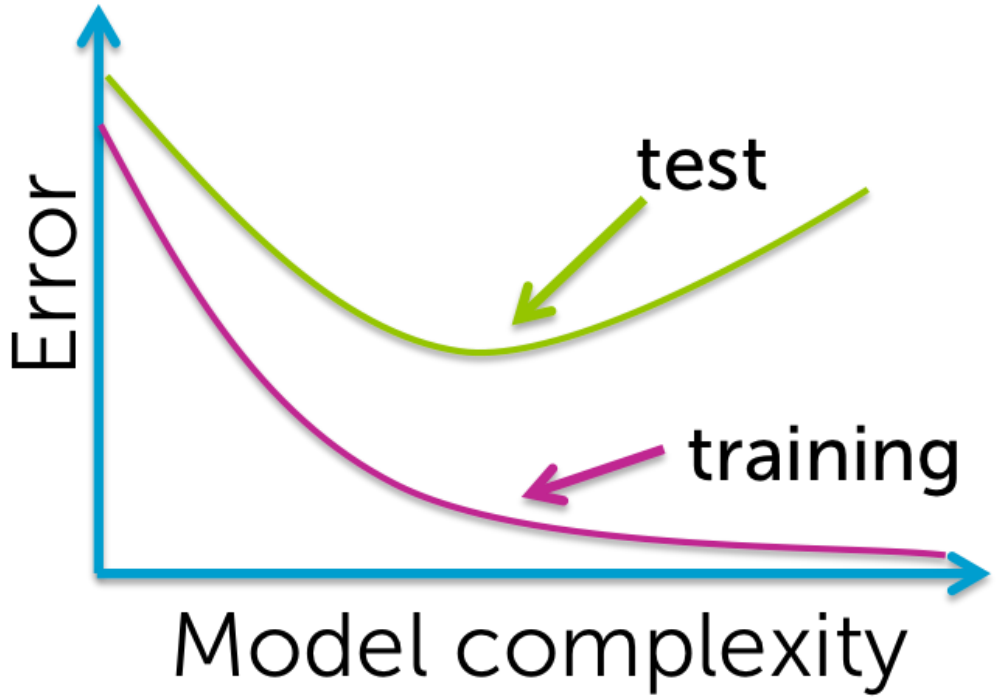
- Lo primero que necesitamos es una manera de medir el rendimiento de un modelo.
- Para esto, típicamente debemos definir dos cosas: una métrica de rendimiento y un conjunto de prueba.
- La métrica de rendimiento estará dada generalmente por la tarea a resolver.
- El conjunto de pruebas debe venir de la misma distribución de datos que el conjunto de entrenamiento (puede ser un subconjunto de este).
- Etiquetas del conjunto de prueba deben ser completamente desconocidas para el modelo (solo se conocen al evaluar el rendimiento).

	Población	Superficie (km^2)	Densidad poblacional por km^2	Ingreso anual promedio per cápita en dólares
Entrenamiento	3.500.000	600	5.833	$45.000
	1.200.000	150	8.000	$35.000
	2.800.000	300	9.333	$40.000
	900.000	350	2.571	$30.000
	600.000	400	1.500	$28.000
	750.000	500	1.500	$32.000
	300.000	200	1.500	$27.000
	1.100.000	250	4.400	$36.000
	2.300.000	450	5.111	$42.000
	500.000	180	2.777	$29.000
Prueba/Test	800.000	190	4.211	?
	950.000	220	4.318	
	410.000	300	1.367	
	1.500.000	280	5.357	
	670.000	330	2.030	
	390.000	210	1.857	

Una forma clara de ver esto es con conjuntos de datos disjuntos



Rendimiento está íntimamente relacionado con la complejidad de los modelos



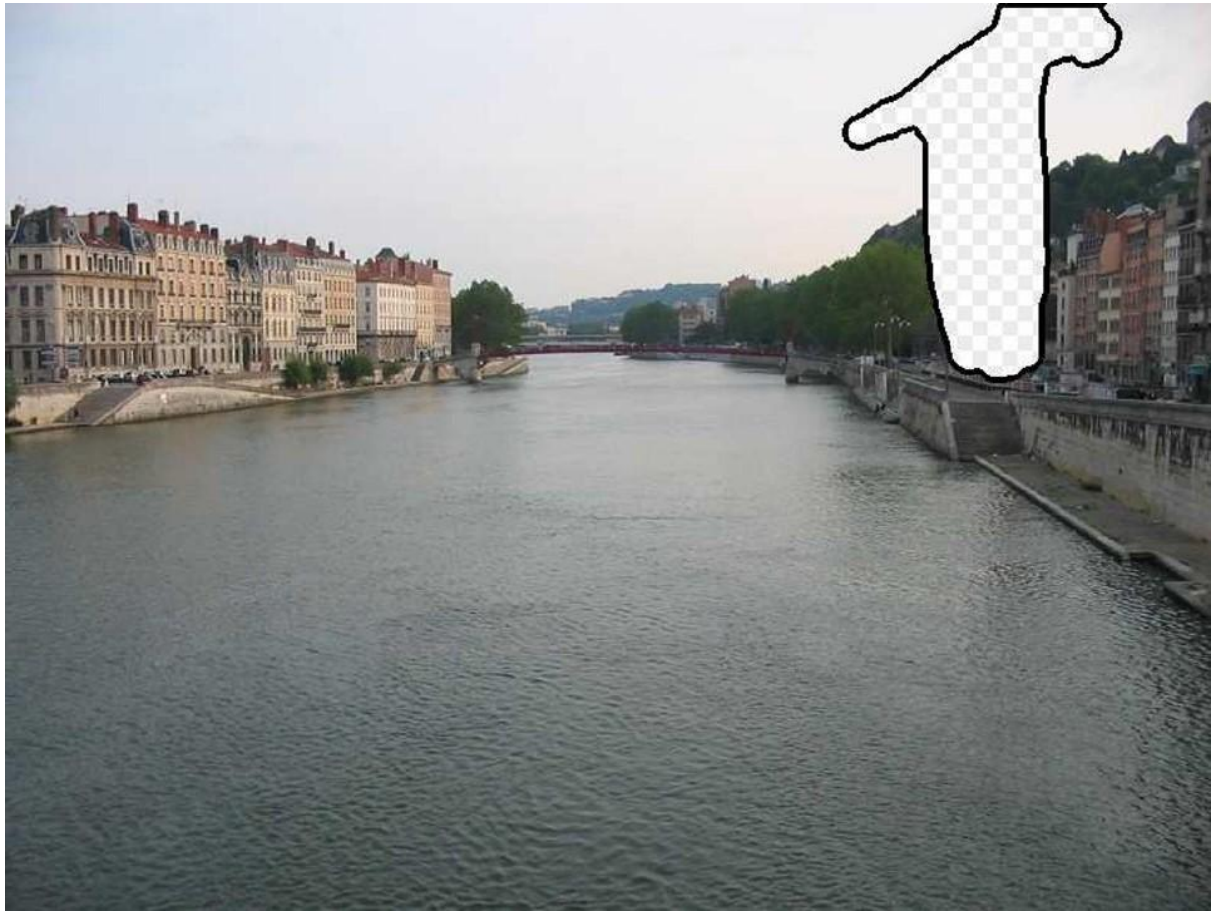
Típicamente, un modelo sobreajustado tendrá un rendimiento mucho mejor en el conjunto de entrenamiento que en el de test

scikit-learn también facilita el proceso de selección de modelos

- scikit-learn posee métodos explícitos para divider los datos en conjuntos de entrenamiento y prueba.
- También es posible utilizar esquemas más robustos de evaluación, como la validación cruzada.
- Hay una gran cantidad de métricas de rendimiento para distintos tipos de problema ya implementadas.
- Se encuentran en el módulo `sklearn.metrics`
- En la práctica, las más usadas son *accuracy*, *precision*, *recall*, error cuadrático medio y matriz de confusión.

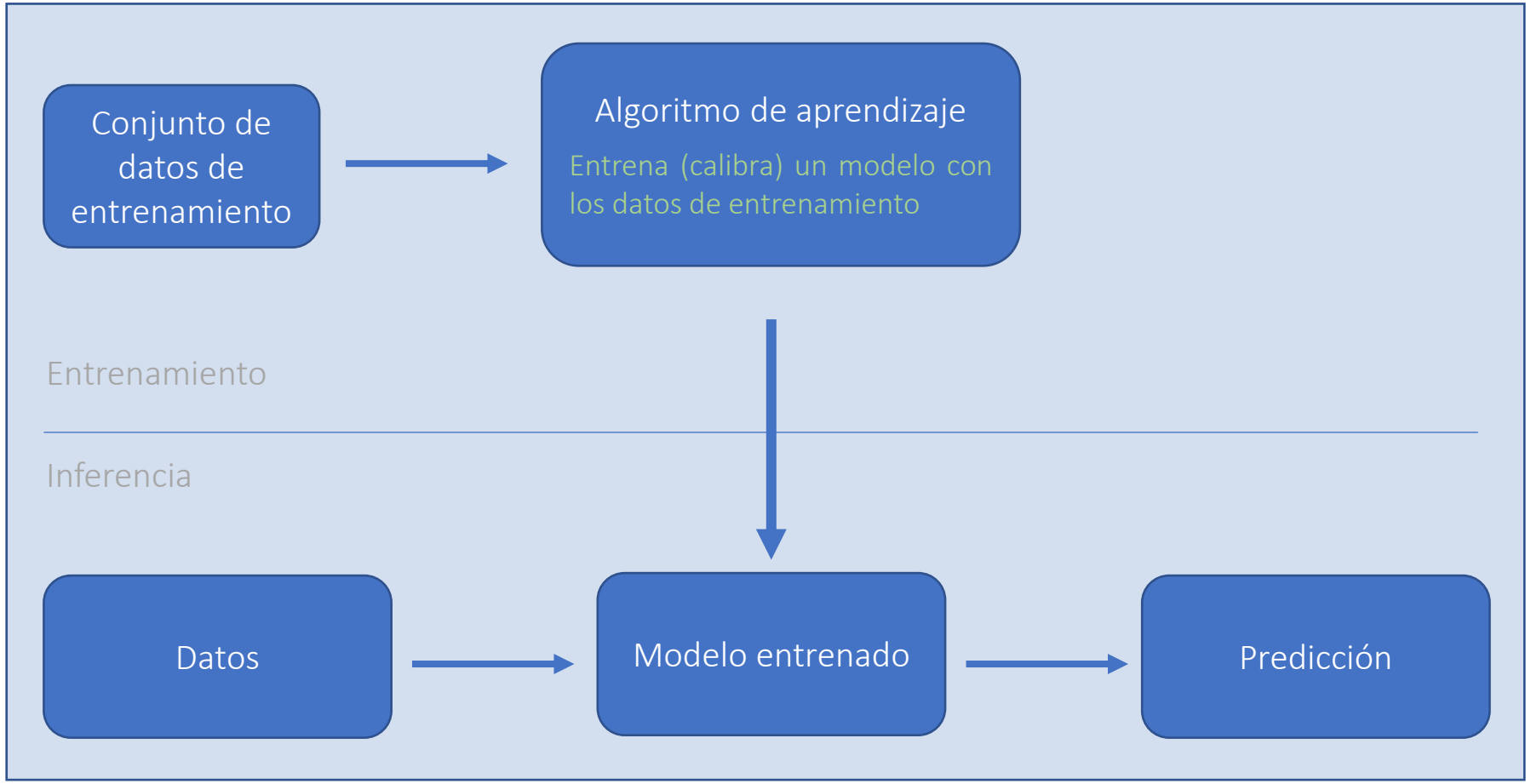
Cerremos con un caso real de estudio

Cerremos con un caso real de estudio



Cerremos con un caso real de estudio

Recordemos que (casi) todas las técnicas de ML usan el mismo esquema de procesamiento

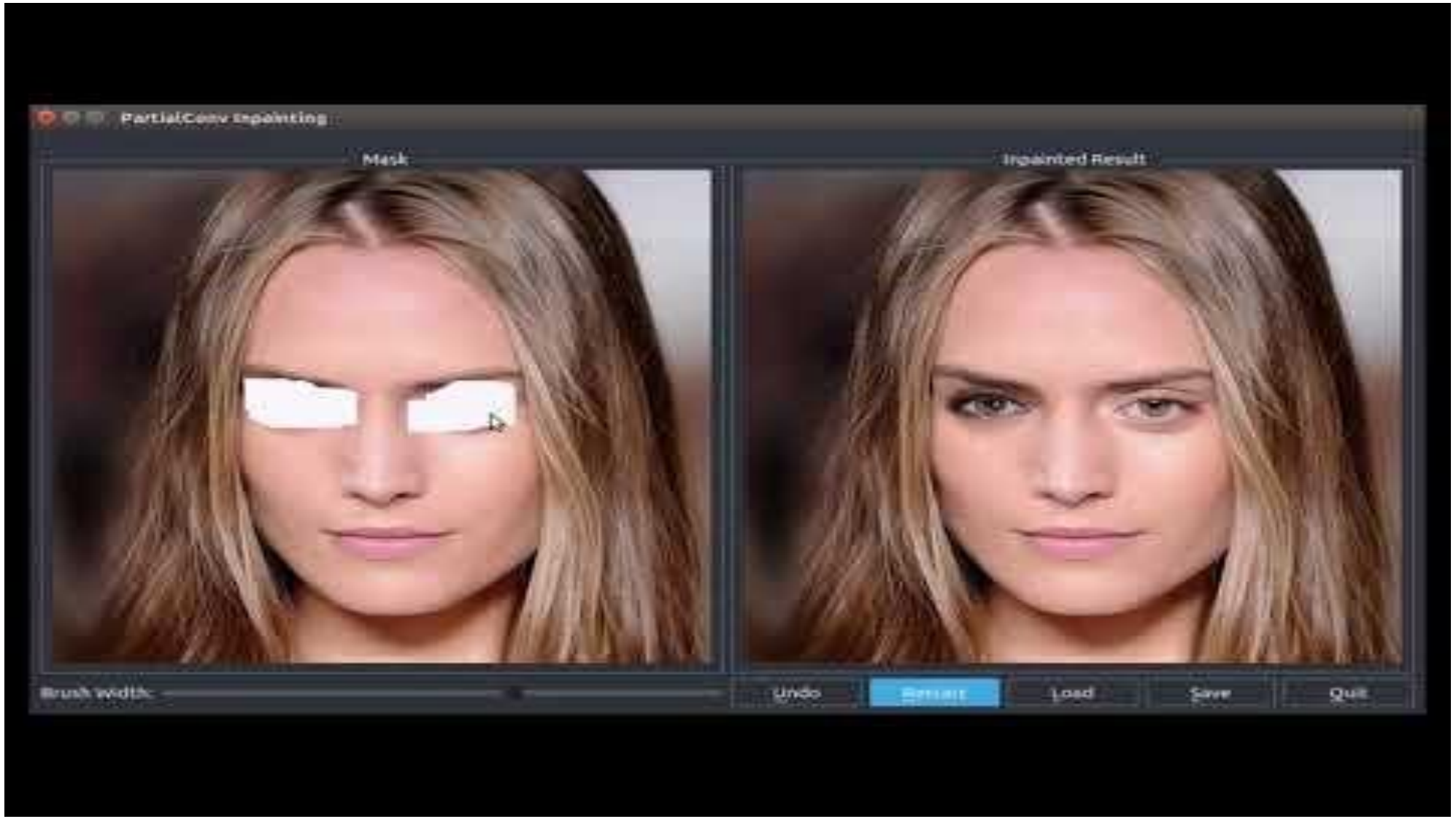


Datos de entrenamiento se generan a partir de imágenes reales

Algoritmo de aprendizaje supervisado

¿  =  ?

Veamos cómo funciona el sistema en la práctica



(https://youtu.be/gg0F5JjKmhA)

En resumen…

- ML se centra en algoritmos/modelos que aprenden de los datos para resolver una tarea
- scikit-learn permite hacer ML en Python de manera práctica y rápida
- (casi) Todas las técnicas de ML funcionan de la misma manera
- Qué técnica usar dependerá de la tarea y los datos disponibles, pero siempre se debe decidir en base a métricas de rendimiento
- Todo esto y mucho más en el curso ICT3115 – Sistemas Urbanos Inteligentes e IIC3697 – Aprendizaje Profundo (ambos dictados los primeros semestres)

Algunos aspectos relevantes para la resolución de problemas

- Preparación de los datos cambia de foco: son más importantes las cosas que están que las que no lo están o que presentan fallas → datos faltantes y outliers pueden corregirse con los mismos modelos.

- Revisar cuidadosamente la materia para cumplir con el orden correcto de pasos durante la preparación de los datos: codificación, división, transformación, etc.

- Correctitud y rigurosidad del protocolo de entrenamiento y evaluación es lo central para la elección del modelo final.

- Rendimiento en bruto no es lo más importante en esta parte inicial.

Cómo sigue la sesión

- Para este capítulo tendremos múltiples ejercicios, divididos en las dos semanas de preparación disponibles para el laboratorio.
- Esta primera semana, los ejercicios tendrán un esquema más dirigido, con el fin de facilitar la comprensión y práctica de los conceptos.
- NO UTILICEN LOS ASISTENTES PARA ANÁLISIS EXTENSOS, SUBDIVIDAN EL PROBLEMA PARA ASEGURAR CORRECTITUD.
- No olviden responder el ticket de salida, siempre considerando que lo indicado en él debe verse reflejado en el repositorio privado.
- A las 17:30 se cerrará el ticket de manera definitiva.

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2115 - Programación como herramienta para la ingeniería

Modelos Predictivos basados en Machine Learning

Profesor: Hans Löbel