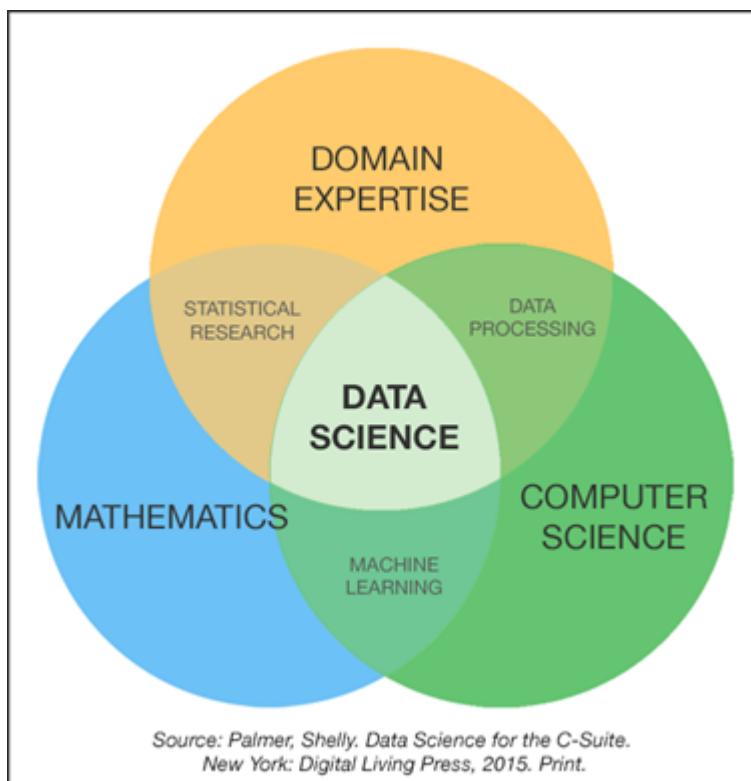


# CIENCIA DE DATOS: APRENDE LOS FUNDAMENTOS DE MANERA PRÁCTICA



## SESION 01 INTRODUCCION AL MACHINE LEARNING

**Juan Antonio Chipoco Vidal**  
[jchipoco@gmail.com](mailto:jchipoco@gmail.com)



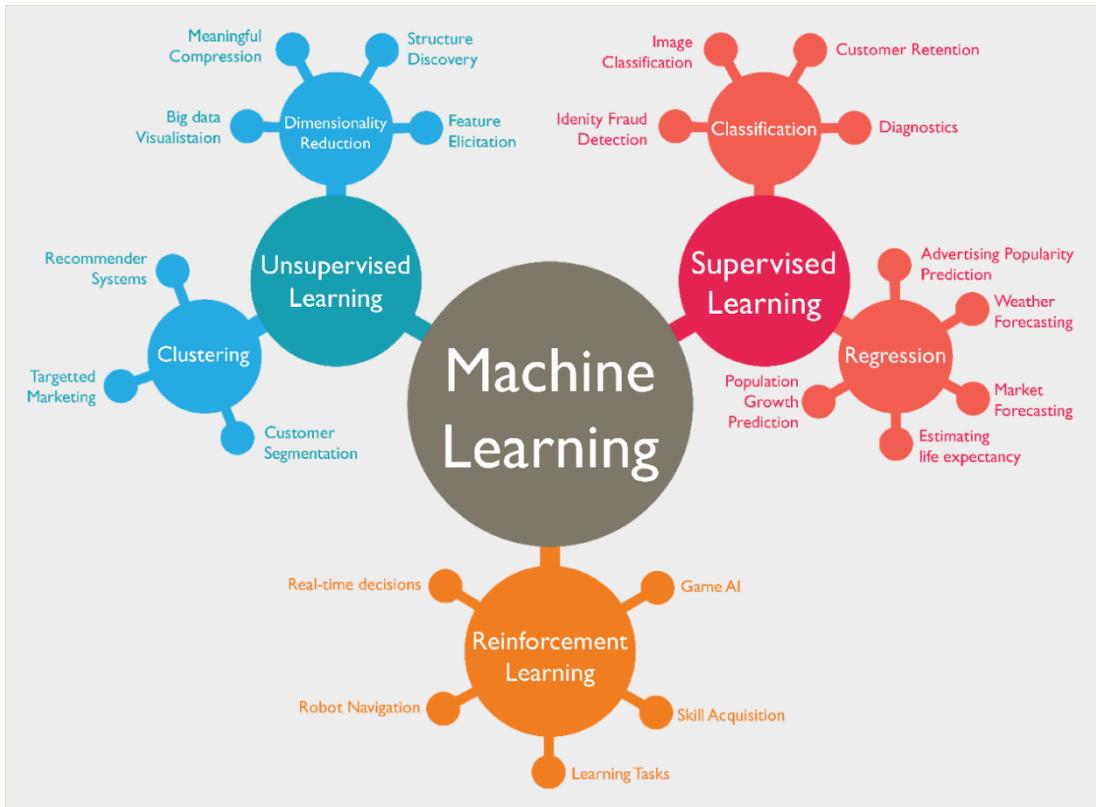
# ÍNDICE

<b>OBJETIVO .....</b>	<b>7</b>
<b>INTRODUCCIÓN – PARTE 1 .....</b>	<b>8</b>
<b>INTRODUCCIÓN – PARTE 2 .....</b>	<b>9</b>
<b>¿QUÉ ES CIENCIA DE DATOS?.....</b>	<b>10</b>
<b>DATA SCIENCE: DATA LABELING .....</b>	<b>11</b>
<b>DATA SCIENCE – TIPOS DE DATOS .....</b>	<b>12</b>
DATOS CATEGÓRICOS .....	12
DATOS NUMÉRICOS .....	13
<b>CICLO DE VIDA DE LA CIENCIA DE DATOS.....</b>	<b>14</b>
1.- ENTENDER EL NEGOCIO .....	14
2.- RECOLECCIÓN DE DATOS.....	14
3.- LIMPIEZA DE DATOS.....	15
3.- ANÁLISIS DE DATOS .....	15
4.- MODELAMIENTO DE DATOS, MODELAMIENTO MACHINE LEARNING.....	15
5.- EVALUACIÓN DEL MODELO .....	15
6.- VISUALIZACIÓN Y REPORTES.....	15
7.- DESPLIEGUE DE MODELO .....	16
<b>MACHINE LEARNING .....</b>	<b>17</b>
¿QUÉ ES EL MACHINE LEARNING? .....	17
ESTADÍSTICA Y MACHINE LEARNING .....	18
CONSTRUCCIÓN DEL MODELO .....	24
APRENDIZAJE SUPERVISADO .....	25
<i>Algunos algoritmos supervisados .....</i>	26
<i>Ejemplo .....</i>	26
APRENDIZAJE NO SUPERVISADO .....	27

Algunos algoritmos no supervisados.....	28
Ejemplo .....	28
APRENDIZAJE REFORZADO .....	29
<i>El complejo juego Go.</i> .....	30
ALGORITMOS DE CLASIFICACIÓN.....	31
CLASIFICADOR BINARIO .....	31
CLASIFICADOR MULTICLASE .....	31
ALGORITMOS DE REGRESIÓN .....	32
DATA SCIENCE LIFE CYCLE .....	34
ANÁLISIS EXPLORATORIO DE DATOS .....	34
FEATURE ENGINEERING .....	35
MODEL BUILDING .....	38
UNDERFITTING .....	40
OVERFITTING .....	42
TRAIN, VALIDATION Y TEST SETS .....	43
<i>Conjunto de entrenamiento.</i> .....	43
<i>Conjunto de validación.</i> .....	43
<i>Conjunto de prueba.</i> .....	43
ENTER VALIDATION .....	44
BIAS-VARIANCE TRADEOFF .....	46
<i>Bias</i> .....	46
<i>El sesgo se refiere a las suposiciones erróneas del modelo generado acerca de los datos. Un sesgo alto o underfitting (ajuste insuficiente) significa que el modelo no puede capturar la tendencia o el patrón en los datos. Por lo general, se produce cuando la función de hipótesis es demasiado simple o tiene muy pocos features.</i> .....	46
<i>El modelo con alto sesgo no aprende bien de los datos de entrenamiento y simplifica demasiado el modelo. Tiene un desempeño deficiente en el conjunto de entrenamiento y prueba porque no puede identificar patrones en los datos.</i> .....	46
<i>Variance</i> .....	46

<b>MACHINE LEARNING .....</b>	<b>48</b>
EVALUACIÓN DEL MODELO.....	48
FUNCIÓN DE PERDIDA Y FUNCIÓN DE COSTO .....	50
EVALUACIÓN DEL MODELO.....	51
MÉTRICAS DE PERFORMANCE PARA REGRESION (FUNCIONES DE PERDIDA) .....	52
MÉTRICAS DE PERFORMANCE PARA CLASIFICACION .....	57
<b>ANEXOS.....</b>	<b>58</b>
MATRIZ DE CONFUSIÓN .....	58
INSTALACIÓN DE ANACONDA EN WINDOWS.....	61
USANDO GOOGLE COLAB .....	67

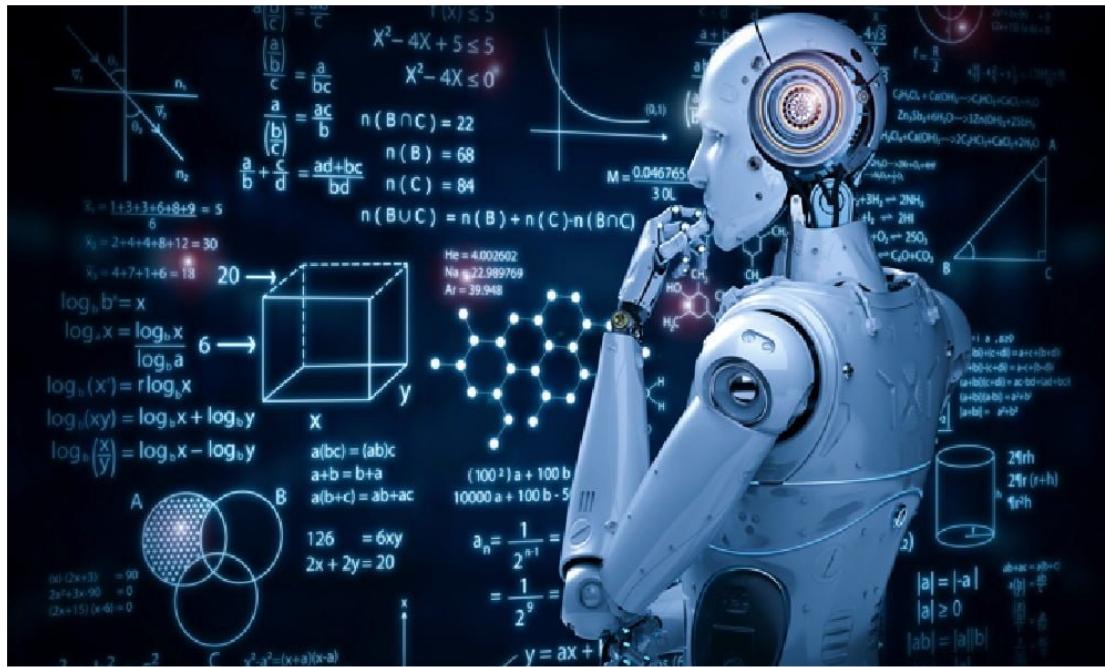
## OBJETIVO



El objetivo de esta semana es conocer la terminología, principales algoritmos y áreas del Machine Learning, así como las herramientas que nos permitirán trabajar en nuestras prácticas semanales del curso.

En esta primera sesión la práctica de laboratorio consistirá en conocer las interfaces gráficas que nos facilitaran el procesamiento y visualización de nuestros resultados, así como las principales librerías de Python para nuestro curso, como son: numpy, pandas, scikit learn, matplotlib.

## INTRODUCCION – parte 1

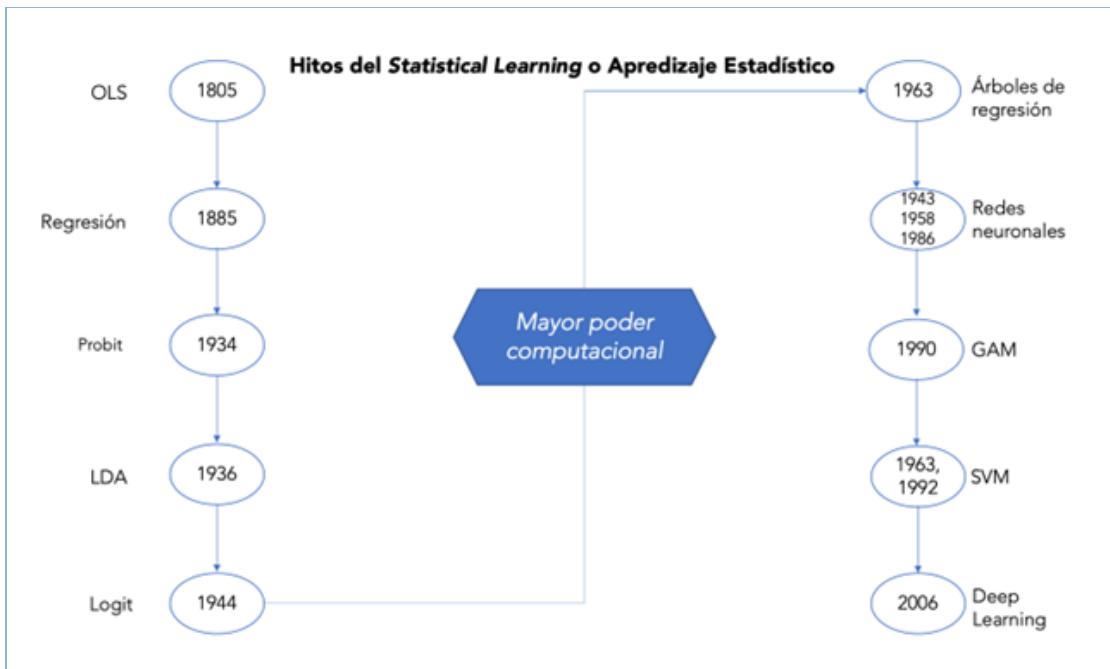


Uno de los vocablos que más se repiten en tecnología en los últimos tiempos es machine learning, o aprendizaje automático, un término que está íntimamente relacionado con la inteligencia artificial.

Brevemente se podría definir machine learning como el aprendizaje automático de los sistemas tecnológicos mediante algoritmos con el objetivo de que puedan llegar a realizar diversas acciones por su cuenta.

Esto que parece propio de la ciencia ficción o incluso de películas apocalípticas donde las máquinas se rebelan (véase Terminator) ya es una realidad, aunque no tan oscura como esos ejemplos cinematográficos. Es más, machine learning es una excelente noticia para mejorar procesos e impedir que las personas tengan que perder un tiempo muy valioso en realizar ciertas tareas. A fin de cuentas, que los sistemas sean capaces de aprender a partir de los datos que obtienen supervisados o sin supervisar por seres humanos supone una evolución clave para el desarrollo tecnológico durante los próximos años y décadas.

## INTRODUCCIÓN – parte 2

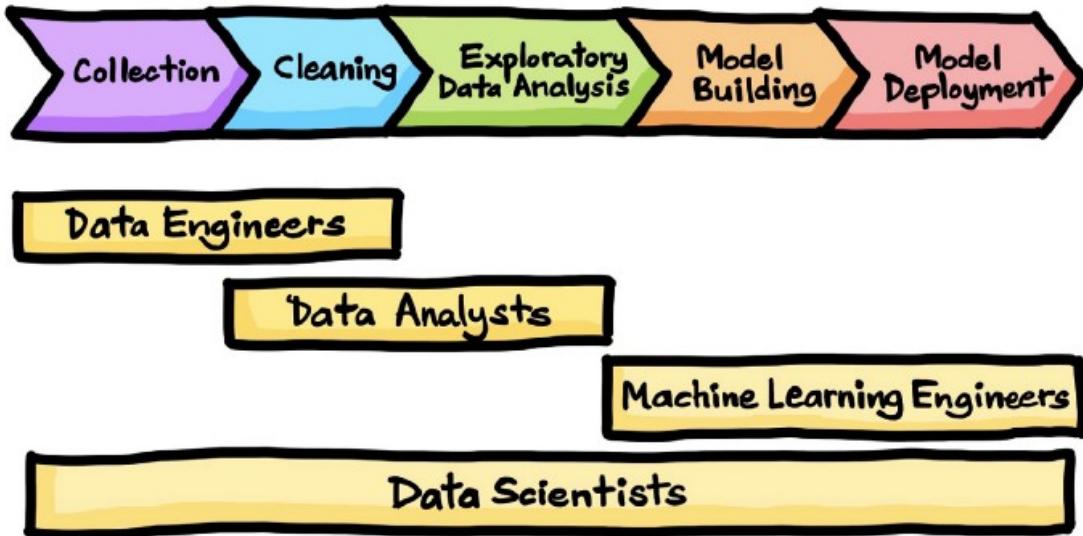


La ciencia de datos es un campo de estudio que tiene como objetivo utilizar un enfoque científico para extraer significado e información de los datos.

El aprendizaje automático, por otro lado, se refiere a un grupo de técnicas utilizadas por los científicos de datos que permiten que las computadoras aprendan de los datos.

La ciencia de datos y el aprendizaje automático son palabras muy populares en la actualidad. Estos dos términos a menudo se juntan, pero no deben confundirse con sinónimos. Aunque la ciencia de datos utiliza el aprendizaje automático, estos son campos amplios con muchas herramientas diferentes.

## ¿QUÉ ES CIENCIA DE DATOS?

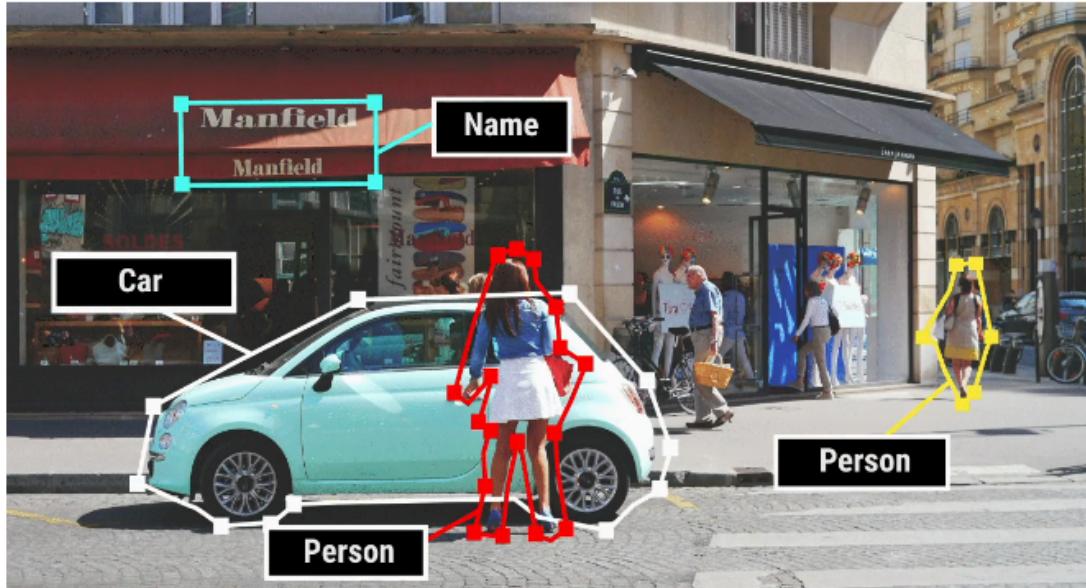


La ciencia de datos es el campo de estudio que combina la experiencia en el dominio, las habilidades de programación y el conocimiento de las matemáticas y las estadísticas para extraer información significativa de los datos.

Los profesionales de la ciencia de datos aplican algoritmos de aprendizaje automático a números, texto, imágenes, video, audio y más para producir sistemas de inteligencia artificial (AI) con el objetivo de realizar tareas que normalmente requieren inteligencia humana.

A su vez, estos sistemas generan conocimientos que los analistas y usuarios comerciales pueden traducir en valor comercial tangible.

## DATA SCIENCE: data labeling



Como sugiere el nombre, los datos etiquetados (data labels) son datos sin procesar (raw data) que hemos recopilado a los cuales les hemos agregado descripciones significativas o les hemos asignado una clase. También se les conoce como datos anotados.

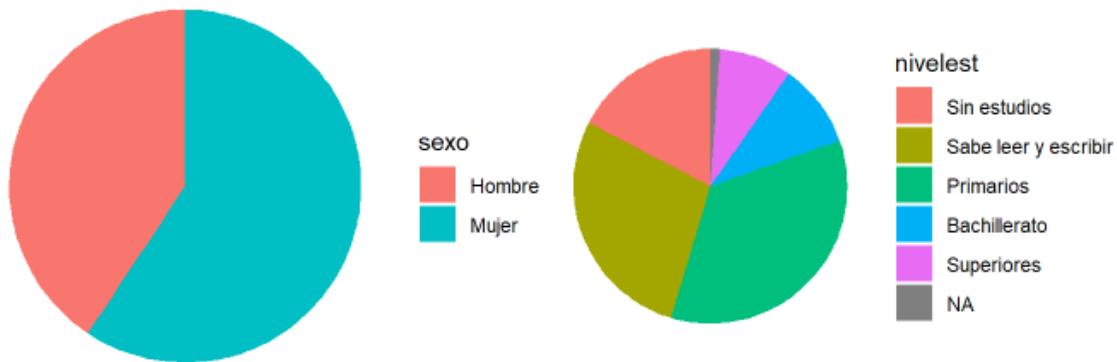
¿Qué es una etiqueta en el aprendizaje automático? Supongamos que estamos construyendo un sistema de reconocimiento de imágenes y ya hemos recopilado varios miles de fotografías. Tal como vemos en la imagen superior las etiquetas le estarían diciendo a la IA que las fotos contienen una 'persona', un 'árbol', un 'automóvil', etc.

Las funciones y etiquetas de aprendizaje automático son asignadas por expertos humanos, y el nivel de experiencia necesario puede variar. En el ejemplo anterior, no necesita personal altamente especializado para etiquetar las fotos. Sin embargo, si tiene, por ejemplo, un conjunto de radiografías y necesita entrenar la IA para buscar tumores, es probable que necesite médicos para trabajar como anotadores de datos. Naturalmente, debido a los recursos humanos necesarios, la fase del etiquetado manual de datos es mucho más costoso que la fase de recopilación de datos los cuales por lo general se encuentran sin etiquetar.

## DATA SCIENCE – tipos de datos

Cuando recopilamos datos para una investigación, es importante conocer la forma de sus datos para poder interpretarlos y analizarlos de manera efectiva. Existen principalmente dos tipos de datos: datos categóricos y datos numéricos.

### Datos categóricos



Los datos categóricos se refieren a un tipo de dato que se pueden almacenar e identificar en función de los nombres o etiquetas que se les asigna. Se realiza un proceso llamado coincidencia, para extraer las similitudes o relaciones entre los datos y luego se agrupan en consecuencia.

Los datos recopilados en forma categórica también se conocen como datos cualitativos. Cada conjunto de datos se puede agrupar y etiquetar según sus cualidades coincidentes, en una sola categoría. Esto hace que las categorías sean mutuamente excluyentes.

Hay dos subtipos de datos categóricos, a saber: datos nominales y datos ordinales.

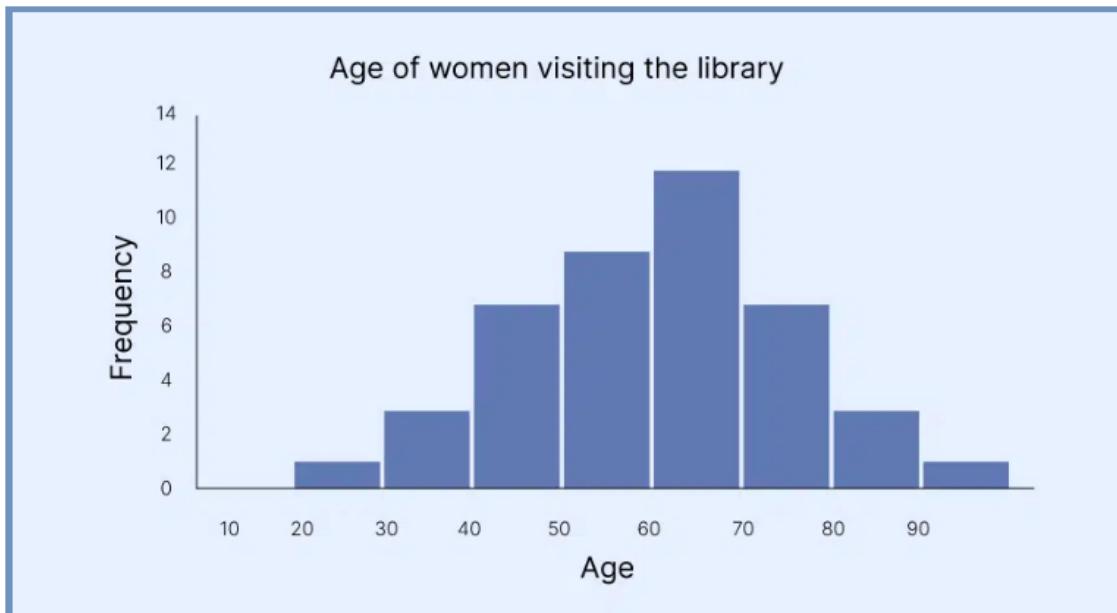
- **Datos nominales**

También se denominan datos de nombres. Este es un tipo que nombra o etiqueta los datos y sus características son similares a un sustantivo. Ejemplo: nombre de la persona, género, nombre de la escuela.

- **Datos ordinales**

Esto incluye datos o elementos de datos que se clasifican, ordenan o utilizan en una escala de calificación. Puedes contar y ordenar datos ordinales, pero no te permite medirlos. Ejemplo: Calificar el resultado de un seminario entre 1 y 5.

## Datos numéricos



Los datos numéricos se refieren a los datos que están en forma de números, y no en ningún idioma o forma descriptiva. A menudo denominados datos cuantitativos, los datos numéricos se recopilan en forma de números y se diferencian de cualquier forma de tipos de datos numéricos debido a su capacidad para calcularse estadística y aritméticamente.

También tiene dos subtipos conocidos como datos discretos y datos continuos.

- **Datos discretos**

Los datos discretos se utilizan para representar elementos que se pueden contar. Puede tomar formas tanto numéricas como categóricas y agruparlas en una lista. Esta lista puede ser finita o infinita también.

Los datos discretos básicamente toman números contables como 1, 2, 3, 4, 5, etc. En el caso del infinito, estos números continuarán.

Ejemplo: días de la semana, días de meses, calificaciones de una prueba, talla de zapatos.

- **Datos continuos**

son un tipo de datos cuantitativos que se pueden medir. Los datos numéricos continuos representan medidas y sus intervalos caen en una recta numérica.

Ejemplo: temperatura, humedad, viscosidad.

## CICLO DE VIDA DE LA CIENCIA DE DATOS



El ciclo de vida de la ciencia de datos se compone esencialmente de:

### 1.- Entender el negocio

Es el punto de partida en el ciclo de vida. Por lo tanto, es importante comprender cuál es la declaración del problema y hacer las preguntas correctas al cliente que nos ayuden a comprender bien los datos y obtener información significativa de los datos.

### 2.- Recolección de datos

El paso principal en el ciclo de vida de los proyectos de ciencia de datos es identificar primero a la persona o personas que sabe qué datos adquirir y cuándo adquirirlos en función de la pregunta a responder. No es necesario que la persona sea un científico de datos, pero cualquiera que conozca la diferencia real entre los diversos conjuntos de datos disponibles y tome decisiones contundentes sobre la estrategia de inversión de datos de una organización, será la persona adecuada para el trabajo.

### 3.- Limpieza de datos

En este paso, comprendemos más acerca de los datos y los preparamos para un análisis posterior. La sección de comprensión de datos de la metodología de ciencia de datos responde a la pregunta: ¿Son los datos que recopiló representativos del problema a resolver?

### 3.- Análisis de datos

El análisis exploratorio a menudo se describe como una filosofía, y no hay reglas fijas sobre cómo abordarlo. No hay atajos para la exploración de datos.

Recuerde que la calidad de sus entradas decide la calidad de su salida. Por lo tanto, una vez que tenga lista su hipótesis comercial, tiene sentido dedicar mucho tiempo y esfuerzo aquí.

### 4.- Modelamiento de datos, modelamiento machine learning

Esta etapa parece ser la más interesante para casi todos los científicos de datos. Mucha gente lo llama “un escenario donde ocurre la magia”. Pero recordemos que la magia solo puede suceder si tienes los accesorios y la técnica correctos. En términos de ciencia de datos, “Datos” es ese apoyo, y la preparación de datos es esa técnica. Entonces, antes de saltar a este paso, asegúrese de pasar suficiente tiempo en los pasos anteriores. El modelado se utiliza para encontrar patrones o comportamientos en los datos. Acá es donde encaja el Machine Learning.

### 5.- Evaluación del modelo

Una pregunta común que los profesionales suelen tener al evaluar el rendimiento de un modelo de aprendizaje automático en qué conjunto de datos debe usar para medir el rendimiento del modelo de aprendizaje automático. Mirar las métricas de rendimiento en el conjunto de datos entrenado es útil, pero no siempre es correcto porque los números obtenidos pueden ser demasiado optimistas, ya que el modelo ya está adaptado al conjunto de datos de entrenamiento. El rendimiento del modelo de aprendizaje automático debe medirse y compararse mediante conjuntos de validación y prueba para identificar el mejor modelo en función de la precisión y el sobreajuste del modelo.

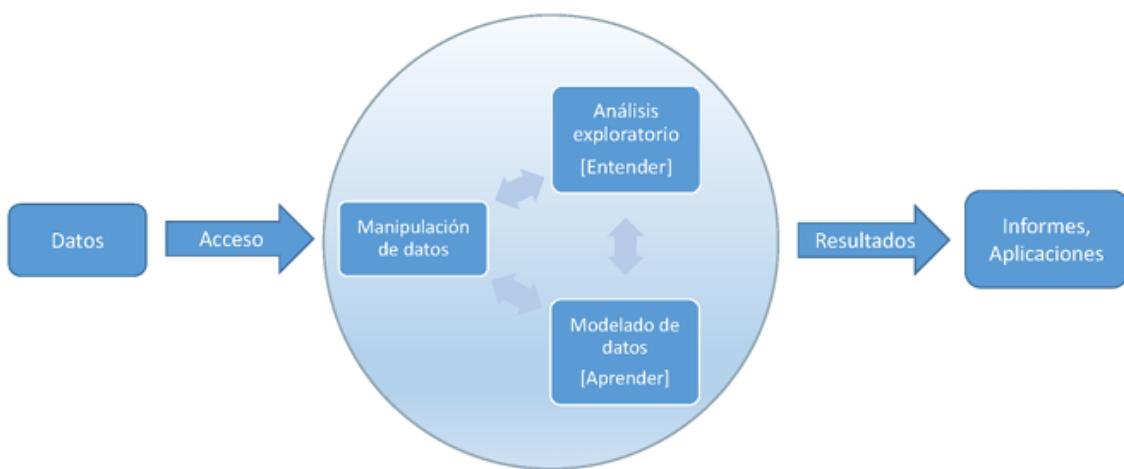
### 6.- Visualización y reportes

En este proceso, las habilidades técnicas por sí solas no son suficientes. Una habilidad esencial que necesita es poder contar una historia clara y procesable. Si su presentación no desencadena acciones en su audiencia, significa que su comunicación no fue eficiente. Debe estar en consonancia con las cuestiones comerciales. Debe ser significativo para la organización y las partes interesadas. La presentación a través de la visualización debe ser

tal que desencadene la acción en la audiencia. Recuerde que se presentará a una audiencia sin conocimientos técnicos, por lo que la forma en que comunica el mensaje es clave.

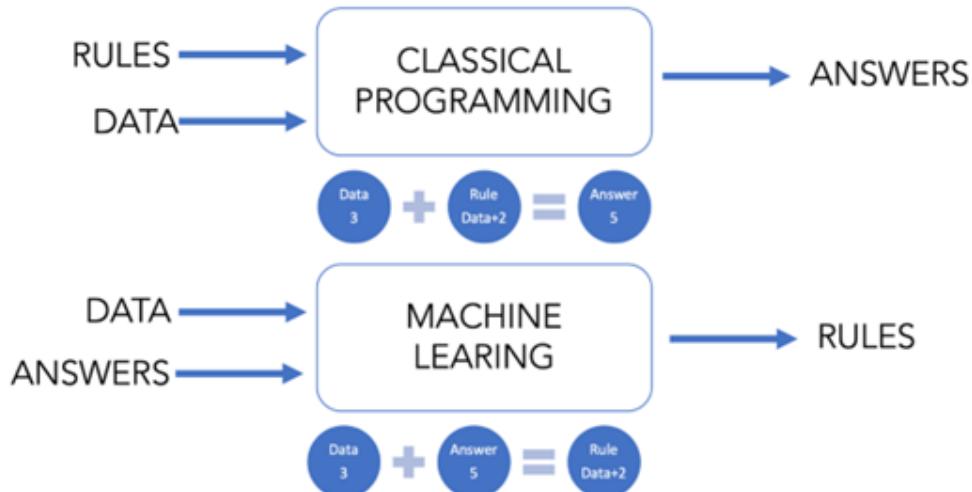
## 7.- Despliegue de modelo

después de construir modelos, primero se implementa en un entorno de preproducción o prueba antes de implementarlos en producción. Cualquiera que sea la forma en que se implemente su modelo de datos, debe exponerse al mundo real. Una vez que los humanos reales lo usen, seguramente recibirás comentarios. Capturar esta retroalimentación se traduce directamente en la vida o la muerte para cualquier proyecto.

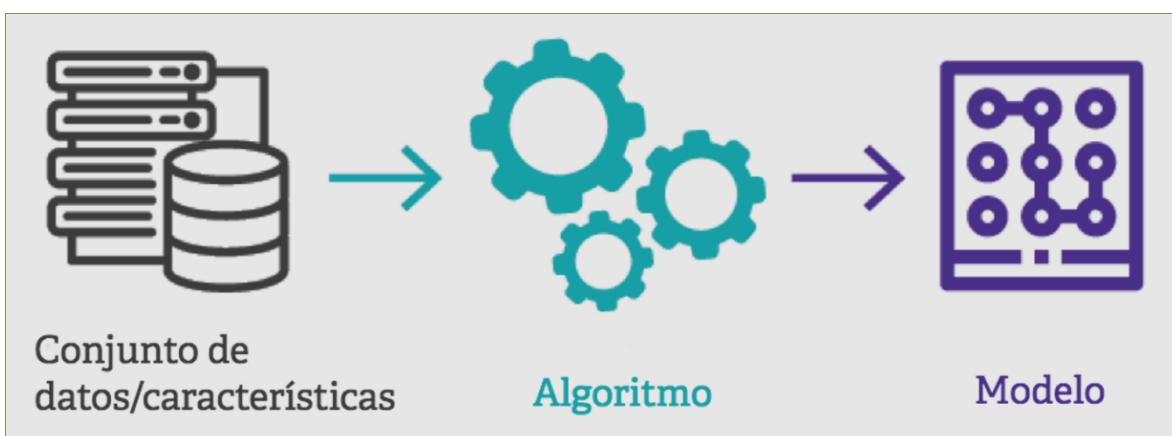


## MACHINE LEARNING

### ¿QUÉ ES EL MACHINE LEARNING?



- Es una rama de la inteligencia artificial. El término se usa desde 1959.
- Es la capacidad de las máquinas para aprender a partir de los datos de manera automatizada.
- Al aprender de manera automatizada, esto implica que no necesitan ser programadas para dicha tarea.
- Esto último es una habilidad indispensable para construir sistemas capaces de identificar patrones entre los datos para hacer predicciones de manera eficiente y confiable.
- El aprendizaje automático es excelente para resolver problemas que requieren mucho trabajo para los humanos, mucho procesamiento de datos.



## Estadística y Machine Learning

Contrariamente a la creencia popular, el aprendizaje automático existe desde hace varias décadas. Inicialmente se rechazó debido a sus grandes requisitos computacionales y las limitaciones de potencia informática presentes en ese momento. Sin embargo, el aprendizaje automático ha experimentado un resurgimiento en los últimos años debido a la preponderancia de los datos derivados de la explosión de la información.

Hay varias declaraciones vagas que escuché a menudo sobre este tema, la más común es algo como esto:

“La principal diferencia entre el aprendizaje automático y las estadísticas es su propósito. Los modelos de aprendizaje automático están diseñados para hacer las predicciones más precisas posibles. Los modelos estadísticos están diseñados para inferir sobre las relaciones entre variables”.

Si bien esto es técnicamente cierto, no da una respuesta particularmente explícita o satisfactoria. Una gran diferencia entre el aprendizaje automático y las estadísticas es, de hecho, su propósito. Sin embargo, decir que el aprendizaje automático tiene que ver con predicciones precisas mientras que los modelos estadísticos están diseñados para la inferencia es casi una declaración sin sentido a menos que esté bien versado en estos conceptos.

En primer lugar, debemos entender que las estadísticas y los modelos estadísticos no son lo mismo. La estadística es el estudio matemático de los datos. No puedes hacer estadísticas a menos que tengas datos. Un modelo estadístico es un modelo para los datos que se utiliza para inferir algo sobre las relaciones dentro de los datos o para crear un modelo que pueda predecir valores futuros. A menudo, estos dos van de la mano.

Entonces, en realidad hay dos cosas que debemos discutir: en primer lugar, en qué se diferencian las estadísticas del aprendizaje automático y, en segundo lugar, en qué se diferencian los modelos estadísticos del aprendizaje automático.

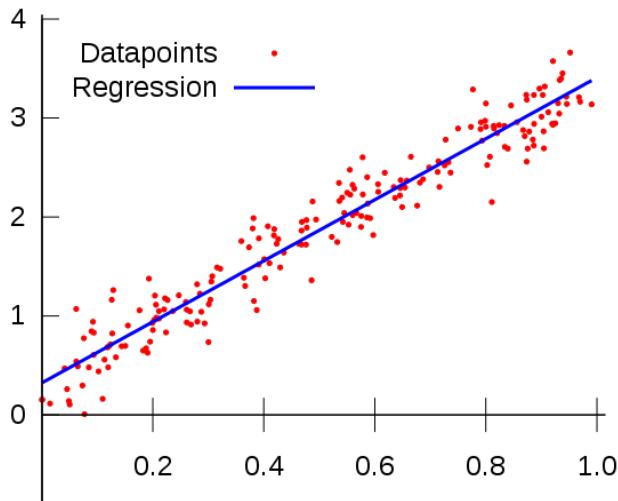
Para hacer esto un poco más explícito, hay muchos modelos estadísticos que pueden hacer predicciones, pero la precisión predictiva no es su punto fuerte.

Del mismo modo, los modelos de aprendizaje automático brindan varios grados de interpretabilidad, desde la regresión de lazo altamente interpretable hasta las redes neuronales impenetrables, pero generalmente sacrifican la interpretabilidad por el poder predictivo.

Desde una perspectiva de alto nivel, esta es una buena respuesta. Lo suficientemente bueno para la mayoría de la gente. Sin embargo, hay casos en los que esta explicación nos deja con un malentendido sobre las diferencias entre el aprendizaje automático y el modelado estadístico.

Veamos el ejemplo de la regresión lineal.

### Modelos estadísticos vs Machine learning — Ejemplo Linear Regression



Me parece que la similitud de los métodos que se utilizan en el modelado estadístico y en el aprendizaje automático ha provocado que las personas asuman que son lo mismo. Esto es comprensible, pero simplemente no es cierto.

El ejemplo más obvio es el caso de la regresión lineal, que es probablemente la principal causa de este malentendido. La regresión lineal es un método estadístico, podemos entrenar un regresor lineal y obtener el mismo resultado que un modelo de regresión estadística con el objetivo de minimizar el error cuadrático entre los puntos de datos.

Vemos que, en un caso, hacemos algo llamado "entrenamiento" del modelo, que implica usar un subconjunto de nuestros datos, y no sabemos qué tan bien funcionará el modelo hasta que "probemos" estos datos en datos adicionales que no estaban presentes. durante el entrenamiento, llamado conjunto de prueba. El propósito del aprendizaje automático, en este caso, es obtener el mejor rendimiento en el conjunto de prueba.

Para el modelo estadístico, encontramos una línea que minimiza el error cuadrático medio en todos los datos, asumiendo que los datos son un regresor lineal con algún ruido aleatorio agregado, que normalmente es de naturaleza gaussiana. No se necesita entrenamiento ni equipo de prueba. Para muchos casos, especialmente en investigación (como el ejemplo del sensor a continuación), el objetivo de nuestro modelo es caracterizar la relación entre los datos y nuestra variable de resultado, no hacer predicciones sobre datos futuros. Llamamos a este procedimiento inferencia estadística, en oposición a predicción. Sin embargo, todavía podemos usar este modelo para hacer predicciones, y este puede ser su objetivo principal,

pero la forma en que se evalúa el modelo no implicará un conjunto de prueba y, en cambio, implicará evaluar la importancia y la solidez de los parámetros del modelo.

El propósito del aprendizaje automático (supervisado) es obtener un modelo que pueda hacer predicciones repetibles. Por lo general, no nos importa si el modelo es interpretable, aunque personalmente recomendaría siempre realizar pruebas para garantizar que las predicciones del modelo tengan sentido. El aprendizaje automático tiene que ver con los resultados, es probable que trabaje en una empresa donde su valor se caracteriza únicamente por su desempeño. Mientras que el modelado estadístico se trata más de encontrar relaciones entre variables y la importancia de esas relaciones, al mismo tiempo que se ocupa de la predicción.

Para dar un ejemplo concreto de la diferencia entre estos dos procedimientos, daré un ejemplo. Un científico ambiental que trabaje principalmente con datos de sensores. Durante el día, puede estar tratando de probar que un sensor pueda responder a cierto tipo de estímulo (como la concentración de un gas), entonces podría un modelo estadístico para determinar si la respuesta de la señal es estadísticamente significativa. Intentaría comprender esta relación y probar su repetibilidad para poder caracterizar con precisión la respuesta del sensor y hacer inferencias basadas en estos datos. Algunas cosas que podría probar son si la respuesta es, de hecho, lineal, si la respuesta se puede atribuir a la concentración de gas y no al ruido aleatorio en el sensor, etc.

Por el contrario, también podría obtener una matriz de 20 sensores diferentes y podría usar esto para intentar predecir la respuesta del sensor recién caracterizado. Esto puede parecer un poco extraño si no sabe mucho sobre sensores, pero actualmente es un área importante de la ciencia ambiental. Un modelo con 20 variables diferentes que predicen el resultado de mi sensor tiene claramente que ver con la predicción, y no espero que sea particularmente interpretable. Este modelo probablemente sería algo un poco más esotérico como una red neuronal debido a las no linealidades que surgen de la cinética química y la relación entre las variables físicas y las concentraciones de gas. Me gustaría que el modelo tuviera sentido, pero mientras pueda hacer predicciones precisas, sería bastante feliz.

Si estoy tratando de probar la relación entre mis variables de datos hasta un grado de significación estadística para poder publicarlo en un artículo científico, usaría un modelo estadístico y no aprendizaje automático. Esto se debe a que me importa más la relación entre las variables que hacer una predicción. Hacer predicciones aún puede ser importante, pero la falta de interpretabilidad que ofrecen la mayoría de los algoritmos de aprendizaje automático hace que sea difícil probar las relaciones dentro de los datos (esto es en realidad un gran problema en la investigación académica actual, con investigadores que usan algoritmos que no entienden y obtienen inferencias engañosas).

Debe quedar claro que estos dos enfoques son diferentes en su objetivo, a pesar de que utilizan medios similares para llegar allí. La evaluación del algoritmo de aprendizaje automático utiliza un conjunto de pruebas para validar su precisión. Mientras que, para un

modelo estadístico, se puede utilizar el análisis de los parámetros de regresión a través de intervalos de confianza, pruebas de significación y otras pruebas para evaluar la legitimidad del modelo. Dado que estos métodos producen el mismo resultado, es fácil ver por qué se podría suponer que son iguales.

### **Estadística vs Machine Learning — Ejemplo Linear Regression**

La física se basa en las matemáticas, es la aplicación de las matemáticas para comprender los fenómenos físicos presentes en la realidad. La física también incluye aspectos de las estadísticas, y la forma moderna de las estadísticas generalmente se construye a partir de un marco que consiste en la teoría de conjuntos de Zermelo-Frankel combinada con la teoría de la medida para producir espacios de probabilidad. Ambos tienen mucho en común porque tienen un origen similar y aplican ideas similares para llegar a una conclusión lógica.

Para dar una idea de hasta dónde llega este debate, en realidad hay un artículo publicado en Nature Methods que describe la diferencia entre las estadísticas y el aprendizaje automático.

<https://www.nature.com/articles/nmeth.4642>

Antes de continuar, aclararé rápidamente otros dos conceptos erróneos comunes relacionados con el aprendizaje automático y las estadísticas. Estos son que la IA es diferente del aprendizaje automático y que la ciencia de datos es diferente de las estadísticas. Estos son problemas bastante indiscutibles, por lo que será rápido.

La ciencia de datos es esencialmente métodos computacionales y estadísticos que se aplican a los datos, estos pueden ser conjuntos de datos pequeños o grandes. Esto también puede incluir cosas como el análisis exploratorio de datos, donde los datos se examinan y visualizan para ayudar al científico a comprender mejor los datos y hacer inferencias a partir de ellos. La ciencia de datos también incluye cosas como la disputa y el preprocesamiento de datos y, por lo tanto, implica cierto nivel de informática, ya que implica codificación, configuración de conexiones y canalizaciones entre bases de datos, servidores web, etc.

No necesariamente necesita usar una computadora para hacer estadísticas, pero realmente no puede hacer ciencia de datos sin una. Una vez más, puede ver que, aunque la ciencia de datos usa estadísticas, claramente no son lo mismo.

Del mismo modo, el aprendizaje automático no es lo mismo que la inteligencia artificial. De hecho, el aprendizaje automático es un subconjunto de la IA. Esto es bastante obvio ya que estamos enseñando ("entrenando") una máquina para hacer inferencias generalizables sobre algún tipo de datos basados en datos previos.

## Análisis descriptivo.

El análisis descriptivo consiste en estudiar todo lo que tiene que ver con el pasado. Se utiliza para describir todos los eventos que han ocurrido, considerando parámetros y referencias que se reflejarán en la toma de decisiones. Para esto, se pueden aplicar varios enfoques y recursos:

**Estadísticas:** Algunos datos estadísticos que se pueden utilizar son el máximo, el mínimo, el promedio, la mediana, los cuartiles, la desviación estándar, la variación o los diez mejores / peores. Esta información se puede ver una por una o agrupada. Un buen ejemplo es el análisis estadístico de las ventas de una empresa multinacional por países.

**Gráficos:** es un elemento visual único que resume los datos que tenemos en las estadísticas. Existen varios tipos de gráficos que, dependiendo de los datos que tenga y de lo que le interesa ver, pueden estar en barras con líneas o circulares, entre varios formatos de organización. Algunos ejemplos pueden ser la evolución de las ventas o los beneficios y costos que puede tener una empresa en particular.

**Tablas:** también es un elemento muy visual para los datos. Un ejemplo son los saldos periódicos de una empresa.

## Análisis predictivo

El análisis predictivo consiste en utilizar el aprendizaje automático para predecir posibles escenarios futuros. Para hacer esto, el usuario debe seguir unos pasos concretos, que son los siguientes:

Definir lo que queremos pronosticar: es esencial aclarar qué predicciones queremos obtener. Por ejemplo, el impacto que tendrá un anuncio en Internet.

Determinar la información sobre la cual se basan las predicciones: es necesario elegir bien los datos para que el pronóstico sea preciso y marque la diferencia en la toma de decisiones y proporcione a la inteligencia artificial los datos históricos necesarios para trabajar en las mejores condiciones posibles.

Los atributos deben ser incluidos, junto con los resultados. Asegurar los datos precisos es esencial. Esto significa que debe crear un modelo que se base en datos de entrada o datos históricos.

Para asegurarse de que el análisis será confiable, el modelo debe ser consistente y constantemente evaluado. Cuando confiamos en nuestro modelo de inteligencia artificial, podemos hacer la predicción final. Un ejemplo podría ser calcular la probabilidad de que un cliente potencial haga clic en anuncios individuales y solicite una compra

## Análisis prescriptivo

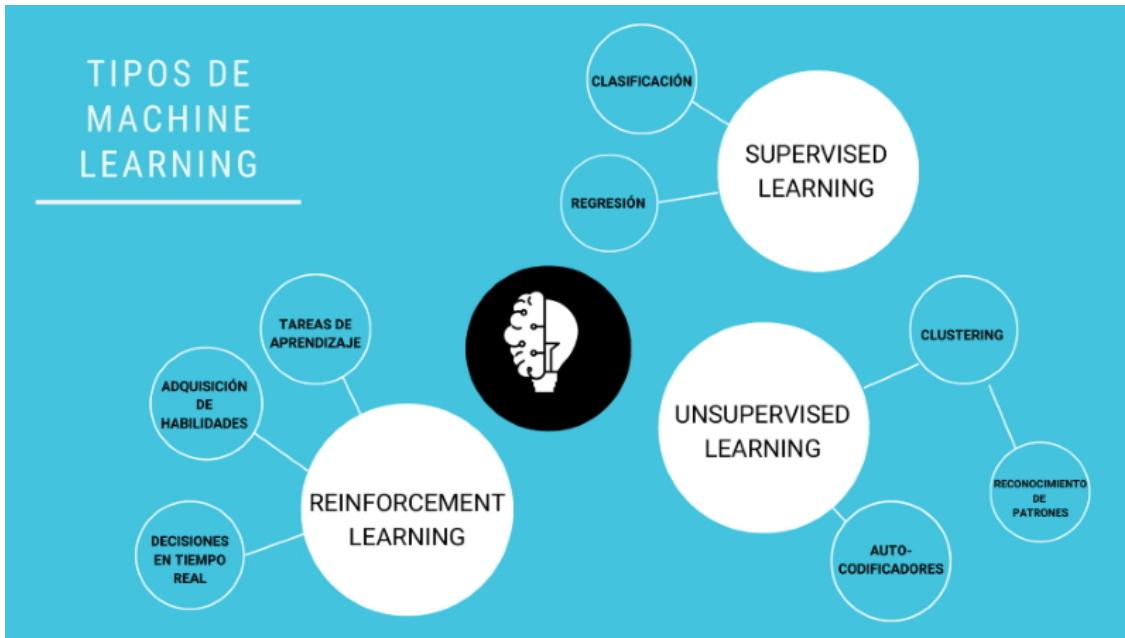
Con el análisis prescriptivo, la inteligencia artificial se pone al servicio de la estrategia de una manera más dinámica y sofisticada, yendo más allá de proporcionar panoramas descriptivos y predictivos. En función de múltiples factores, se indican los mejores caminos a seguir y el posible impacto de diferentes variables.

En otras palabras, con este tipo de análisis evaluamos las decisiones en escenarios futuros, como el impacto que puede tener una acción correctiva dada para que los resultados sean consistentes con el objetivo propuesto.

Por lo tanto, la empresa puede tomar decisiones basadas en un historial de hechos y en vista de diferentes posibilidades y obtener recomendaciones estratégicas para optimizar los resultados en diferentes sectores. Un ejemplo podría ser una compañía telefónica que se da cuenta de que el uso que hace un cliente de sus servicios está disminuyendo. El análisis prescriptivo puede sugerir que existe una optimización de los servicios o un ajuste de los precios para evitar la pérdida de ese cliente.

Los tres tipos de análisis constituyen un poderoso conjunto de herramientas estratégicas de negocios y ciertamente facilitan la toma de decisiones: elevar el nivel de asertividad, potencializar oportunidades y generar resultados.

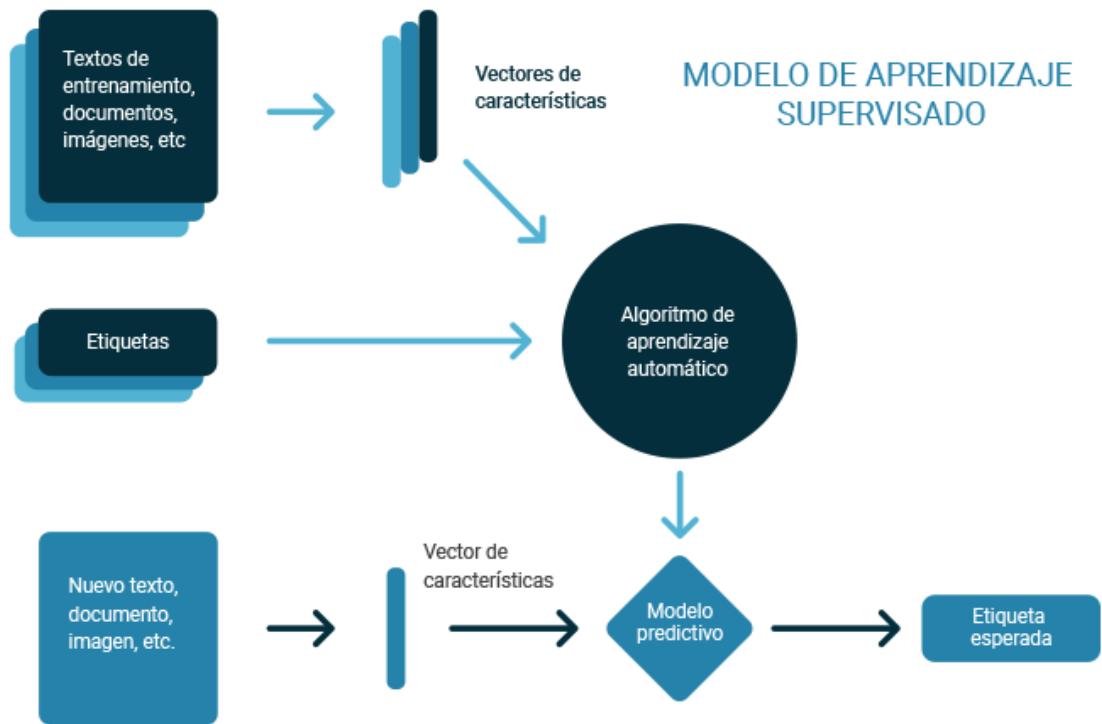
## Construcción del modelo



**¿Qué es un modelo?** Un modelo es simplemente un sistema para mapear entradas a salidas. Por ejemplo, si queremos predecir los precios de las casas, podríamos hacer un modelo que tome los pies cuadrados de una casa y genere un precio. Un modelo representa una teoría sobre un problema: hay alguna conexión entre los pies cuadrados y el precio y hacemos un modelo para aprender esa relación. Los modelos son útiles porque podemos usarlos para predecir los valores de las salidas para nuevos puntos de datos dadas las entradas.

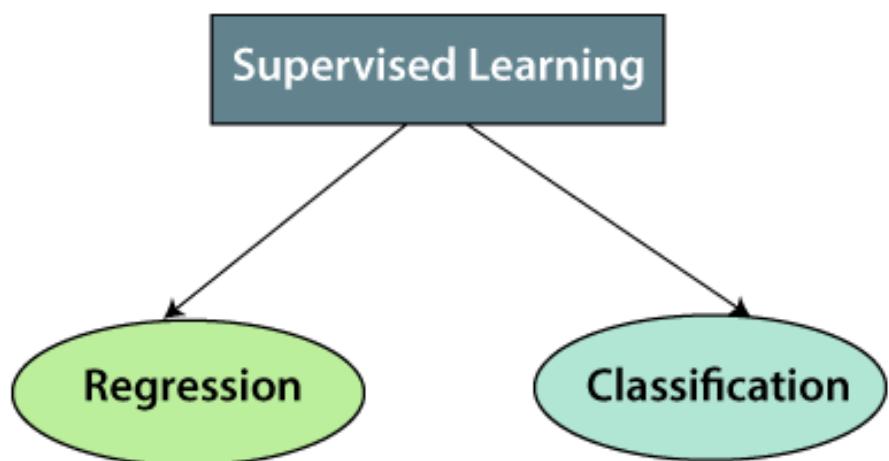
El modelado se utiliza para encontrar patrones o comportamientos en los datos. Estos patrones pueden ser extraídos dependiendo de las necesidades del problema, el ambiente en el que se van a desenvolver y los factores que afectarán la toma de decisiones, utilizando algunos tipos de algoritmos de aprendizaje, entre los cuales vamos a hablar de 3 de ellos: supervisado, no supervisado y por refuerzo.

## Aprendizaje supervisado



El aprendizaje supervisado es el tipo de aprendizaje automático en el que las máquinas se entrena utilizando datos de entrenamiento bien "etiquetados" y, sobre la base de esos datos, las máquinas predicen el resultado. Los datos "etiquetados" significan que algunos datos de entrada ya están marcados con la salida correcta.

Los algoritmos supervisados pueden dividirse en dos tipos de problemas:

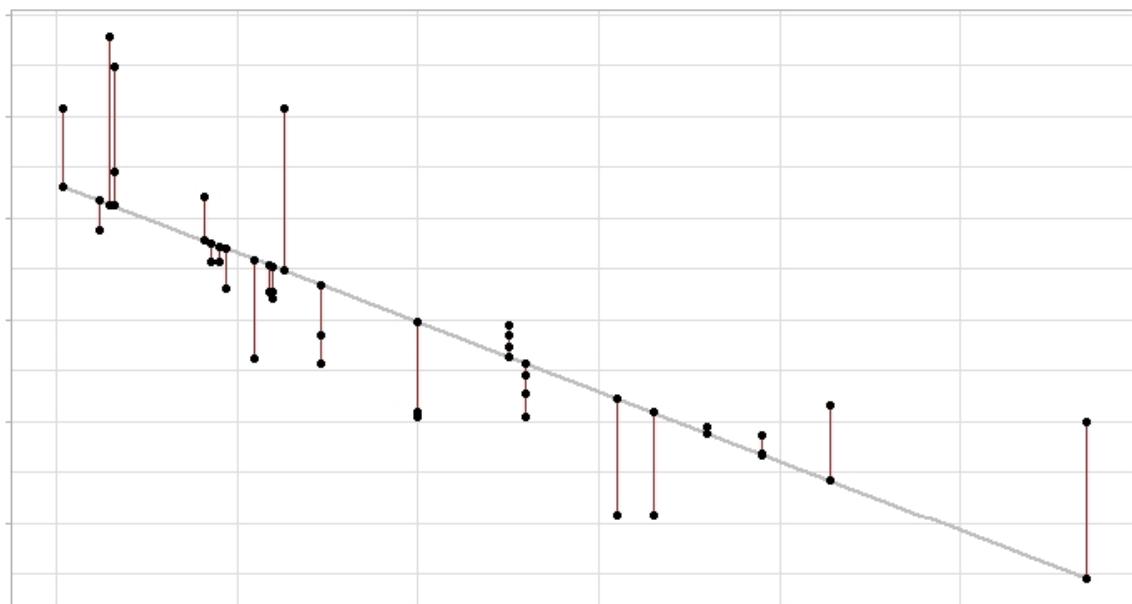


## Algunos algoritmos supervisados

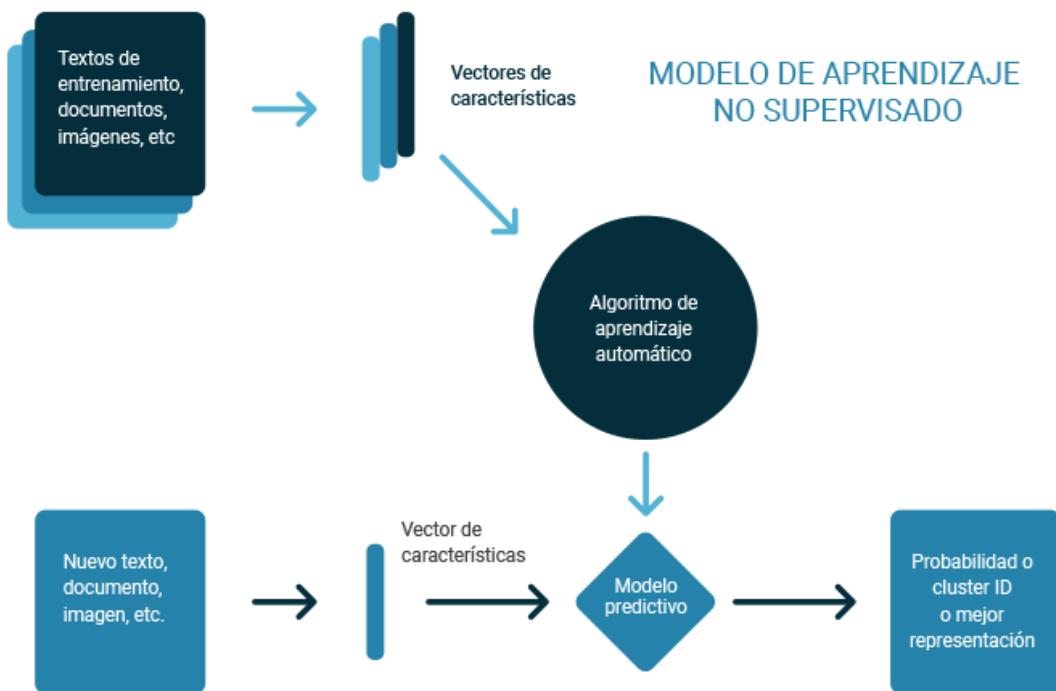
1. Naive Bayes (Clasificacion, modelo no lineal)
2. Neural Networks
3. k-Nearest Neighbor (kNN) (Clasificacion, modelo no lineal)
4. Linear Regression (Regresion)
5. Logistic Regression (Clasificacion, modelo lineal)
6. Support Vector Machines(SVM) (Clasificacion, modelo lineal)
7. Decision Trees (Clasificacion, modelo no lineal)
8. Random Forest (Clasificacion, modelo no lineal)

## Ejemplo

Regresión Lineal

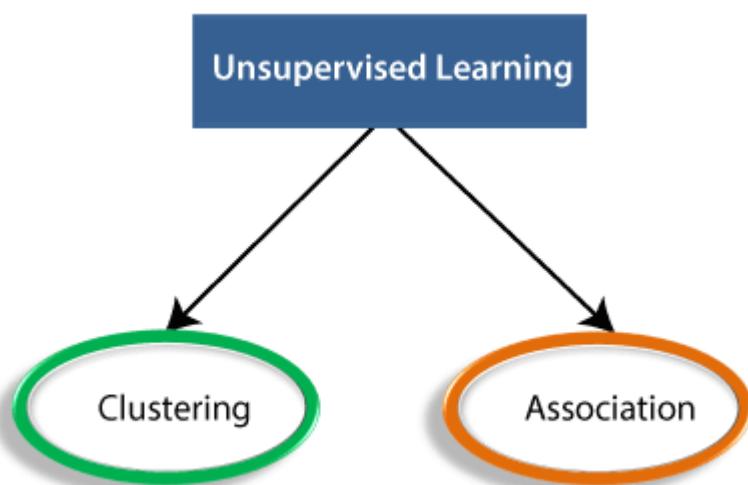


## Aprendizaje no supervisado



Como sugiere el nombre, el aprendizaje no supervisado es una técnica de aprendizaje automático en la que los modelos no se supervisan mediante un conjunto de datos de entrenamiento. En cambio, los propios modelos encuentran los patrones ocultos y los conocimientos de los datos proporcionados. Se puede comparar con el aprendizaje que tiene lugar en el cerebro humano mientras aprende cosas nuevas.

Los algoritmos no supervisados pueden dividirse en dos tipos de problemas:



El objetivo del aprendizaje no supervisado puede ser descubrir grupos de ejemplos similares dentro de los datos, lo que se denomina agrupación (*clustering*), o determinar la distribución de datos dentro del espacio de entrada, conocido como *estimación de densidad*. La estimación de densidad y agrupamiento se puede elegir para conocer los patrones en los datos. Proyectar los datos desde un espacio multidimensional hasta dos o tres dimensiones se elegirá con el propósito de poder visualizar los mismos.

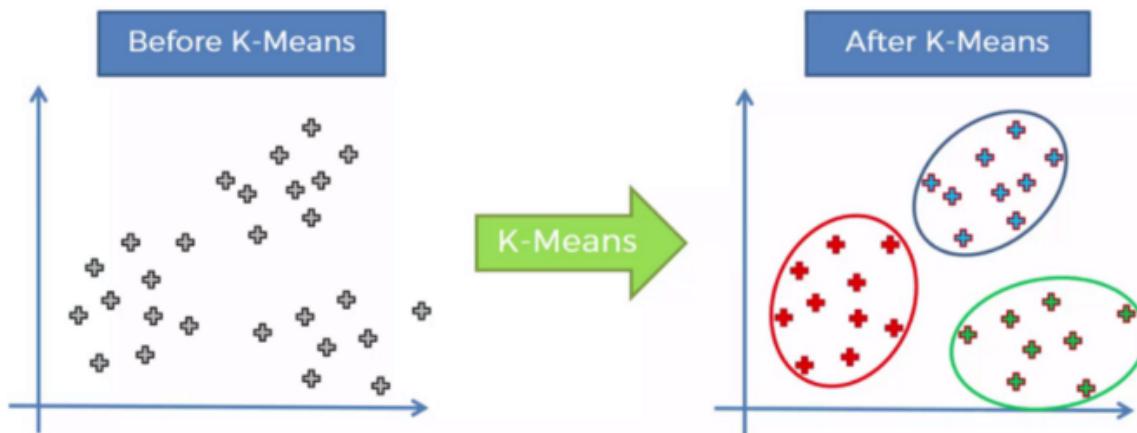
## Algunos algoritmos no supervisados

1. Principle Component Analysis (PCA)
2. KMeans/Kmeans++
3. Hierarchical Clustering
4. DBSCAN
5. Market Basket Analysis

## Ejemplo

Ejemplo de clustering con k-means en Python.

<http://exponentis.es/ejemplo-de-clustering-con-k-means-en-python>



## Aprendizaje reforzado

MODELO DE APRENDIZAJE POR REFUERZO



RL es una aplicación especializada de técnicas de aprendizaje automático/profundo, diseñadas para resolver problemas de una manera particular. A diferencia del aprendizaje supervisado y no supervisado, el aprendizaje por refuerzo es un tipo de aprendizaje que se basa en la interacción con los entornos. Es decir, los algoritmos aprenden a reaccionar ante un entorno por sí mismos. Por lo tanto, la mayor parte de RL es el proceso de prueba y error.

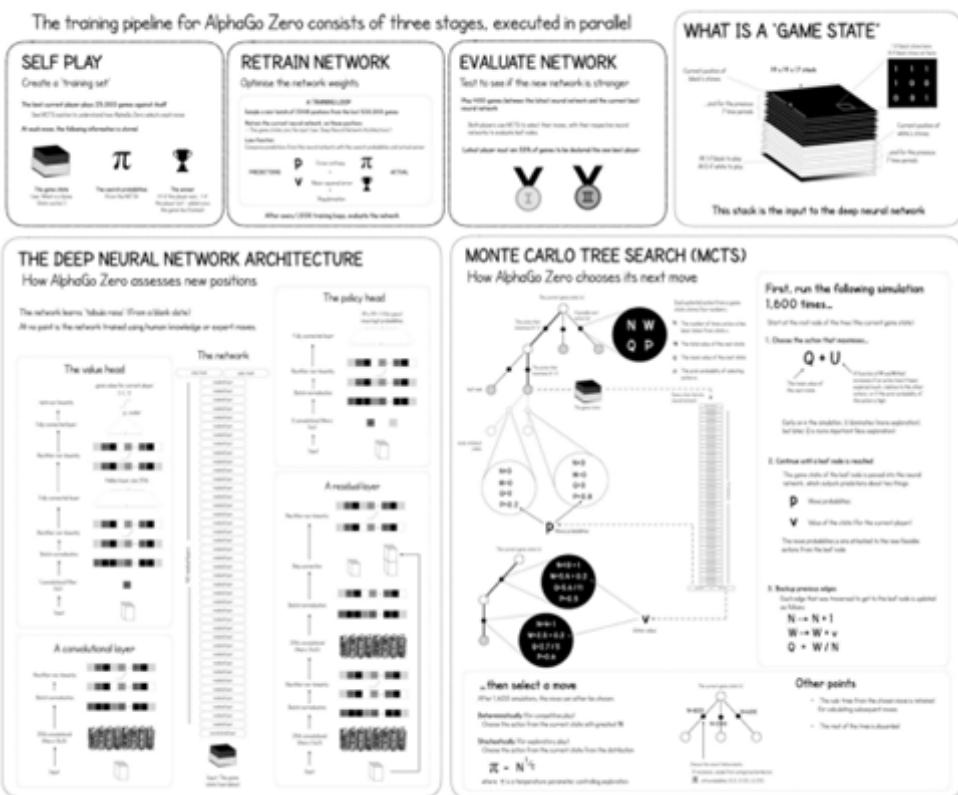
Los modelos RL consisten en algoritmos que utilizan los errores estimados como recompensas o penalizaciones. Si el error es grande, entonces la sanción es alta y la recompensa baja. Si el error es pequeño, la penalización es baja y la recompensa alta. La Figura es una ilustración simple de RL. La forma en que el aprendizaje por refuerzo resuelve problemas es permitiendo que una pieza de software llamada "agente" explore, interactúe y aprenda del entorno.

## El complejo juego Go

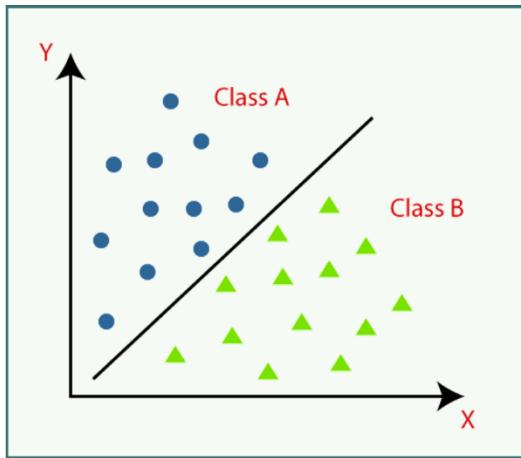


El Go es un juego de mesa tradicional chino con más de 2500 años de antigüedad. Se trata de un juego para 2 personas que, por turnos, van colocando piezas blancas y negras en un tablero estándar de  $19 \times 19$ . El objetivo es capturar las piezas del oponente, eliminándolas así del tablero, o rodear espacios vacíos para hacer puntos de territorio.

### ALPHAGO ZERO CHEAT SHEET



# ALGORITMOS DE CLASIFICACIÓN



El algoritmo de clasificación es una técnica de aprendizaje supervisado que se utiliza para identificar la categoría de nuevas observaciones sobre la base de datos de entrenamiento. En Clasificación, un programa aprende del conjunto de datos u observaciones dado y luego clasifica la nueva observación en una serie de clases o grupos. Por ejemplo, Sí o No, 0 o 1, Spam o No Spam, gato o perro, etc. Las clases se pueden denominar como objetivos/etiquetas o categorías.

A diferencia de la regresión, la variable de salida de Clasificación es una categoría, no un valor, por ejemplo "Verde o Azul", "fruta o animal", etc. Dado que el algoritmo de Clasificación es una técnica de aprendizaje supervisado, toma datos de entrada etiquetados, que significa que contiene entrada con la salida correspondiente.

El algoritmo que implementa la clasificación en un conjunto de datos se conoce como clasificador. Hay dos tipos de Clasificaciones:

## Clasificador binario

Si el problema de clasificación tiene solo dos resultados posibles, se denomina clasificador binario.

Ejemplos: SI o NO, MASCULINO o FEMENINO, SPAM o NO SPAM, GATO o PERRO, etc.

## Clasificador multiclas

Si un problema de clasificación tiene más de dos resultados, se denomina clasificador multiclas.

Ejemplo: Clasificaciones de tipos de cultivos, Clasificación de tipos de música.

## ALGORITMOS DE REGRESIÓN

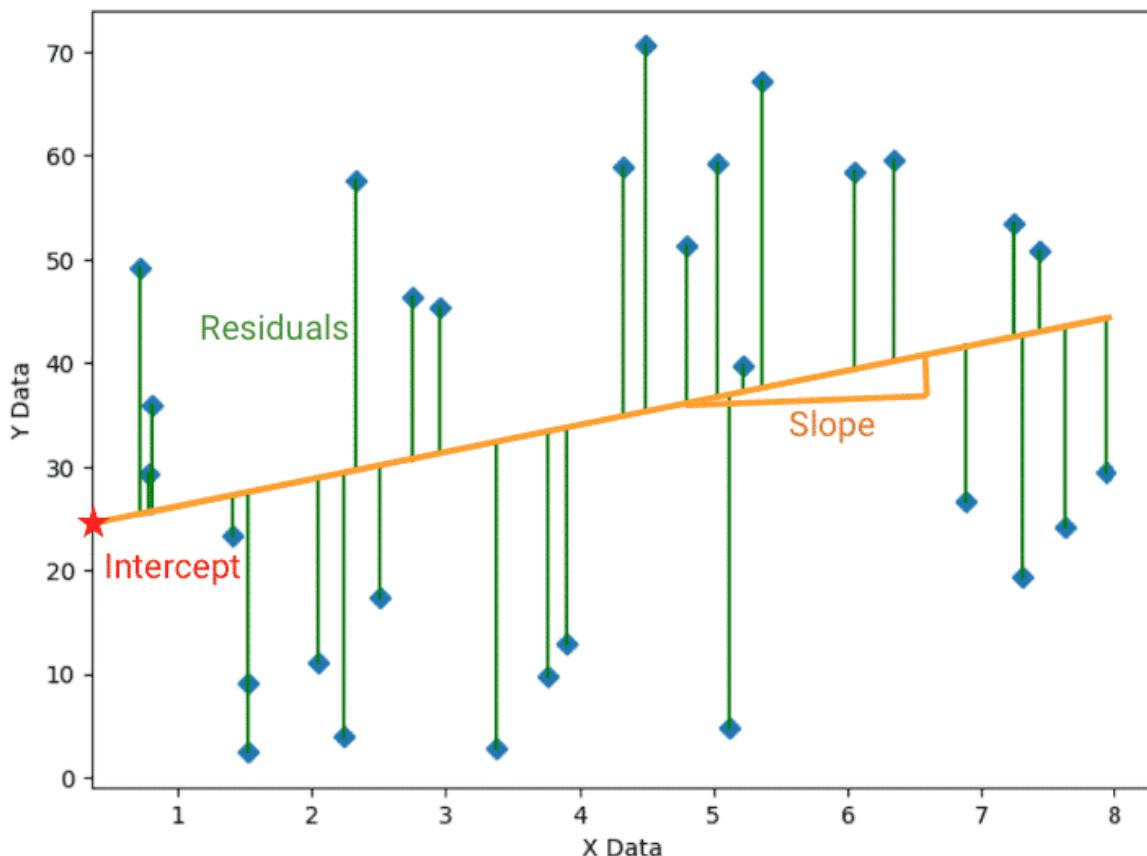
Los algoritmos de regresión intentan estimar la función de mapeo ( $f$ ) a partir de las variables de entrada ( $x$ ) a variables de salida numéricas o continuas ( $y$ ). Ahora, la variable de salida podría ser un valor real, que puede ser un número entero o un valor de coma flotante. Por lo tanto, los problemas de predicción de regresión suelen ser cantidades o tamaños.

Por ejemplo, si se le proporciona un conjunto de datos sobre casas y se le pide que prediga sus precios, se trata de una tarea de regresión porque el precio será una salida continua.

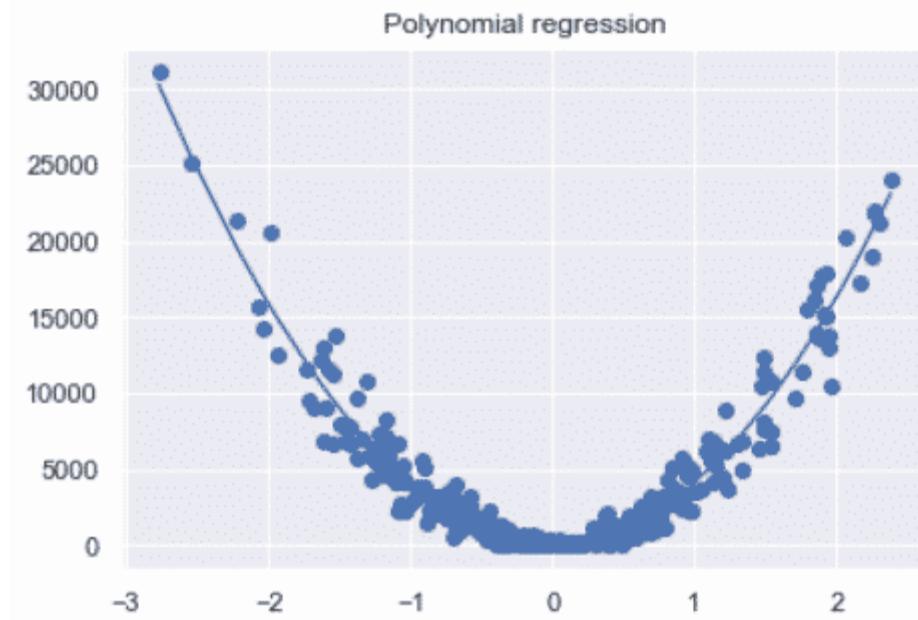
Los ejemplos de los algoritmos de regresión comunes incluyen la regresión lineal, la regresión de vectores de soporte (SVR) y los árboles de regresión.

El análisis de regresión trata de ajustar una línea (o curva) en un gráfico de dispersión de dos variables continuas de manera que los puntos de datos se encuentren colectivamente lo más cerca posible de la línea.

A continuación, se muestra un ejemplo de una regresión lineal en la que la intersección y la pendiente de la línea se colocan de manera que se minimiza la suma de los residuos.

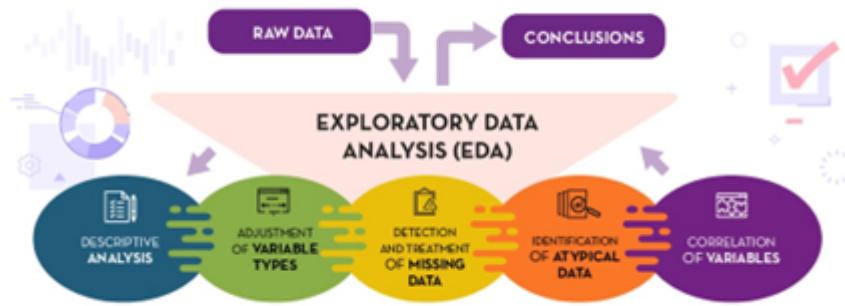


Ejemplo de regresión polinomial:



## DATA SCIENCE LIFE CYCLE

### Análisis exploratorio de datos

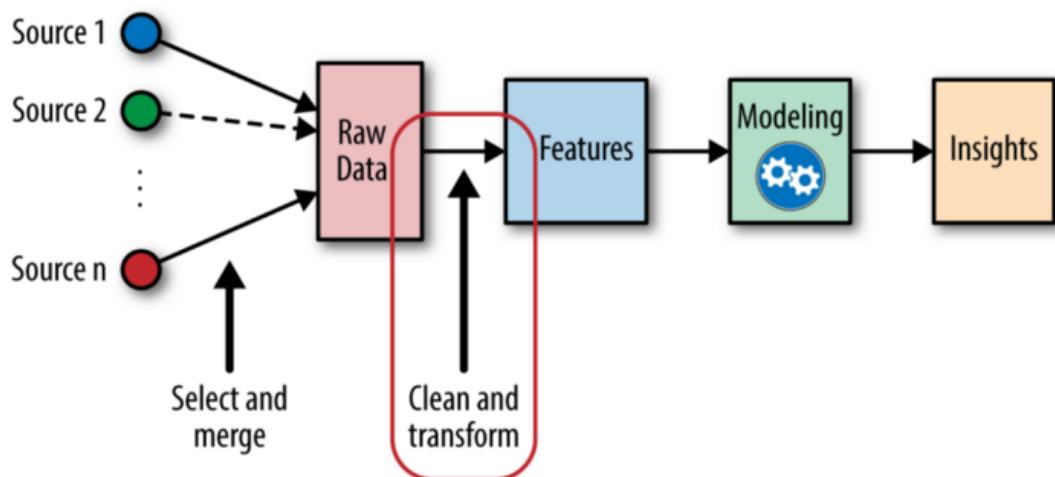


Antes de realizar análisis de datos, con fines estadísticos o predictivos, usando por ejemplo técnicas de aprendizaje automático, es necesario entender la materia prima (raw data) con la que vamos a trabajar. Es necesario comprender y evaluar la calidad de los datos para, entre otros aspectos, detectar y tratar los datos atípicos (outliers) o incorrectos, evitando posibles errores que puedan repercutir en los resultados del análisis.

EDA consiste en aplicar un conjunto de técnicas estadísticas destinadas a explorar, describir y resumir la naturaleza de los datos, de forma que podamos entender claramente como están relacionadas nuestras variables de interés.

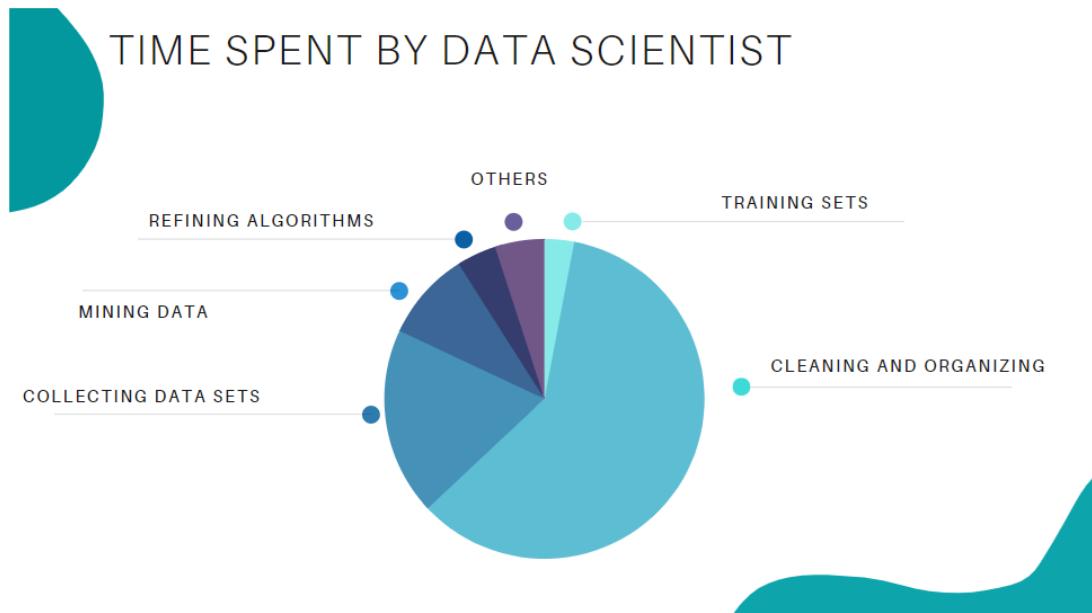
Todo esto nos permite identificar posibles errores, revelar la presencia de outliers, comprobar la relación entre variables (correlaciones) y su posible redundancia, y realizar un análisis descriptivo de los datos mediante representaciones gráficas y resúmenes de los aspectos más significativos.

## Feature engineering



La ingeniería de variables es el proceso de seleccionar, manipular y transformar datos sin procesar en características que se pueden usar en el aprendizaje supervisado. Para que el aprendizaje automático funcione bien en tareas nuevas, puede ser necesario diseñar y entrenar mejores características. Como sabrá, una "variable" es cualquier entrada medible que se puede usar en un modelo predictivo; podría ser el color de un objeto o el sonido de la voz de alguien. La ingeniería de variables, en términos simples, es el acto de convertir observaciones sin procesar en características deseadas utilizando enfoques estadísticos o de aprendizaje automático.

La ingeniería de variables es una técnica de aprendizaje automático que aprovecha los datos para crear nuevas variables que no están en el conjunto de entrenamiento. Puede producir nuevas funciones para el aprendizaje supervisado y no supervisado, con el objetivo de simplificar y acelerar las transformaciones de datos y, al mismo tiempo, mejorar la precisión del modelo. Se requiere ingeniería de funciones cuando se trabaja con modelos de aprendizaje automático. Independientemente de los datos o la arquitectura, una característica terrible tendrá un impacto directo en su modelo.



La ingeniería de variables es un paso muy importante en el aprendizaje automático. La ingeniería de variables se refiere al proceso de diseñar características artificiales en un algoritmo. Estas características artificiales son luego utilizadas por ese algoritmo para mejorar su rendimiento o, en otras palabras, obtener mejores resultados. Los científicos de datos pasan la mayor parte de su tiempo con los datos, y se vuelve importante hacer que los modelos sean precisos.

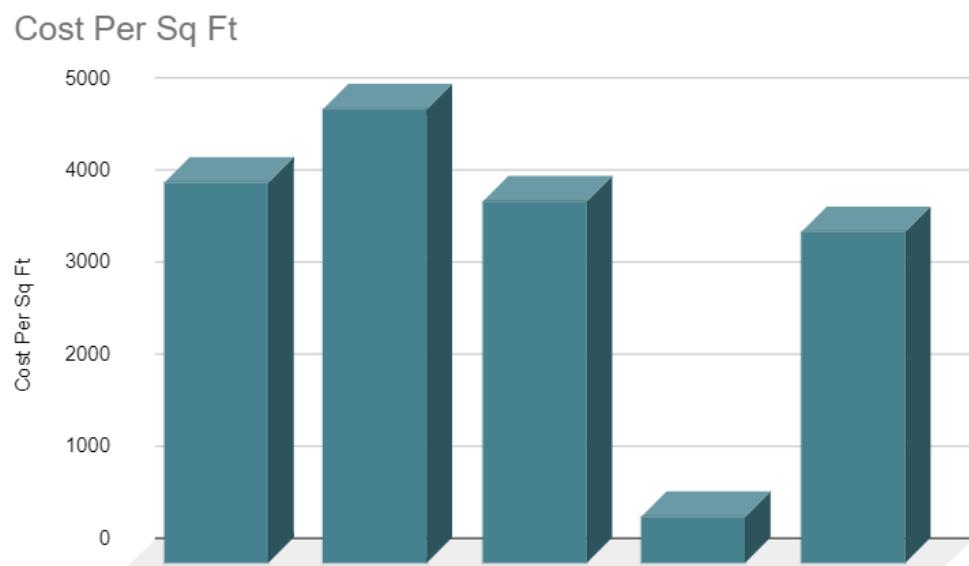
Ahora, para entenderlo de una manera mucho más fácil, tomemos un ejemplo simple. A continuación, se muestran los precios de las propiedades en una ciudad x. Muestra el área de la casa y el precio total.

Sq Ft.	Amount
2400	9 Million
3200	15 Million
2500	10 Million
2100	1.5 Million
2500	8.9 Million

Ahora bien, estos datos pueden tener algunos errores o pueden ser incorrectos, no todas las fuentes en Internet son correctas. Para comenzar, agregaremos una nueva columna para mostrar el costo por pie cuadrado.

Sq Ft.	Amount	Cost Per Sq Ft
2400	9 Million	4150
3200	15 Million	4944
2500	10 Million	3950
2100	1.5 Million	510
2500	8.9 Million	3600

Esta nueva característica nos ayudará a entender mucho sobre nuestros datos. Entonces, tenemos una nueva columna que muestra el costo por pie cuadrado. Hay tres formas principales de encontrar cualquier error. Puede ponerse en contacto con un asesor inmobiliario o agente de bienes raíces y mostrarle la tarifa por pie cuadrado. Si su abogado afirma que el precio por pie cuadrado no puede ser inferior a 3400, es posible que tenga un problema. Los datos se pueden visualizar.

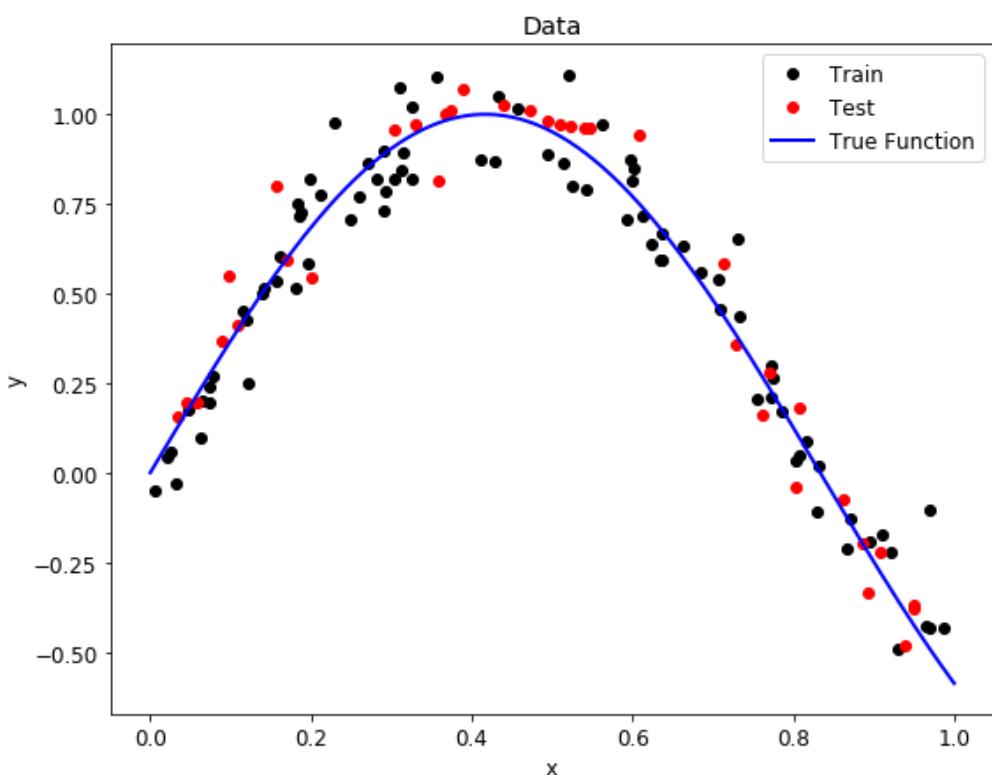


Cuando graficamos los datos, notaremos que un precio es significativamente diferente del resto. En el método de visualización, puede notar fácilmente el problema. La tercera forma es usar Estadísticas para analizar sus datos y encontrar cualquier problema. La ingeniería de características consta de varios procesos:

- Feature Creation
- Transformations
- Feature Extraction
- Exploratory Data Analysis
- Benchmark

## Model building

Para hacer un modelo, primero necesitamos datos que tengan una relación subyacente. Para este ejemplo, crearemos nuestro propio conjunto de datos simple con valores x (características) y valores y (etiquetas). Una parte importante de nuestra generación de datos es agregar ruido aleatorio a las etiquetas. En cualquier proceso del mundo real, ya sea natural o artificial, los datos no se ajustan exactamente a una tendencia. Siempre hay ruido u otras variables en la relación que no podemos medir.

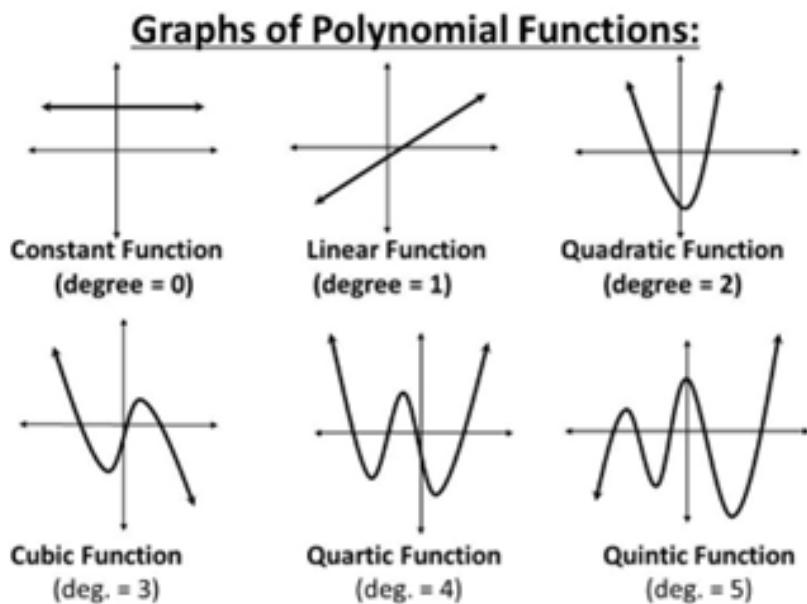


Podemos ver que nuestros datos se distribuyen con alguna variación alrededor de la función verdadera (una onda sinusoidal parcial) debido al ruido aleatorio que agregamos. Durante el entrenamiento, queremos que nuestro modelo aprenda la verdadera función sin ser "distraído" por el ruido.

Para elegir un modelo una buena regla es comenzar de manera simple y luego avanzar. El modelo más simple es una regresión lineal, donde las salidas son una combinación ponderada linealmente de las entradas. En nuestro modelo, usaremos una extensión de la regresión lineal llamada regresión polinomial para conocer la relación entre  $x$  e  $y$ . La regresión polinomial, donde las entradas se elevan a diferentes potencias, todavía se considera una forma de regresión "lineal" aunque el gráfico no forma una línea recta. La ecuación general para un polinomio se encuentra a continuación.

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n + \varepsilon.$$

Aquí  $y$  representa la etiqueta y  $x$  es la característica. Los términos beta son los parámetros del modelo que se aprenderán durante el entrenamiento, y la épsilon es el error presente en cualquier modelo. Una vez que el modelo ha aprendido los valores beta, podemos introducir cualquier valor para  $x$  y obtener una predicción correspondiente para  $y$ . Un polinomio se define por su orden, que es la potencia más alta de  $x$  en la ecuación. Una recta es un polinomio de grado 1 mientras que una parábola tiene 2 grados.



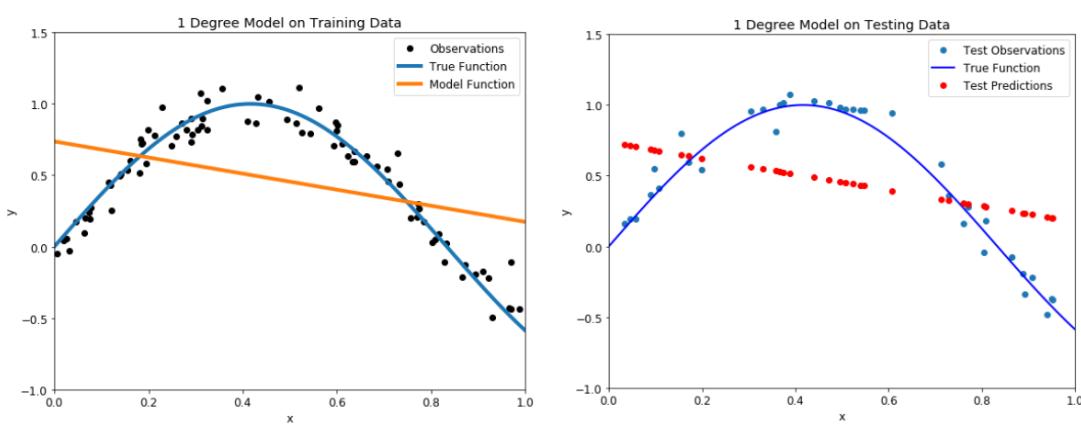
## Underfitting

En estadísticas y aprendizaje automático, generalmente dividimos nuestros datos en dos subconjuntos: datos de entrenamiento y datos de prueba (y, a veces, en tres: entrenar, validar y probar), y ajustamos nuestro modelo en los datos de entrenamiento, para hacer predicciones sobre los datos de prueba. . Cuando hacemos eso, puede suceder una de dos cosas: overfitting (sobreajustamos nuestro modelo) o (underfitting) no ajustamos bien nuestro modelo. No queremos que suceda ninguna de estas cosas, porque afectan la previsibilidad de nuestro modelo; es posible que estemos usando un modelo que tiene menor precisión y/o no está generalizado (lo que significa que no puede generalizar sus predicciones en otros datos).

**¿Qué ocurre cuando sobreentrenamos nuestro modelo? ¿Y si generalizamos demasiado el conocimiento que pretendemos que el modelo adquiera? Que es el overfitting y underfitting, cuales son las causas y consecuencias de un modelo mal entrenado.**

Cuando un modelo no ha aprendido bien los patrones en los datos de entrenamiento y no puede generalizar bien los nuevos datos, se conoce como ajuste insuficiente. Un modelo inadecuado tiene un rendimiento deficiente en los datos de entrenamiento y dará como resultado predicciones poco confiables. El ajuste insuficiente ocurre debido al alto sesgo y la baja varianza.

Un modelo de ajuste insuficiente con un ajuste polinomial de 1 grado. En la imagen inferior izquierda, la función del modelo en naranja se muestra encima de la función real y las observaciones de entrenamiento. A la derecha, se muestran las predicciones del modelo para los datos de prueba en comparación con la función real y los puntos de datos de prueba.

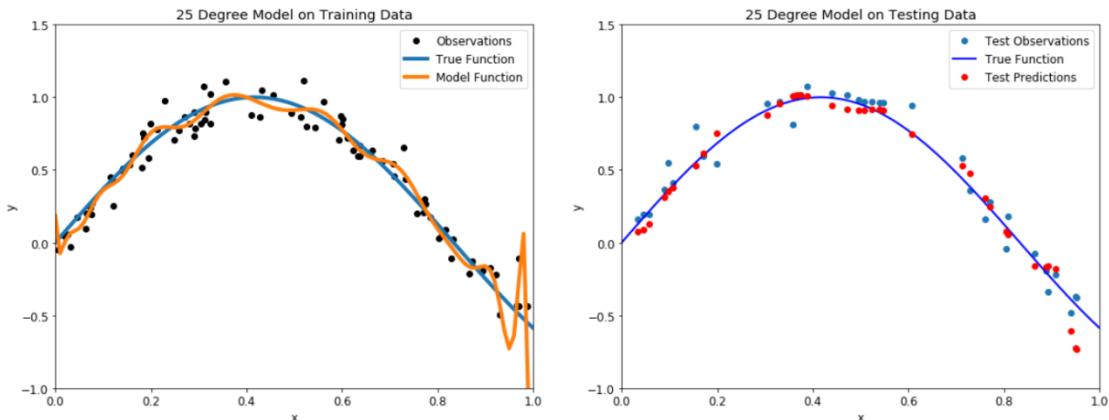


Nuestro modelo presenta **underfitting** pues tiene una varianza baja y un sesgo alto. La varianza se refiere a cuánto depende el modelo de los datos de entrenamiento. Para el caso de un polinomio de 1 grado, el modelo depende muy poco de los datos de entrenamiento porque apenas presta atención a los puntos. En cambio, el modelo tiene un alto sesgo, lo que

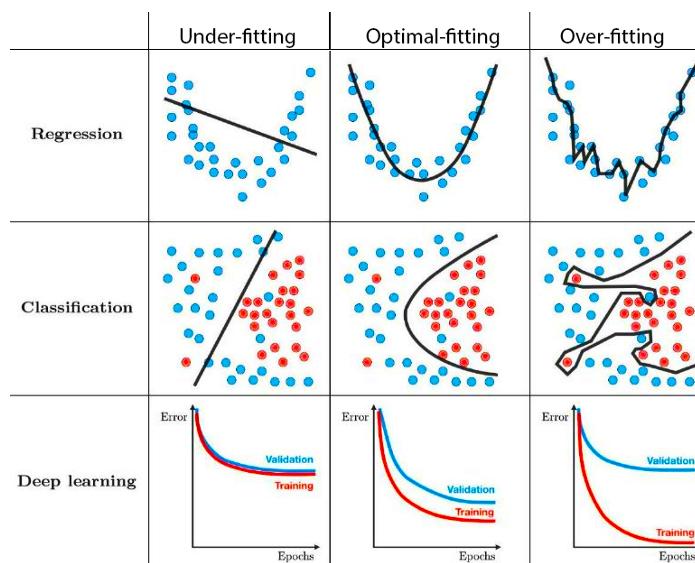
significa que hace una fuerte suposición sobre los datos. Para este ejemplo, la suposición es que los datos son lineales, lo que evidentemente es bastante erróneo. Cuando el modelo hace predicciones de prueba, el sesgo lo lleva a hacer estimaciones inexactas. El modelo no pudo aprender la relación entre x e y debido a este sesgo, un claro ejemplo de ajuste insuficiente.

## Overfitting

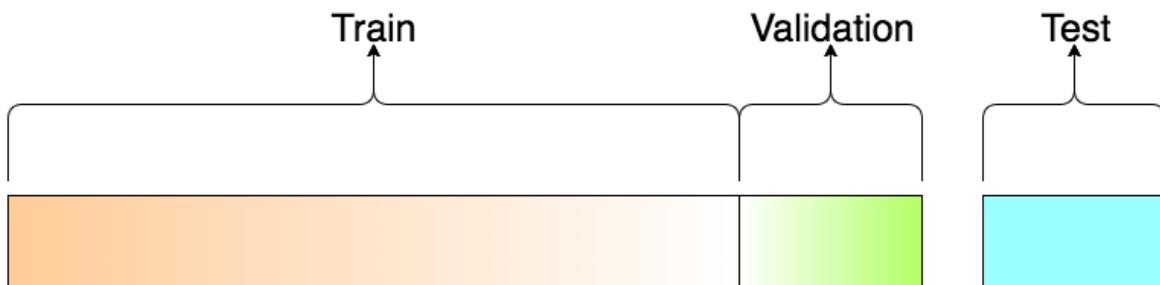
Vimos que un grado bajo conduce al ***underfitting***. Una conclusión natural sería aprender bien los datos de entrenamiento, solo deberíamos aumentar el grado del modelo para capturar cada cambio en los datos.



Esto no es siempre o mejor. Con un grado tan alto de flexibilidad, el modelo hace todo lo posible para tener en cuenta cada punto de entrenamiento. Esto puede parecer una buena idea, ¿no queremos aprender de los datos? Además, el modelo tiene una gran puntuación en los datos de entrenamiento porque se acerca a todos los puntos. Si bien esto sería aceptable si las observaciones de entrenamiento representaran perfectamente la función verdadera, debido a que hay ruido en los datos, nuestro modelo termina ajustando el ruido. Este es un modelo con una varianza alta, porque cambiará significativamente dependiendo de los datos de entrenamiento. Las predicciones en el conjunto de prueba son mejores que el modelo de un grado, pero el modelo de veinticinco grados aún no aprende la relación porque esencialmente memoriza los datos de entrenamiento y el ruido.



## Train, validation y test sets



### Conjunto de entrenamiento

Conjunto de ejemplos utilizados para el aprendizaje. El conjunto de entrenamiento suele ser el conjunto más grande, en términos de tamaño, que se crea a partir del conjunto de datos original y se utiliza para encontrar el modelo. En otras palabras, los puntos de datos incluidos en el conjunto de entrenamiento se utilizan para aprender los parámetros del modelo de interés.

### Conjunto de validación

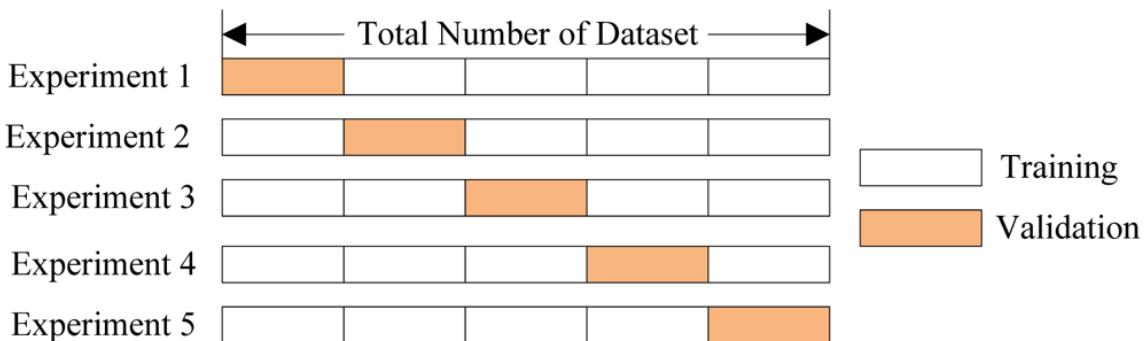
Un conjunto de ejemplos utilizados para ajustar los híper parámetros de un clasificador, por ejemplo, para elegir el número de unidades ocultas en una red neuronal. Luego veremos como introducir la validación como una validación k-fold.

### Conjunto de prueba

Un conjunto de ejemplos utilizados solo para evaluar el rendimiento de un clasificador completamente especificado. El conjunto de prueba se utiliza para evaluar el rendimiento de este modelo y garantizar que pueda generalizarse bien a puntos de datos nuevos e invisibles.

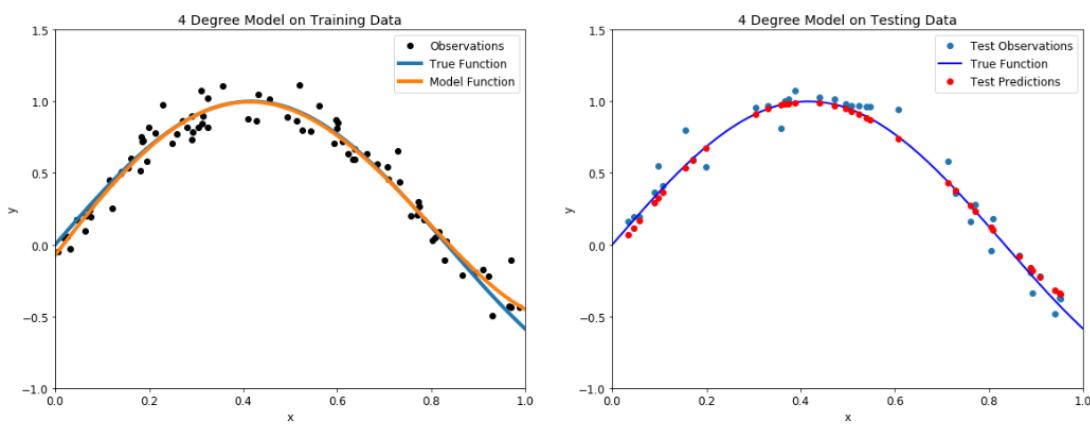
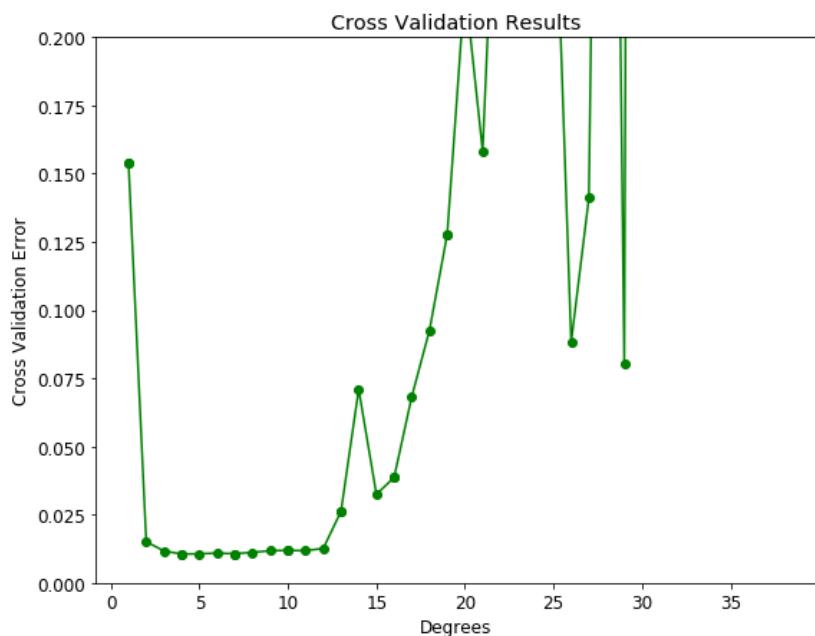
## Enter validation

Necesitamos que nuestro modelo no "memorice" los datos de entrenamiento, sino que aprenda la relación real. ¿Cómo podemos encontrar un modelo balanceado con el grado polinomial correcto? Si elegimos el modelo con la mejor puntuación en el conjunto de entrenamiento, simplemente seleccionaremos el modelo de sobreajuste, pero esto no se puede generalizar bien para los datos de prueba. Afortunadamente, existe una técnica de ciencia de datos bien establecida para desarrollar el modelo óptimo: la validación.

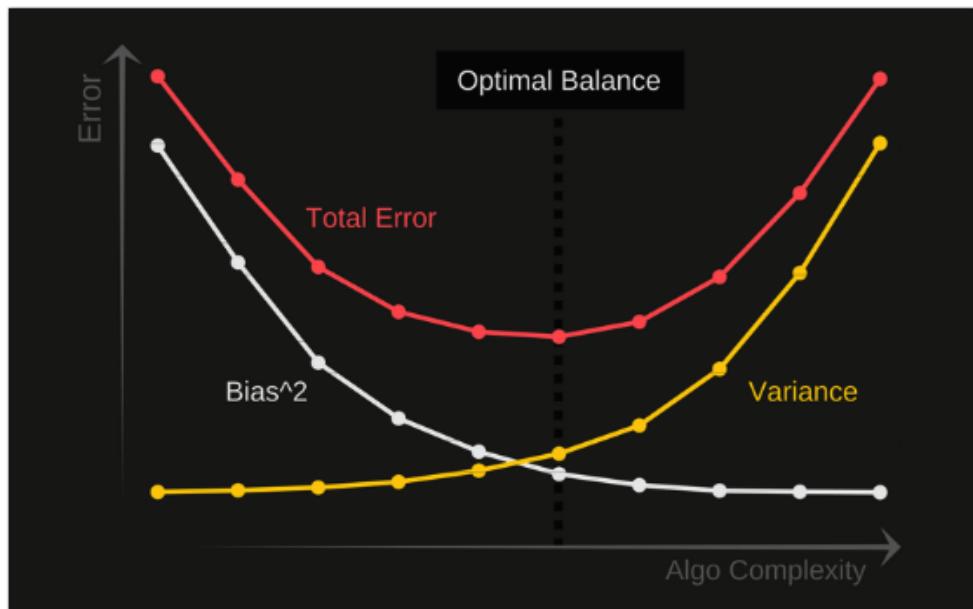


En lugar de usar un conjunto de validación separado, dividimos el conjunto de entrenamiento en varios subconjuntos, llamados folds. Usemos cinco folds como ejemplo. Realizamos una serie de ciclos de entrenamiento y evaluación donde cada vez entrenamos en 4 de los folds y probamos en el quinto, llamado conjunto de espera. Repetimos este ciclo 5 veces, cada vez usando un fold diferente para la evaluación. Al final, promediamos las puntuaciones de cada uno de los folds para determinar el rendimiento general de un modelo determinado. Esto nos permite optimizar el modelo antes de la implementación sin tener que utilizar datos adicionales.

Para nuestro problema, podemos usar la validación cruzada para seleccionar el mejor modelo creando modelos con un rango de diferentes grados y evaluar cada uno usando la validación cruzada de 5 veces. El modelo con la puntuación de validación cruzada más baja se desempeñará mejor en los datos de prueba y logrará un equilibrio entre el ajuste insuficiente y el ajuste excesivo. Se sugiere usar modelos con grados del 1 al 40 para cubrir una amplia gama. Para comparar modelos, calculamos el error cuadrático medio, la distancia promedio entre la predicción y el valor real al cuadrado. La siguiente tabla muestra los resultados de la validación cruzada ordenados por menor error y el gráfico muestra todos los resultados con error en el eje y.



## Bias-variance tradeoff



Cada vez que discutimos la predicción del modelo, es importante comprender los errores de predicción (sesgo y varianza). Existe una compensación entre la capacidad de un modelo para minimizar el sesgo y la varianza. Obtener una comprensión adecuada de estos errores nos ayudara no solo a construir modelos precisos, sino también a evitar el error de sobreajuste y ajuste insuficiente.

### Bias

El sesgo se refiere a las suposiciones erróneas del modelo generado acerca de los datos. Un sesgo alto o underfitting (ajuste insuficiente) significa que el modelo no puede capturar la tendencia o el patrón en los datos. Por lo general, se produce cuando la función de hipótesis es demasiado simple o tiene muy pocos features.

El modelo con alto sesgo no aprende bien de los datos de entrenamiento y simplifica demasiado el modelo. Tiene un desempeño deficiente en el conjunto de entrenamiento y prueba porque no puede identificar patrones en los datos.

### Variance

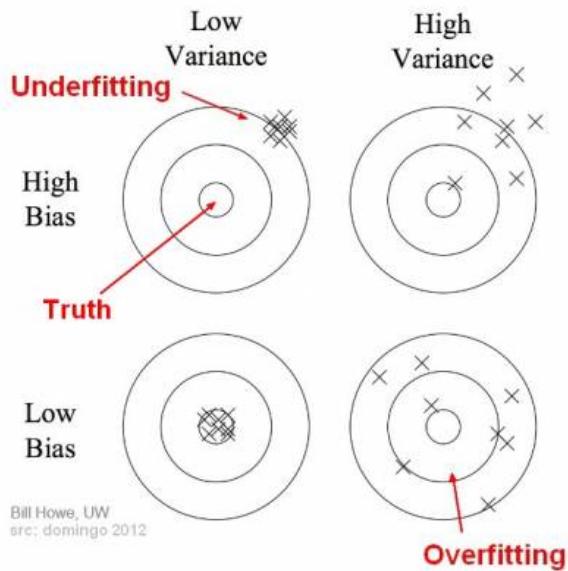
La varianza se refiere a la capacidad del modelo para medir la dispersión de los datos. Varianza alta o overfitting (sobreajuste) significa que el modelo se ajusta demasiado a los datos de entrenamiento pero no generaliza bien para predecir con datos nuevos. Por lo general, se produce cuando la función de hipótesis es demasiado compleja e intenta ajustar con precisión cada punto de datos en el conjunto de datos de entrenamiento, lo

que genera muchas curvas y ángulos innecesarios que no están relacionados con los datos.

Como resultado, un modelo con varianza alta se desempeña muy bien en el conjunto de entrenamiento, pero deficientemente en el conjunto de prueba o validación cruzada. No puede generalizar y se desempeña mal en cualquier conjunto de datos que no haya visto antes. Por lo tanto, la precisión del entrenamiento será alta y la precisión de la prueba será baja.

## MACHINE LEARNING

### Evaluación del modelo



En el diagrama anterior, el centro del objetivo es un modelo que predice perfectamente los valores correctos. A medida que nos alejamos de la diana, nuestras predicciones empeoran cada vez más. Podemos repetir nuestro proceso de creación de modelos para obtener resultados separados en el objetivo.

En el aprendizaje supervisado, el **underfitting** ocurre cuando un modelo no puede capturar el patrón subyacente de los datos. Estos modelos suelen tener un alto sesgo y una baja varianza. Ocurre cuando tenemos muy poca cantidad de datos para construir un modelo preciso o cuando intentamos construir un modelo lineal con datos no lineales.

En el aprendizaje supervisado, el **overfitting** ocurre cuando nuestro modelo captura el ruido junto con el patrón subyacente en los datos. Ocurre cuando entrenamos mucho nuestro modelo sobre un conjunto de datos ruidoso. Estos modelos tienen un sesgo bajo y una varianza alta.

#### ¿Cómo arreglar el alto sesgo?

Podemos aumentar las características o realizar ingeniería de características para agregar factores más significativos a los datos. Esto puede ayudar al modelo a comprender bien los datos. Aumentar el grado del polinomio en la función de hipótesis también puede ayudar a combatir el alto sesgo porque los modelos con alto sesgo son demasiado simples y aumentar el grado del polinomio puede aumentar la complejidad y, por lo tanto, reducir el sesgo.

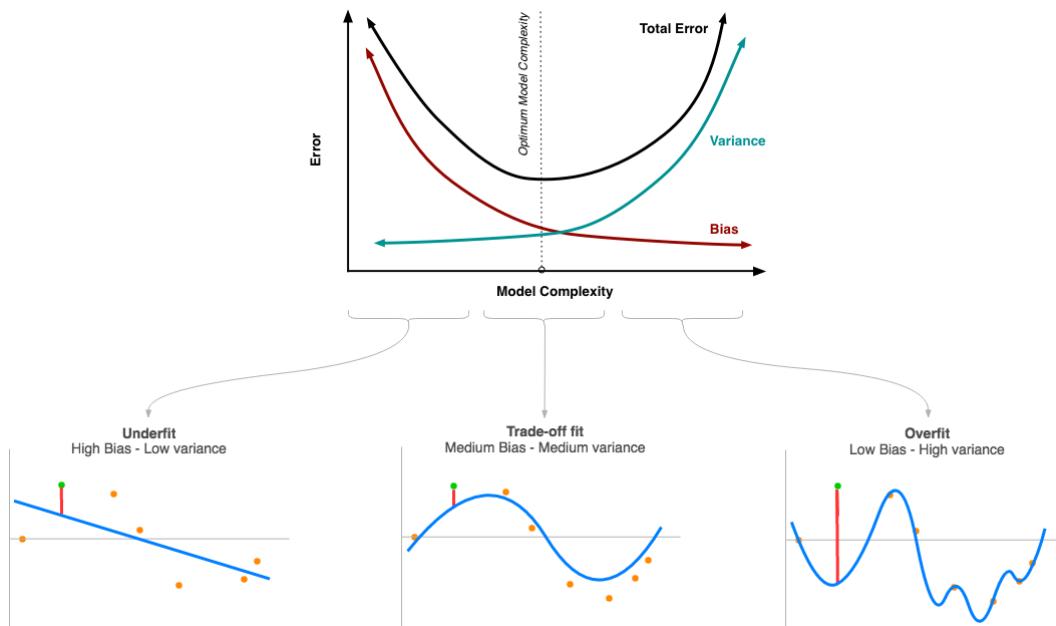
Sin embargo solo hasta cierto punto se debe aumentar la complejidad, caso contrario, el error de validación cruzada comienza a aumentar. También se puede intentar disminuir el parámetro alfa de Regularización

### ¿Cómo arreglar la alta varianza?

Puede reducir la varianza alta al reducir la cantidad de features en el modelo. Hay varios métodos disponibles para verificar qué features no agregan mucho valor al modelo y cuáles son importantes. Aumentar el tamaño del conjunto de entrenamiento también puede ayudar a generalizar el modelo. Disminuir el grado del polinomio puede ayudar a disminuir la complejidad del modelo y solucionar el problema de la alta varianza.

### ¿Cómo mantener un equilibrio de sesgo y varianza?

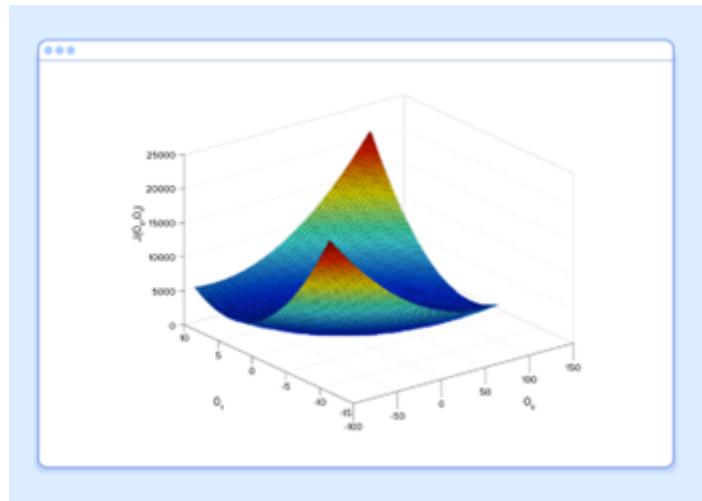
Aumentar el sesgo puede disminuir la varianza, mientras que aumentar la varianza puede disminuir el sesgo. ¿Cómo podemos lograr el punto perfecto u óptimo para un buen modelo?



<http://www.ebc.cat/author/eduard-bonadagmail-com/>

Como se muestra en la figura anterior, existe un punto en el que el error de validación cruzada comienza a aumentar debido al aumento de la varianza y la disminución del sesgo. Este es el punto exacto donde el modelo necesita dejar de aumentar su complejidad y usar todos los parámetros definidos por ese punto en la curva. Por lo general, aquí es donde las curvas de sesgo y varianza se cruzan creando el punto óptimo de complejidad del modelo. En este punto, el modelo tiene un sesgo bajo y una varianza baja sin que se produzca un ajuste insuficiente o excesivo del modelo.

## Función de perdida y función de costo

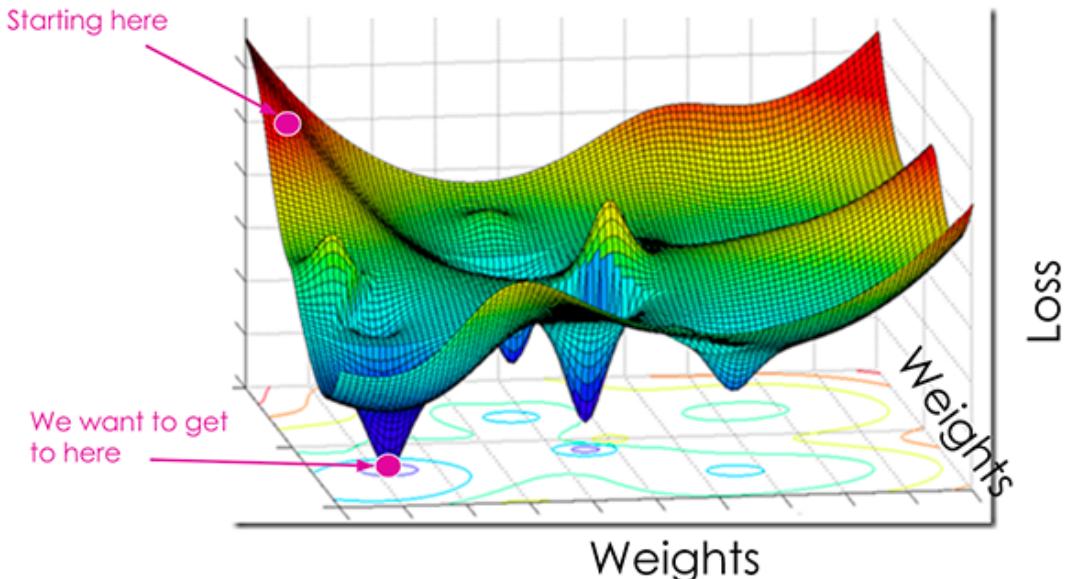


Una función de pérdida  $J(x)$  mide que tan insatisfechos estamos con las predicciones de nuestro modelo con respecto a una respuesta correcta y utilizando ciertos valores de  $\theta$ . Existen varias funciones de pérdida y la selección de uno de ellos depende de varios factores como el algoritmo seleccionado o el nivel de confianza deseado, pero principalmente depende del objetivo de nuestro modelo.

La palabra 'Pérdida' establece la penalización por no lograr el resultado esperado. Si la desviación en el valor predicho del valor esperado por nuestro modelo es grande, entonces la función de pérdida da el número más alto como resultado y si la desviación es pequeña y mucho más cercana al valor esperado, genera un número menor.

Una función que calcula la pérdida para 1 punto de datos se llama función de pérdida. Función de costo es el promedio de todos los errores de la muestra en todo el conjunto de entrenamiento.

## Evaluación del modelo



Imagínese esto: ha entrenado un modelo de aprendizaje automático en un conjunto de datos determinado y está listo para ponerlo frente a su cliente. Pero, ¿cómo puede estar seguro de que este modelo dará el resultado óptimo? ¿Existe alguna métrica o técnica que lo ayude a evaluar rápidamente su modelo en el conjunto de datos?

La evaluación del modelo es un método para evaluar la corrección de los modelos en los datos de prueba. Los datos de prueba consisten en puntos de datos que el modelo no ha visto antes.

Los modelos se pueden evaluar utilizando múltiples métricas. Sin embargo, la elección correcta de una métrica de evaluación es crucial y, a menudo, depende del problema que se está resolviendo. Una comprensión clara de una amplia gama de métricas puede ayudar al evaluador a encontrar una coincidencia adecuada entre el enunciado del problema y una métrica.

Medir la precisión (o el error) de los pronósticos no es una tarea fácil, ya que no existe un indicador único para todos. Solo la experimentación le mostrará qué indicador clave de rendimiento (KPI) es mejor para cada caso. Como veremos, cada indicador evitirá algunas trampas pero será propenso a otras.

Tomar en cuenta que es irresponsable establecer objetivos arbitrarios de desempeño de pronósticos (como MAPE < 10 % es Excelente, MAPE < 20 % es Bueno) sin el contexto de la capacidad de pronóstico de sus datos.

## Métricas de performance para regresión (Funciones de perdida)

### Error Absoluto Medio (MAE)

$$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Los errores no se ponderan más o menos, sino que las puntuaciones aumentan linealmente con el aumento de los errores. La puntuación MAE se mide como el promedio de los valores de error absolutos. Por tanto, la diferencia entre un valor esperado y un valor predicho puede ser positiva o negativa pero será necesariamente positiva al calcular el MAE. Cuanto menor sea el MAE, mayor será la precisión de un modelo.

#### *Ventajas de MAE:*

Es una métrica de evaluación fácil de calcular.

Todos los errores se ponderan en la misma escala ya que se toman valores absolutos.

Es útil si los datos de entrenamiento tienen valores atípicos, ya que MAE no penaliza los errores elevados causados por los valores atípicos.

Proporciona una medida uniforme de qué tan bien está funcionando el modelo.

#### *Desventajas de MAE:*

A veces, los grandes errores que provienen de los valores atípicos terminan siendo tratados como errores pequeños.

MAE sigue una medida de precisión dependiente de la escala en la que utiliza la misma escala que los datos que se están midiendo. Por lo tanto, no se puede usar para comparar series usando diferentes medidas.

Una de las principales desventajas de MAE es que no es diferenciable en cero. Muchos algoritmos de optimización tienden a utilizar la diferenciación para encontrar el valor óptimo de los parámetros en la métrica de evaluación.

Puede ser un desafío calcular gradientes en MAE.

### Error Cuadrático Medio (MSE)

MSE es como una medida combinada de sesgo y varianza de su predicción, es decir,  $MSE = \text{Sesgo}^2 + \text{Varianza}$

$$\frac{1}{n} \sum_i^n (y_i - \hat{y}_i)^2$$

*Ventajas de MSE:*

Los valores de MSE se expresan en ecuaciones cuadráticas. Por lo tanto, cuando lo trazamos, obtenemos un descenso de gradiente con solo un mínimo global.

Para errores pequeños, converge a los mínimos de manera eficiente. No hay mínimos locales.

MSE penaliza el modelo por tener grandes errores al elevarlos al cuadrado.

Es particularmente útil para eliminar valores atípicos con grandes errores del modelo al ponerles más peso.

*Desventajas MSE:*

Una de las ventajas de MSE se convierte en desventaja cuando hay una mala predicción. La sensibilidad a los valores atípicos magnifica los errores elevados al elevarlos al cuadrado.

MSE tendrá el mismo efecto para un solo error grande que para muchos errores pequeños. Pero sobre todo buscaremos un modelo que funcione lo suficientemente bien a nivel general.

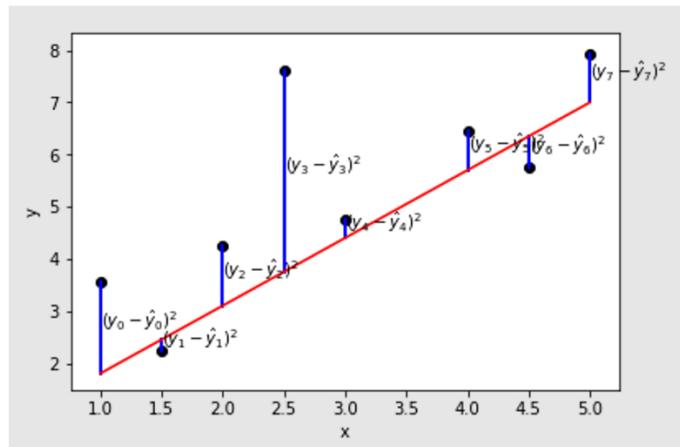
MSE depende de la escala, ya que su escala depende de la escala de los datos. Esto lo hace altamente indeseable para comparar diferentes medidas.

Cuando se introduce un nuevo valor atípico en los datos, el modelo intentará tomar el valor atípico. Al hacerlo, producirá una línea diferente de mejor ajuste que puede causar que los resultados finales sean sesgados.

### Raíz del error cuadrático medio (RMSE)

RMSE se refiere a Root MSE, tomar una raíz de MSE devolverá la unidad real, fácil de interpretar la precisión de su modelo.

Dado que los errores se elevan al cuadrado antes de promediarlos, el RMSE otorga un peso relativamente alto a los errores grandes. Esto significa que el RMSE es más útil cuando los errores grandes son particularmente indeseables.



$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

#### Ventajas de RMSE:

RMSE es fácil de entender.

Sirve como una heurística para entrenar modelos.

Es computacionalmente simple y fácilmente diferenciable, lo que desean muchos algoritmos de optimización.

RMSE no penaliza los errores tanto como lo hace MSE debido a la raíz cuadrada.

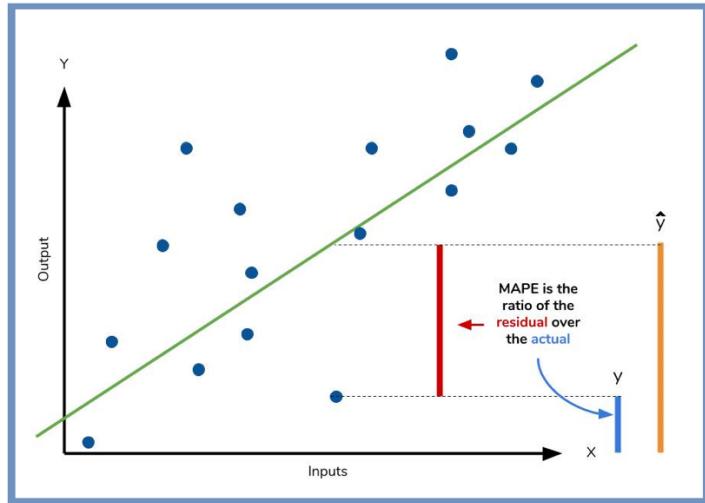
#### Desventajas de RMSE:

Al igual que MSE, RMSE depende de la escala de los datos. Aumenta en magnitud si aumenta la escala del error.

Un inconveniente importante de RMSE es su sensibilidad a los valores atípicos por lo que los valores atípicos deben eliminarse para que funcione correctamente.

RMSE aumenta con un aumento en el tamaño de la muestra de prueba. Este es un problema cuando calculamos los resultados en diferentes muestras de prueba.

## Mean absolute percentage error (MAPE)



$$MAPE = \frac{100\%}{n} \sum \left| \frac{y - \hat{y}}{y} \right|$$

Similar a MAE, pero normalizado por true observation ( $y$ ). La desventaja es que cuando la true observation es cero, esta métrica será problemática.

*Ventajas de MAPE:*

MAPE es independiente de la escala de las variables ya que sus estimaciones de error son porcentuales.

Todos los errores están normalizados en una escala común y es fácil de entender.

Como MAPE utiliza errores porcentuales absolutos, se evita el problema de que los valores positivos y los valores negativos se anulen entre sí.

*Desventajas de MAPE:*

Un inconveniente principal de MAPE es cuando el valor del denominador encuentra cero. Nos encontramos ante el problema de la “división por cero” ya que no está definido.

MAPE penaliza más los errores negativos que los errores positivos. Por lo tanto, está sesgado cuando comparamos la precisión de los métodos de predicción, ya que elegirá un método cuyos valores sean demasiado bajos de forma predeterminada.

## Métrica $R^2$

El coeficiente de determinación es un número entre 0 y 1 que mide qué tan bien un modelo estadístico predice un resultado. Más técnicamente,  $R^2$  es una medida de bondad de ajuste. Es la proporción de la varianza en la variable dependiente que es explicada por el modelo.

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}}$$

$$= 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

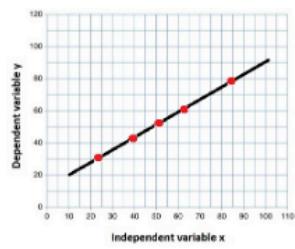
La regresión de suma al cuadrado es la suma de los residuos al cuadrado, y la suma total de los cuadrados es la suma del cuadrado de las distancias a la que se encuentran los datos de la media . Al ser un porcentaje tomara valores entre 0 y 1.

### $R^2$ Values

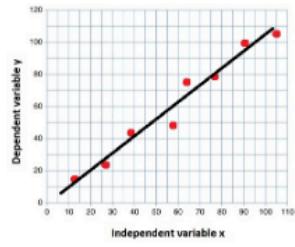
### Interpretation

### Graph

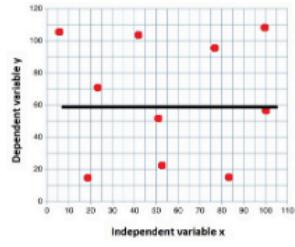
$R^2 = 1$  All the variation in the  $y$  values is accounted for by the  $x$  values



$R^2 = 0.83$  83% of the variation in the  $y$  values is accounted for by the  $x$  values



$R^2 = 0$  None of the variation in the  $y$  values is accounted for by the  $x$  values



## Métricas de performance para clasificación



Los problemas de clasificación son una de las áreas más investigadas del mundo. Los casos de uso están presentes en casi todos los entornos industriales y de producción. Reconocimiento de voz, reconocimiento facial, clasificación de texto: la lista es interminable.

Los modelos de clasificación tienen una salida discreta, por lo que necesitamos una métrica que compare clases discretas de alguna forma. Las métricas de clasificación evalúan el rendimiento de un modelo y te dicen qué tan buena o mala es la clasificación, pero cada una de ellas lo evalúa de manera diferente.

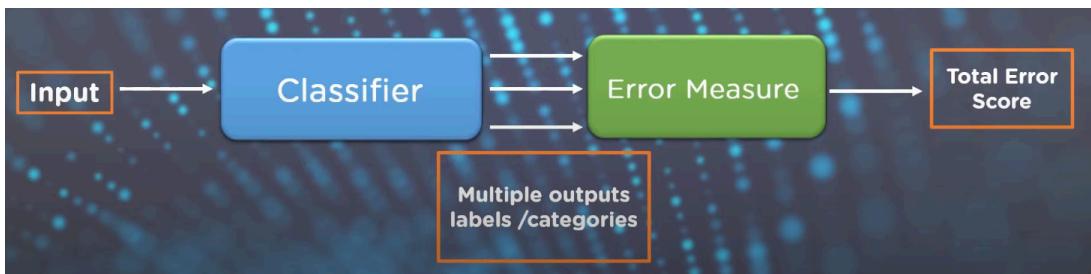
Para evaluar los modelos de clasificación, discutiremos estas métricas en detalle:

- ✓ Exactitud (Accuracy)
- ✓ Matriz de confusión
- ✓ Precisión
- ✓ Recall
- ✓ Puntuaje F1 (F1-Score)
- ✓ ROC-AUC

## ANEXOS

### Matriz de confusión

Los modelos de clasificación tienen múltiples categorías de salida. La mayoría de las métricas de error nos indicarán el error total de un modelo, pero a partir de eso no podremos averiguar errores individuales en nuestro modelo.



Una matriz de confusión es una tabla con las diferentes salidas predichas en un problema de clasificación que nos ayuda a visualizar los resultados de una manera más clara.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Supongamos que deseamos predecir cuántas personas están infectadas con un virus peligroso antes de que muestren los síntomas y en base a eso aislarlos de la población sana.

Nuestro conjunto de datos es un ejemplo de un conjunto de datos no balanceados (imbalanced dataset). Hay 947 puntos de datos para la clase negativa y 3 puntos de datos para la clase positiva.

ID	Actual Sick?	Predicted Sick?	Outcome
1	1		1 TP
2	0		0 TN
3	0		0 TN
4	1		1 TP
5	0		0 TN
6	0		0 TN
7	1		0 FP
8	0		1 FN
9	0		0 TN
10	1		0 FP
:	:	:	:
1000	0		0 FN

Así es como calcularemos la exactitud:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

De la tabla obtenemos TP = 30, TN = 930, FP = 30, FN = 10, calculamos la exactitud:

$$Accuracy = \frac{30 + 930}{30 + 30 + 930 + 10} = 0.96$$

El 96% a primera vista parece deciros que porcentaje de la gente se enfermara.

Si analizamos bien el resultado, el modelo está prediciendo que porcentaje de la gente no se enfermara.

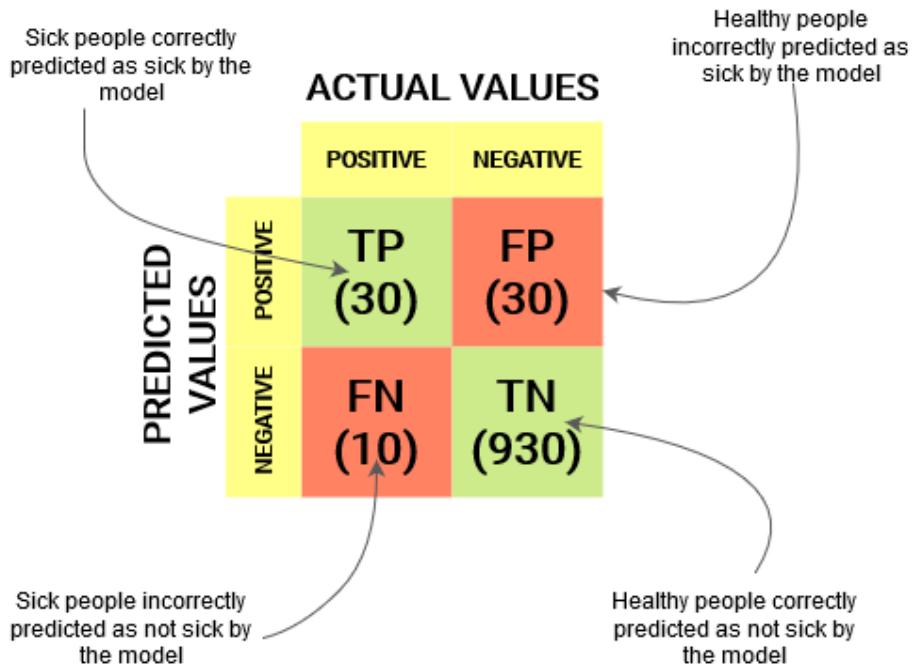
Para analizar mejor esto introducimos los conceptos de Precision y Recall

**Precision**, nos dice cuántos de los casos predichos como positivos resultaron ser positivos realmente.

$$Precision = \frac{TP}{TP + FP}$$

**Recall**, nos dice cuántos de los casos positivos reales pudimos predecir correctamente con nuestro modelo.

$$Recall = \frac{TP}{TP + FN}$$



$$\text{Precision} = \frac{30}{30 + 30} = 0.5$$

$$\text{Recall} = \frac{30}{30 + 10} = 0.75$$

En nuestro ejemplo, Recall sería una mejor métrica porque no queremos dar de alta accidentalmente a una persona infectada y dejar que se mezcle con la población sana, propagando así el virus contagioso.

Recall es importante en los casos médicos en los que no importa si activamos una falsa alarma, pero los casos positivos reales no deben pasar desapercibidos!

La precisión es importante en los sistemas de recomendación de música o video, sitios web de comercio electrónico, etc. Los resultados incorrectos pueden provocar la pérdida de clientes y ser perjudiciales para el negocio.

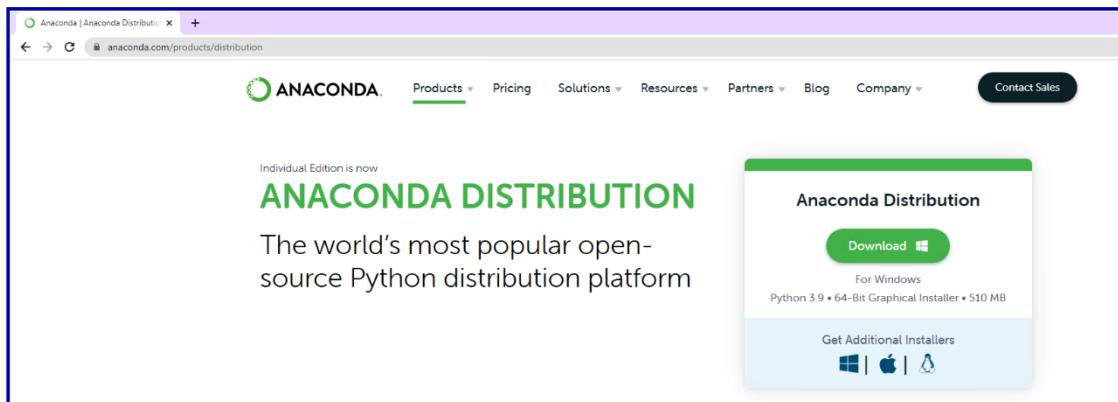
En los casos que no está claro cuál de los dos es más importante procedemos a combinarlos:

$$F1 - score = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$$

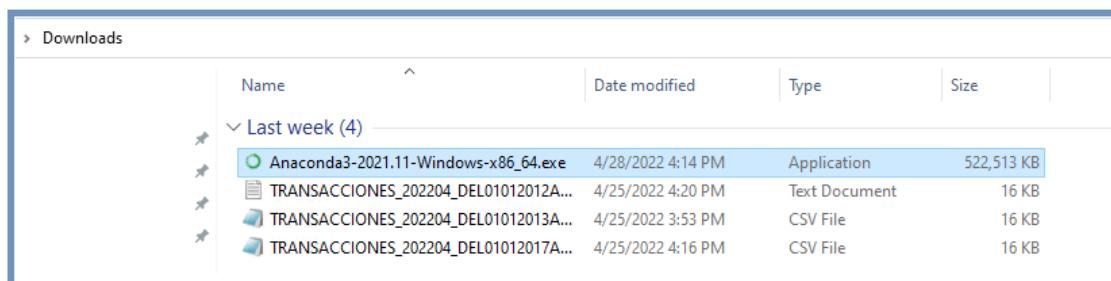
## Instalación de Anaconda en Windows

- Anaconda es una distribución de los lenguajes de programación Python y R para computación científica (ciencia de datos, aplicaciones de Machine Learning, procesamiento de datos a gran escala, análisis predictivo, etc.).
- Tiene como ventaja simplificar la gestión e implementación de paquetes. La distribución incluye paquetes de “data science” adecuados para Windows, Linux y macOS.
- Para instalar Anaconda ingresar a la siguiente página:

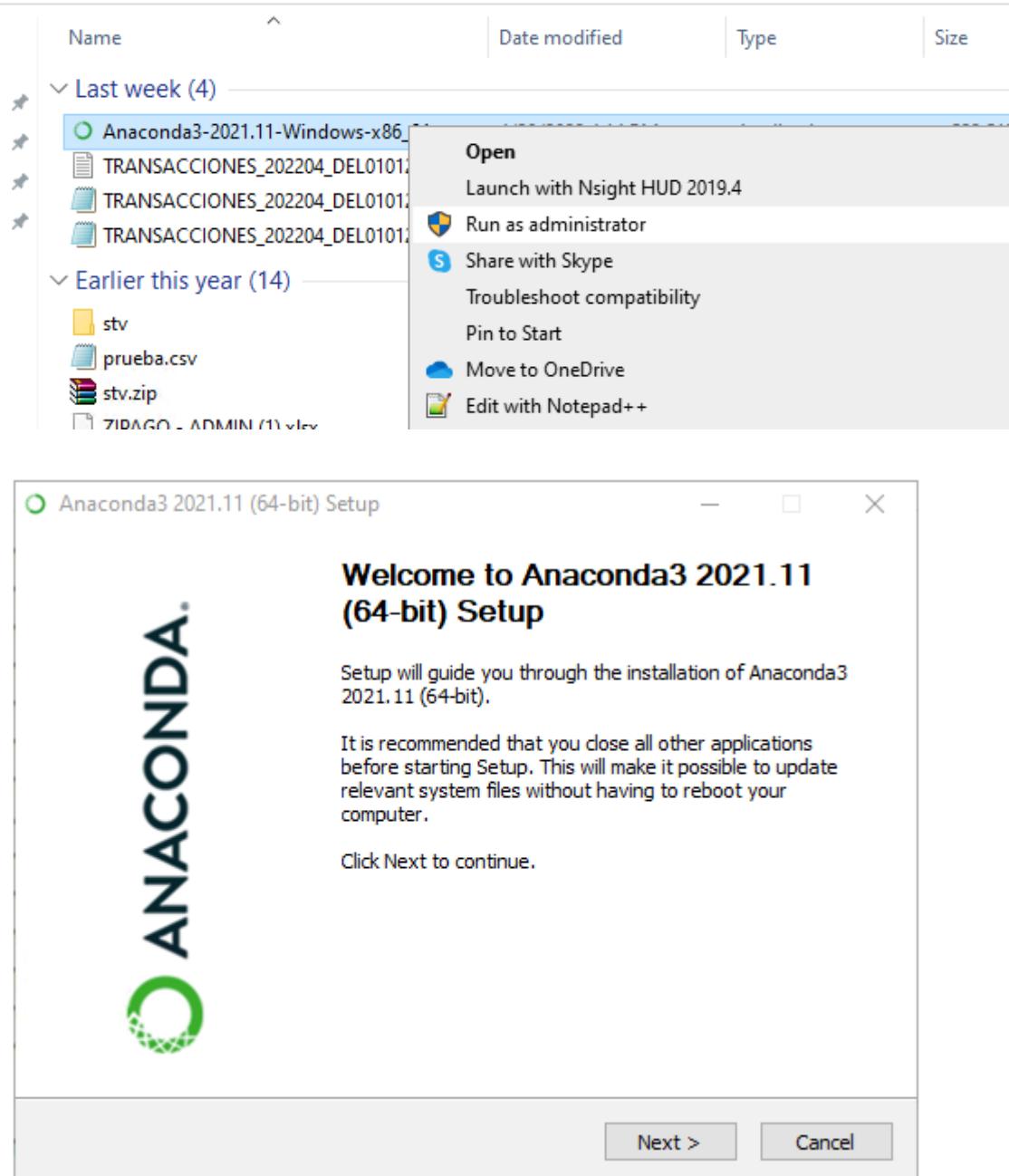
<https://www.anaconda.com/products/distribution>



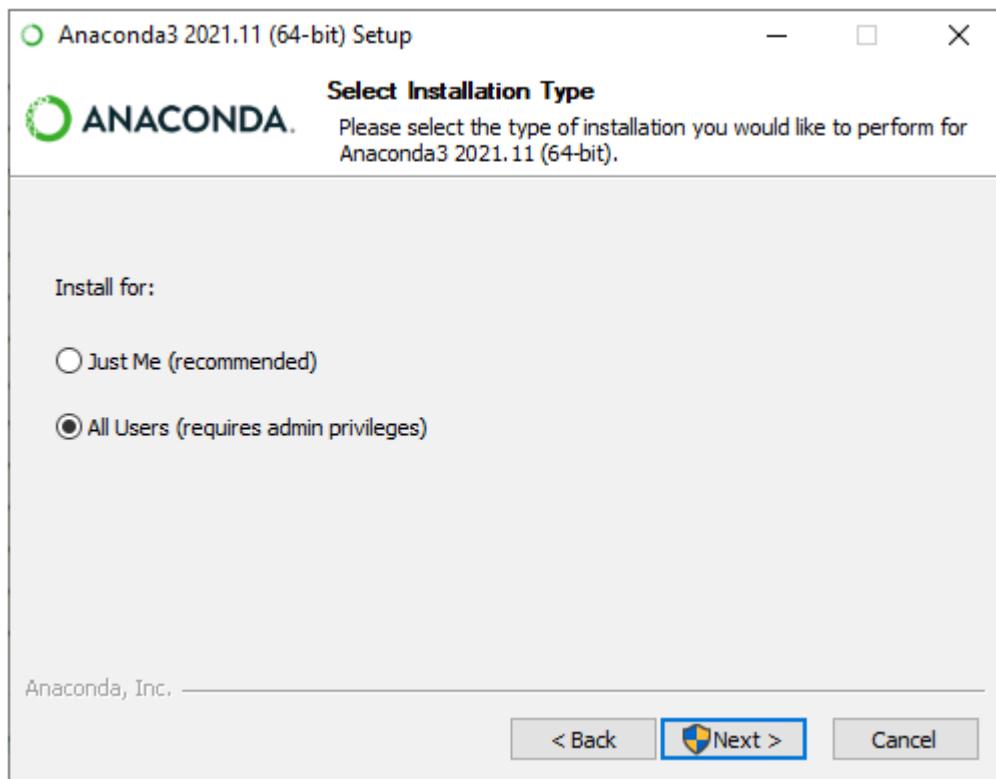
- Descargar el instalador presionando en **Download**:



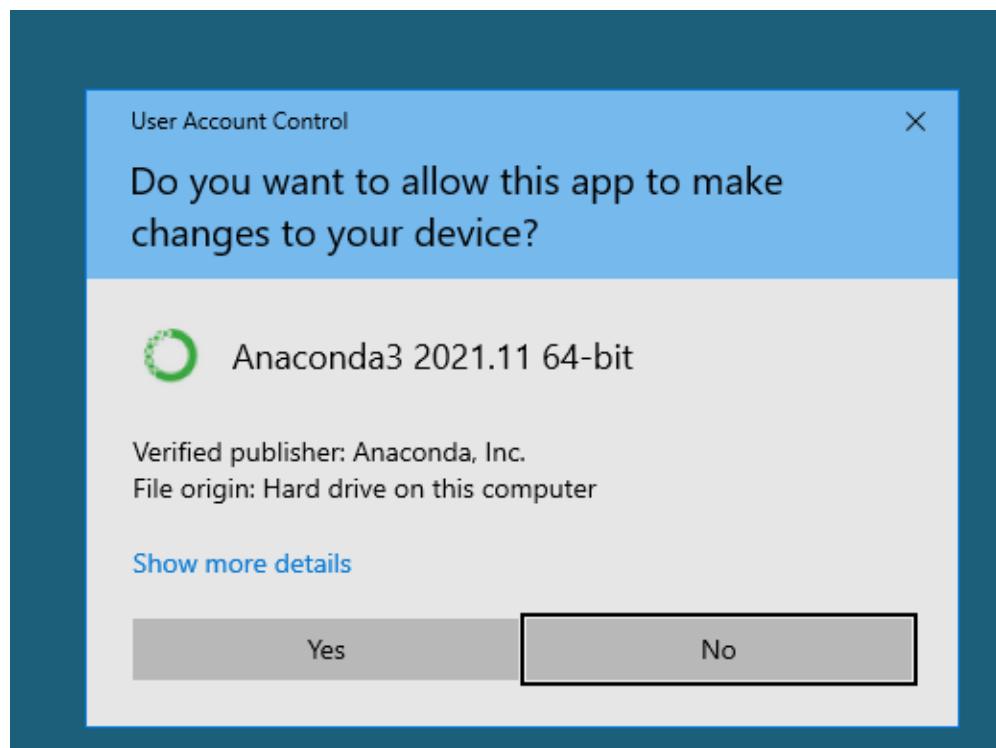
Una vez descargado ejecutar el instalador como usuario administrador:



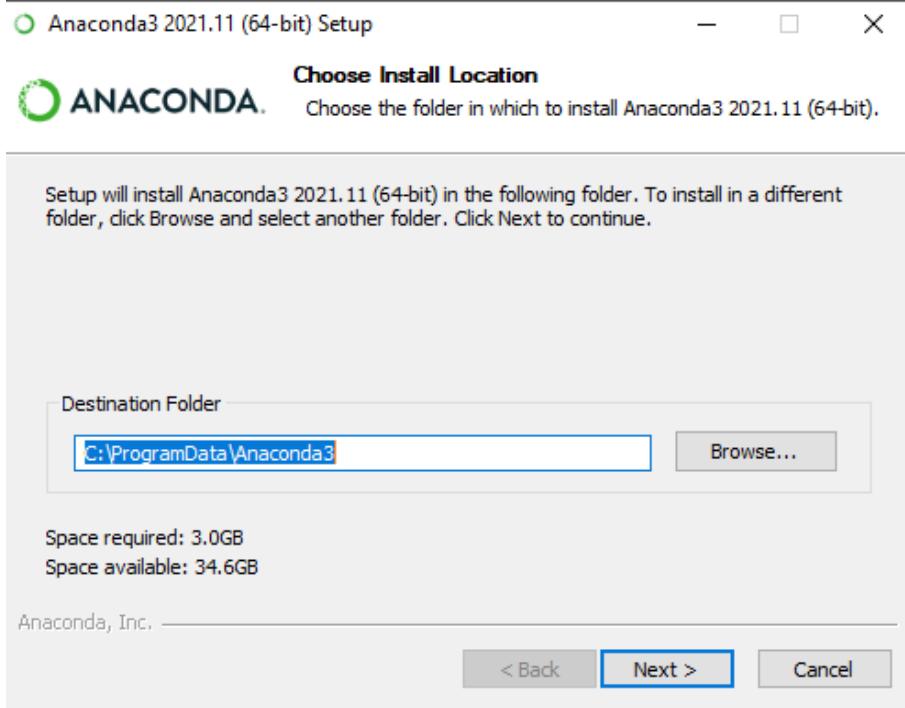
Presionar Next



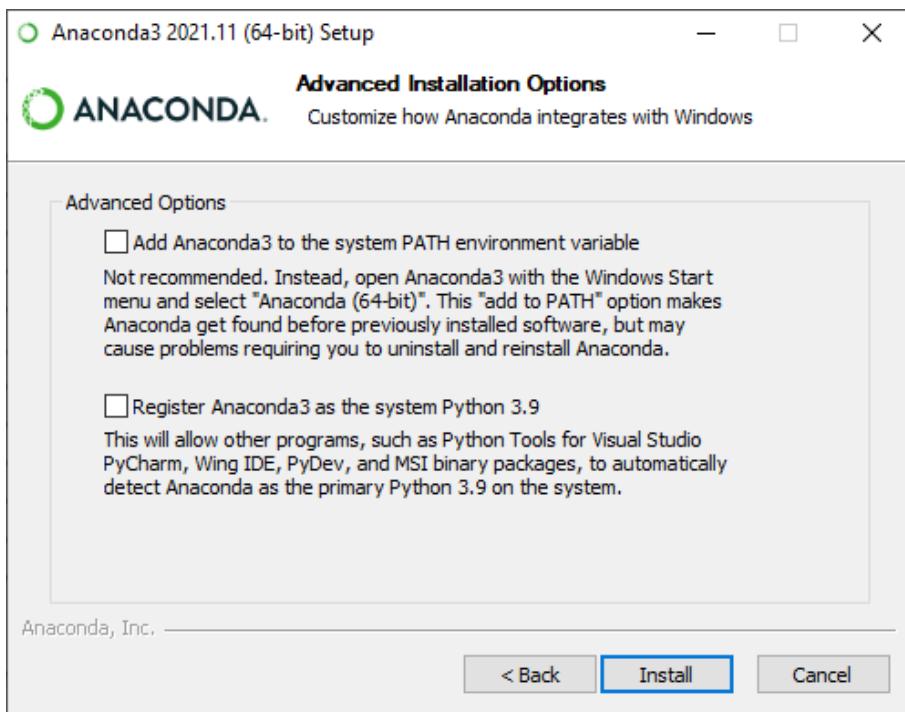
Presionar Next



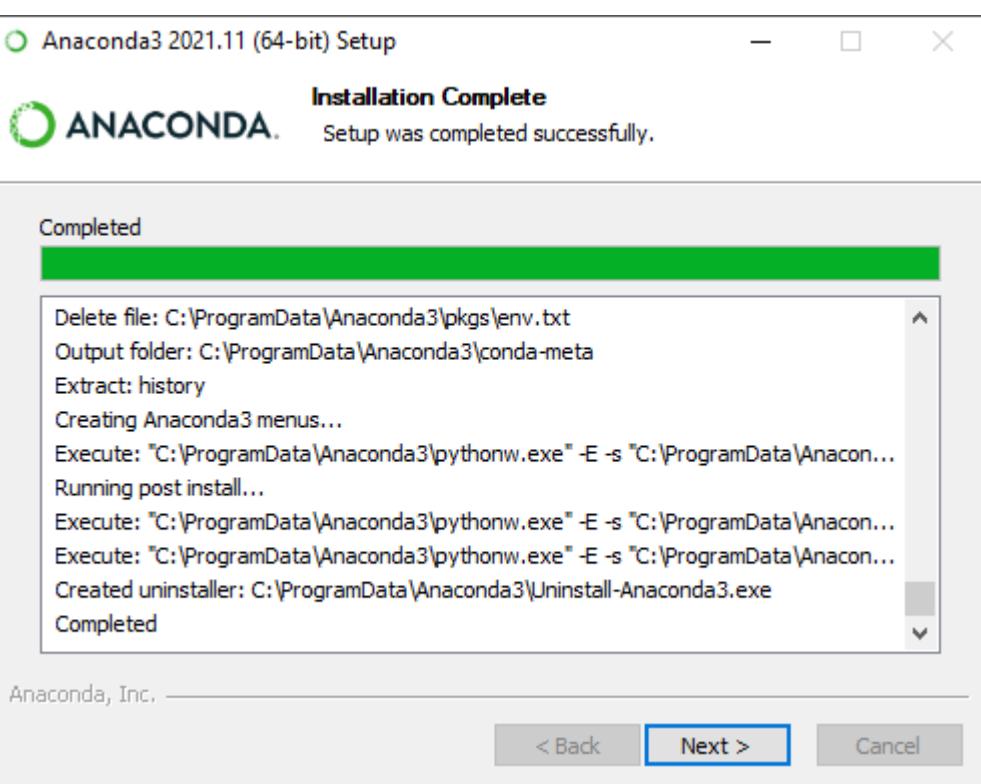
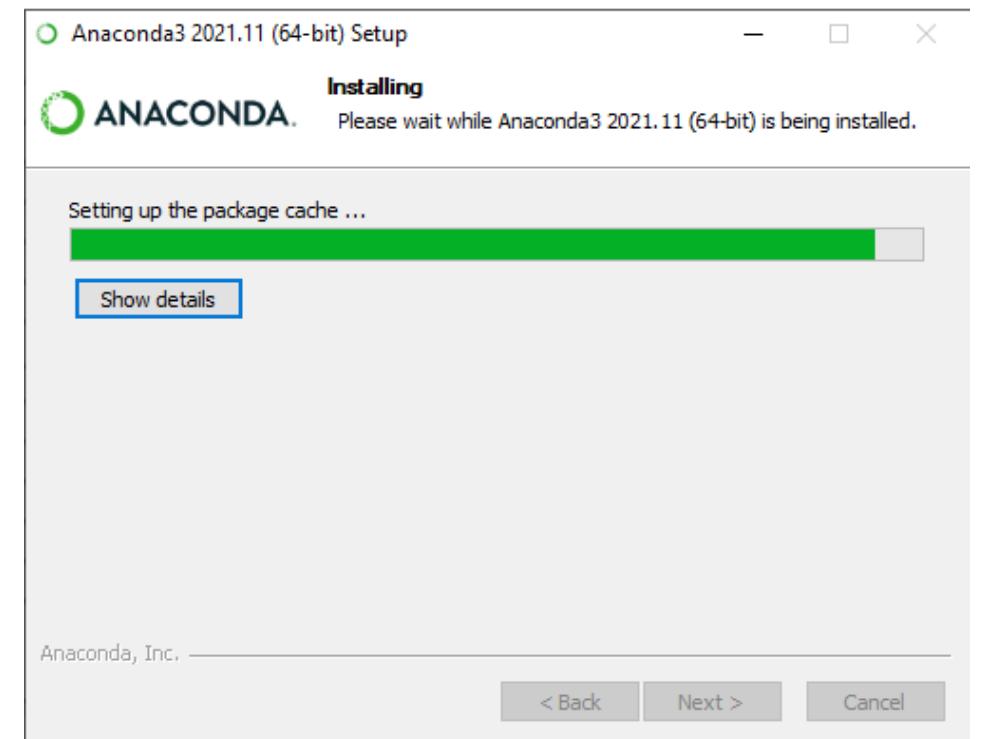
Presionar Yes



Presionar Next



Presionar Install



Presionar Next

○ Anaconda3 2021.11 (64-bit) Setup — □ ×

 **Anaconda3 2021.11 (64-bit)**  
Anaconda + JetBrains

Working with Python and Jupyter notebooks is a breeze with PyCharm Pro, designed to be used with Anaconda. Download now and have the best data tools at your fingertips.

<https://www.anaconda.com/pycharm>



Anaconda, Inc. —

< Back

Next >

Cancel

Presionar Next

○ Anaconda3 2021.11 (64-bit) Setup — □ ×

**Completing Anaconda3 2021.11 (64-bit) Setup**

Thank you for installing Anaconda Individual Edition.

Here are some helpful tips and resources to get you started.  
We recommend you bookmark these links so you can refer back to them later.

Anaconda Individual Edition Tutorial

Getting Started with Anaconda



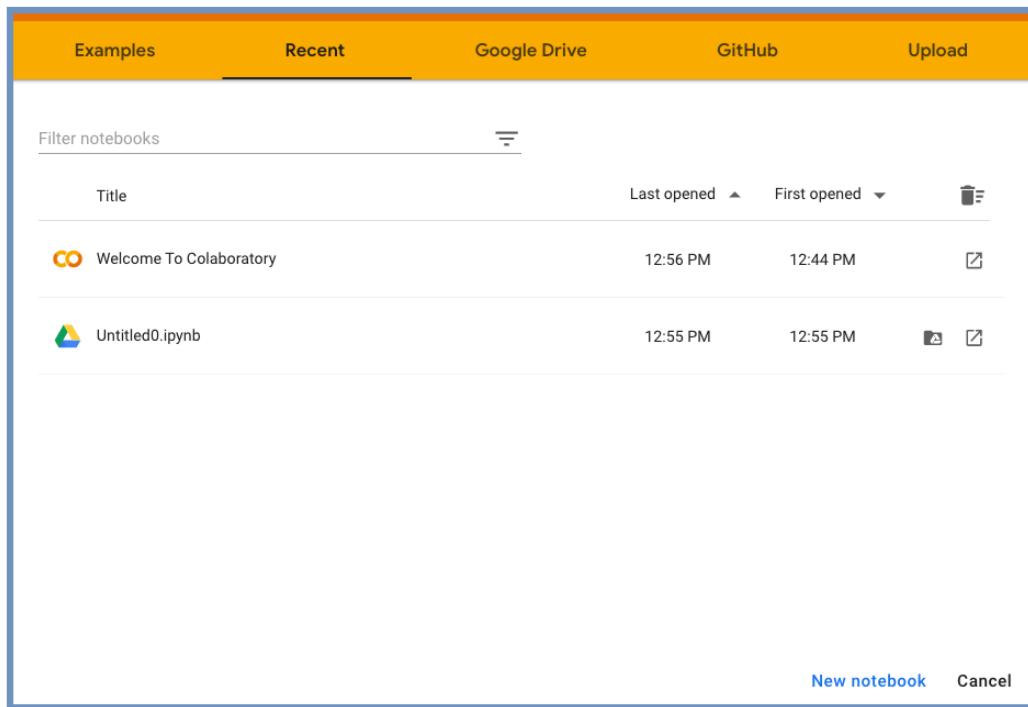
< Back

Finish

Cancel

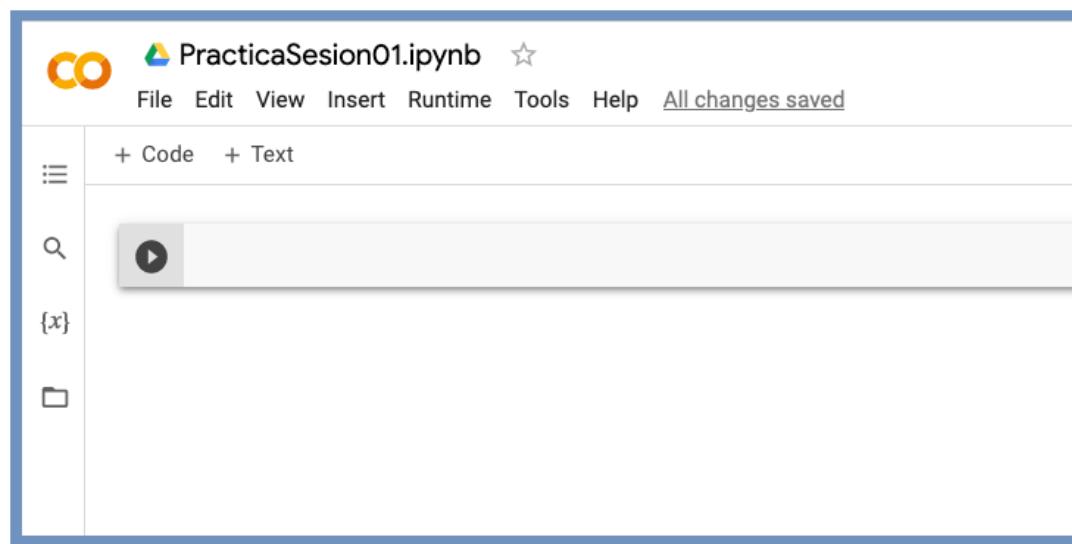
## Usando Google Colab

- Para utilizar google colab primero debemos estar logueados a nuestra cuenta de Gmail
- Ingresar a la ruta: <https://colab.research.google.com/>

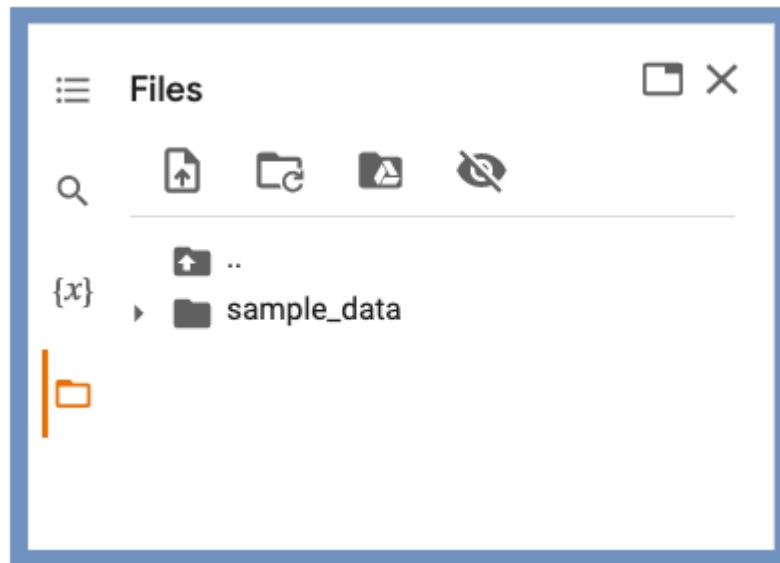


Presionar **New Notebook**

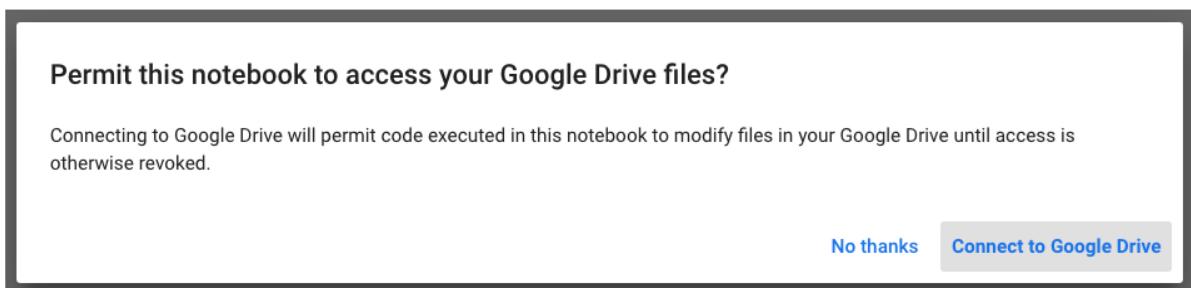
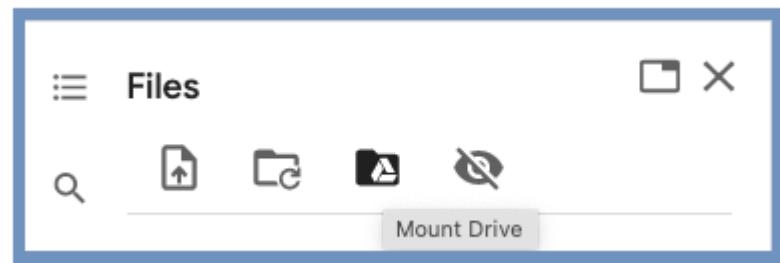
Se abrirá un nuevo IPython Note Book más conocido como Jupyter Notebook, un entorno computacional interactivo, en el que puede combinar ejecución de código, texto enriquecido, matemáticas, gráficos etc. Podemos modificar el nombre a nuestro gusto.

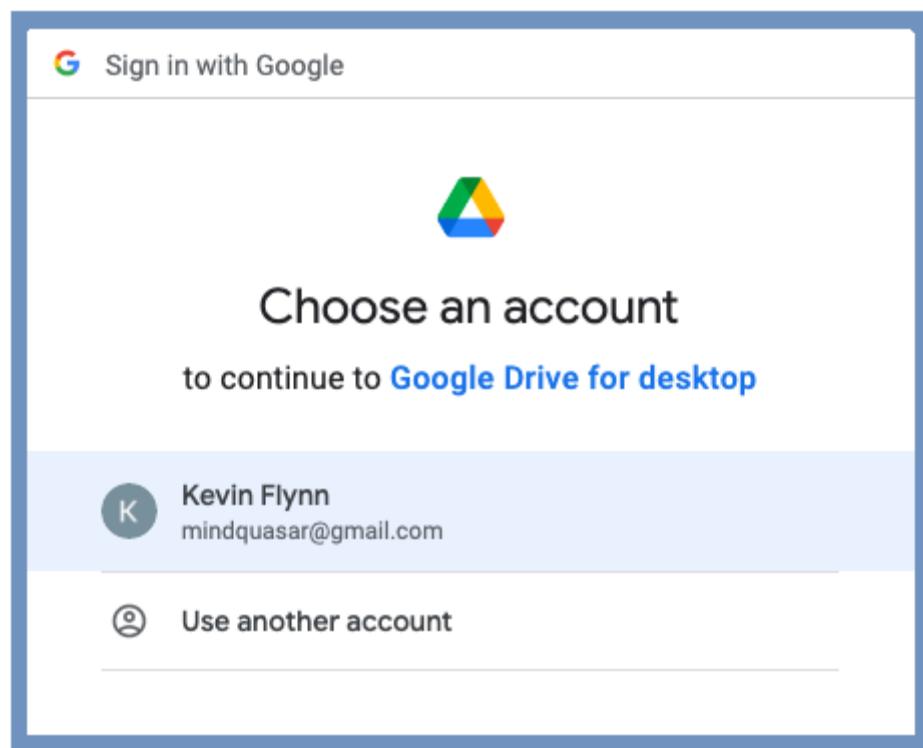


- Seleccionemos el folder en el menú lateral izquierdo y podremos navegar en los directorios de nuestra computadora.

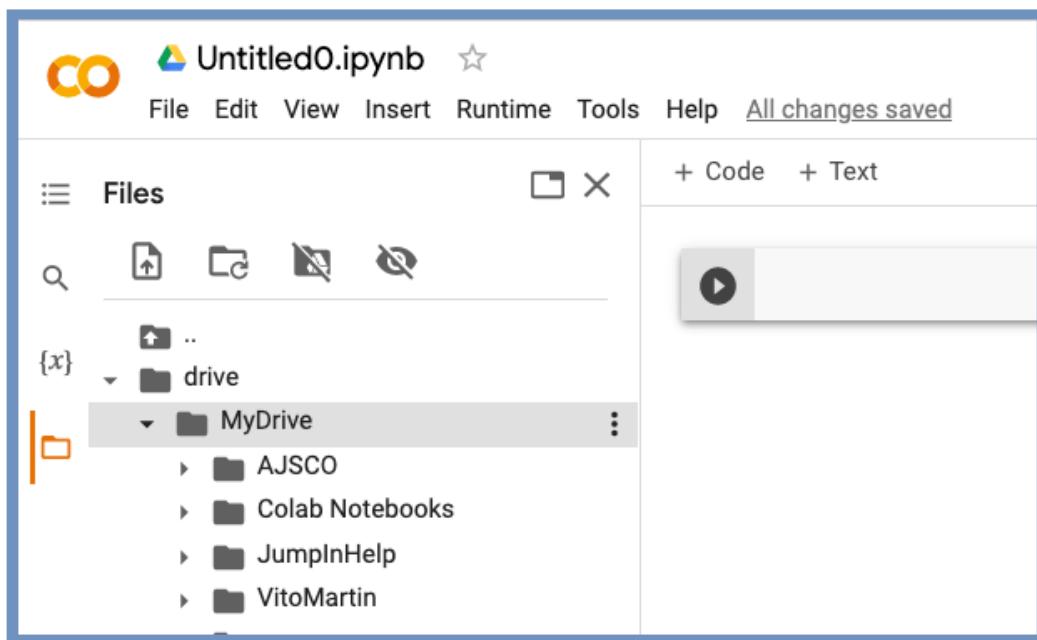


- Seleccionemos el folder con el ícono de reciclaje en el menú lateral superior y podremos montar nuestro google drive.



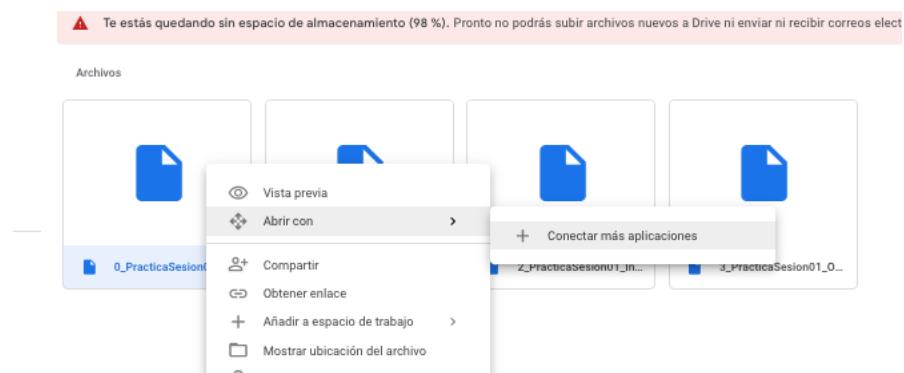
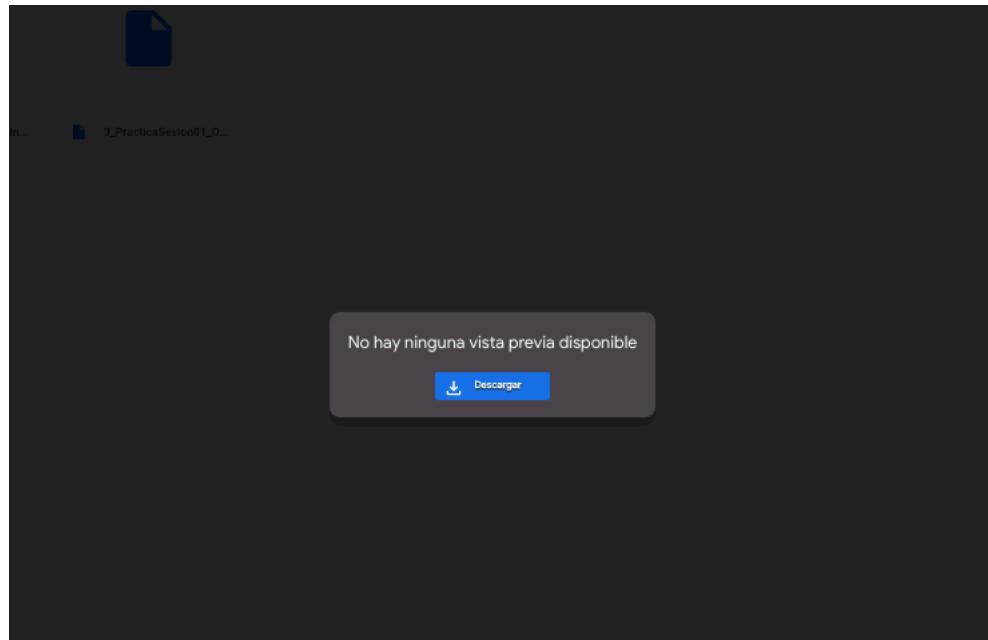


- Una vez montado nuestro google drive podemos tener acceso a nuestros archivos que están en la nube de google.



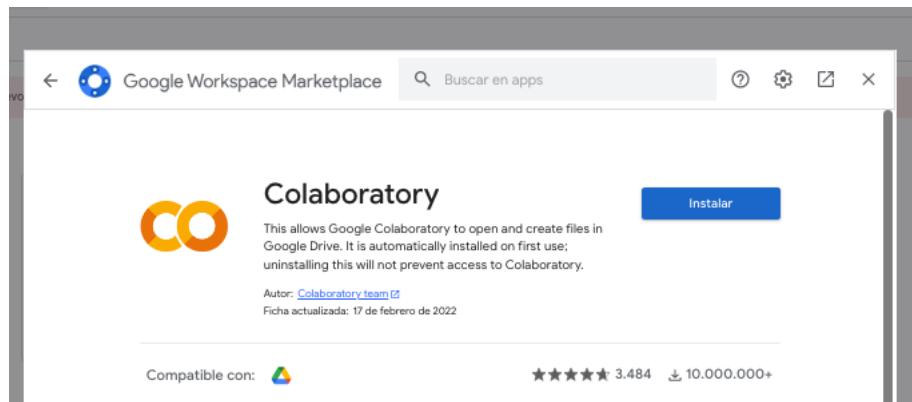
Ahora estamos listos para codificar.

Si les sale el siguiente mensaje:

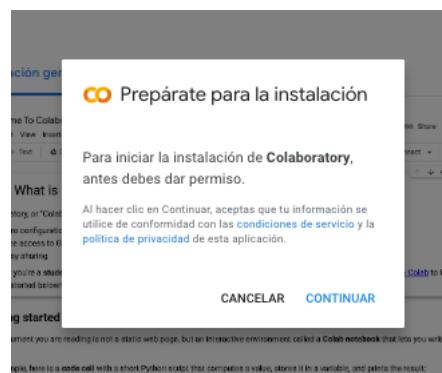


### Selección Colaboratory

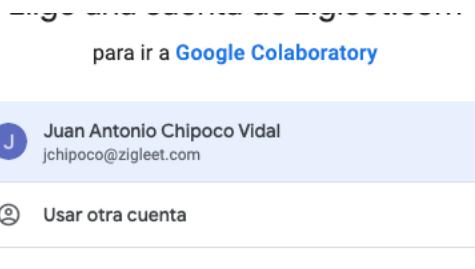




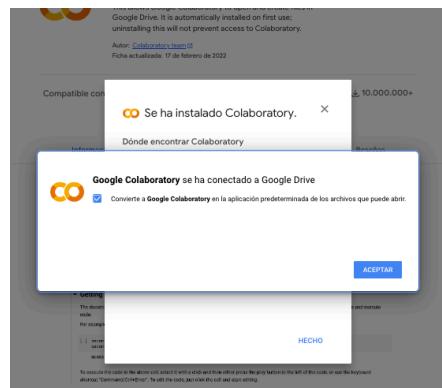
Presionar instalar



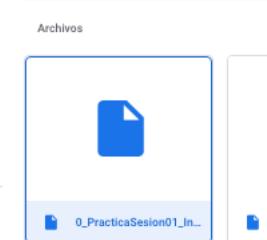
Presionar continuar



Seleccionamos nuestra cuenta gmail

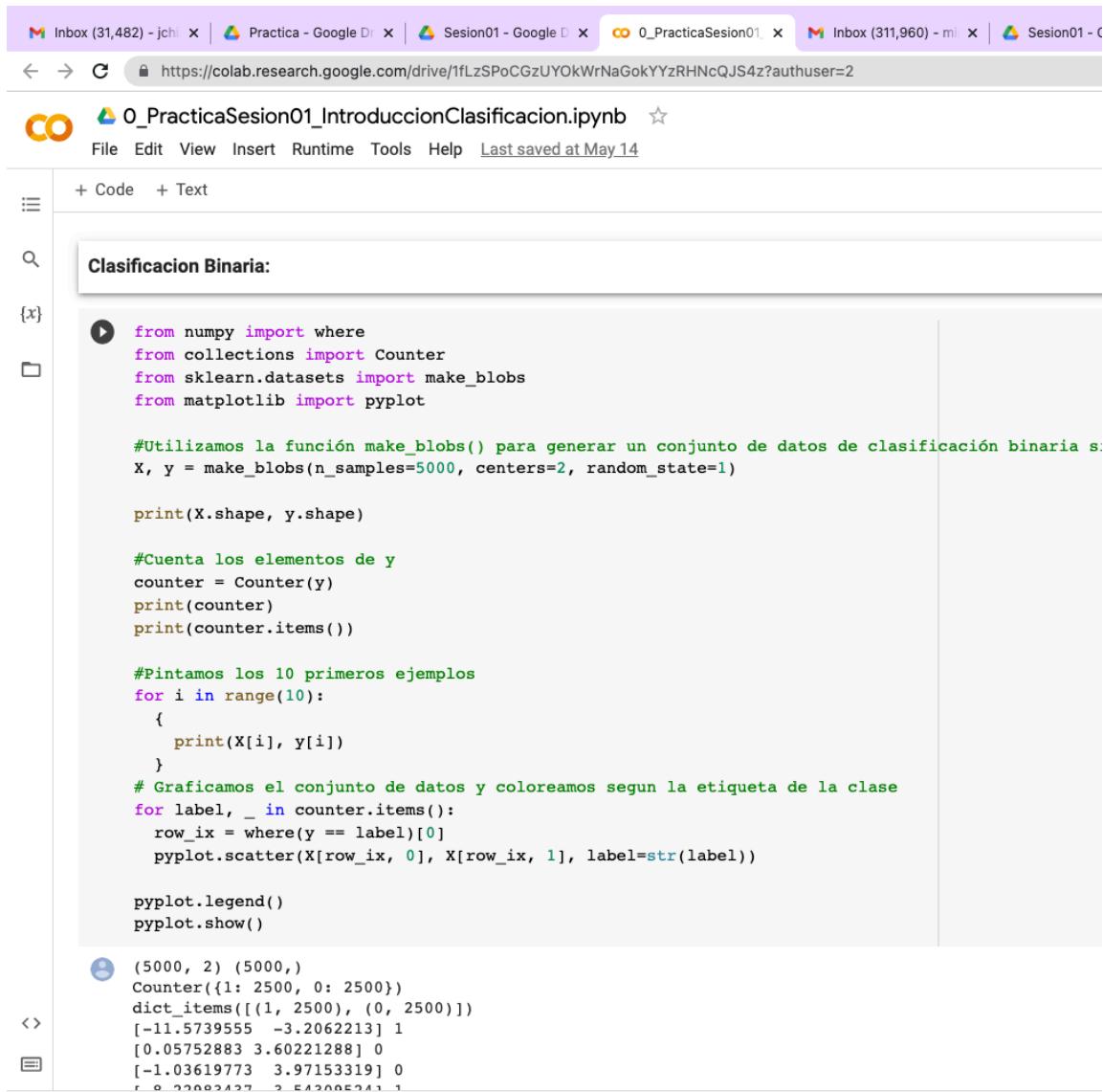


Presionar aceptar



Hacemos doble click sobre nuestro jupyter notebook

Ya podremos ejecutar las fuentes



```

O_PracticaSesion01_IntroduccionClasificacion.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at May 14

+ Code + Text

Clasificacion Binaria:

from numpy import where
from collections import Counter
from sklearn.datasets import make_blobs
from matplotlib import pyplot

#Utilizamos la función make_blobs() para generar un conjunto de datos de clasificación binaria s:
X, y = make_blobs(n_samples=5000, centers=2, random_state=1)

print(X.shape, y.shape)

#Cuenta los elementos de y
counter = Counter(y)
print(counter)
print(counter.items())

#Pintamos los 10 primeros ejemplos
for i in range(10):
    print(X[i], y[i])

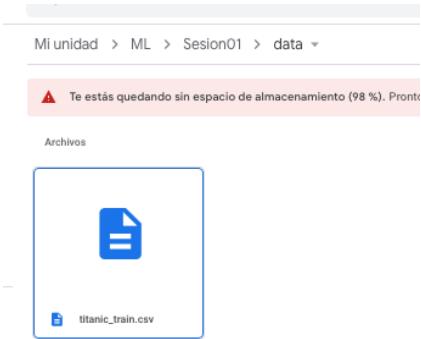
# Graficamos el conjunto de datos y coloreamos según la etiqueta de la clase
for label, _ in counter.items():
    row_ix = where(y == label)[0]
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1], label=str(label))

pyplot.legend()
pyplot.show()

(5000, 2) (5000,)
Counter({1: 2500, 0: 2500})
dict_items([(1, 2500), (0, 2500)])
[-11.5739555 -3.2062213] 1
[0.05752883 3.60221288] 0
[-1.03619773 3.97153319] 0
[ 0.22003427 3.54300524] 1

```

Si nuestro colab usa algún csv debemos crear una carpeta en nuestro google drive donde lo colocaremos:

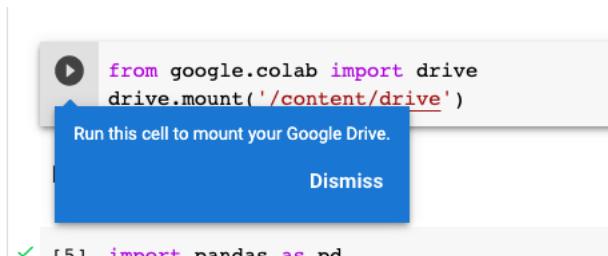


A continuación, en nuestro colab debemos montar nuestro google drive presionando la carpeta gris con el símbolo de reciclaje:

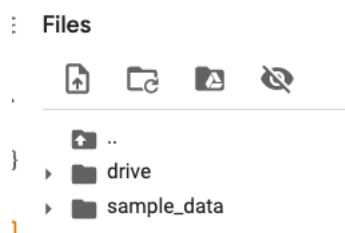


ve  
mple\_data

Aparecerá el siguiente mensaje:



Ejecutamos la celda presionando el botón play y veremos que aparece la carpeta drive que nos da acceso a nuestro google drive:



En nuestro colab para acceder a nuestro csv debemos modificar la ruta con nuestra propia ruta:

```
[6] titanic_df_train = pd.read_csv('/content/drive/MyDrive/ML/Sesion01/data/titanic_train.csv')
```