Machine Learning en Python

Introducción

En este curso, aprenderás cómo se utiliza Machine Learning en varios campos e industrias clave. Por ejemplo, en la industria del cuidado de la salud, los científicos de datos utilizan Machine Learning para predecir si una célula humana que se cree está en riesgo de desarrollar cáncer, es benigno o maligna.

Como tal, Machine Learning puede desempeñar un papel clave al determinar el estado de salud y bienestar de una persona.

También aprenderás sobre el valor de los árboles de decisión y cómo construir un buen árbol de decisiones de los datos históricos que ayuda a los médicos a prescribir la medicina adecuada para cada uno de sus pacientes.

Aprenderás cómo los banqueros usan Machine Learning para tomar decisiones sobre si aprobar o no una solicitud de un préstamo. Y aprenderás a utilizar Machine Learning para realizar la segmentación de clientes en un banco, donde normalmente no es fácil de ejecutar debido al gran volúmen de datos variables.

En este curso, verán cómo Machine Learning ayuda a sitios web como YouTube, Amazon o Netflix sugerir recomendaciones a sus clientes sobre varios productos o servicios, tales como cuales películas podrían estar interesados para ir a ver o qué libros comprar.
¡Hay tanto que puedes hacer con Machine Learning!

Aquí, aprenderás a como utilizar bibliotecas populares de python para crear tu modelo. Por ejemplo, dado un conjunto de set de datos de automóviles, utilizamos la biblioteca de sci-kit learn (sklearn) para estimar la emisión de CO2 de los automóviles, utilizando el tamaño de su motor o cilindros. Incluso podemos predecir cuáles serán las emisiones de CO2 de un automóvil que ni siquiera ha sido producido todavía!

Y veremos cómo el sector de la industria de las telecomunicaciones puede predecir el abandono de clientes. Puedes ejecutar y practicar el código de todos estos ejemplos utilizando el entorno del laboratorio incorporado en este curso.

No tienes que instalar nada en tu ordenador o hacer algo en la nube. Todo lo que tiene que hacer es hacer clic en un botón para iniciar el entorno de laboratorio en tu navegador de internet.
El código para las ejemplos ya está escrito usando el lenguaje de programación python, en Jupyter notebooks, y puedes ejecutarlo para ver los resultados, o cambiarlo para que entiendas mejor los algoritmos.

Entonces, ¿Qué serás capaz de lograr al tomar este curso? Bueno, invirtiendo unas pocas horas a la semana en las próximas semanas, vas a adquirir nuevas habilidades para añadir a tu cv, tales como regresión, clasificación, agrupamiento, sci-kit learn y SciPy.

También obtendrás nuevos proyectos que puedes agregar a tu portafolio, incluyendo la detección de cáncer, predicción de tendencias económicas, predicción de abandono de clientes, motores de recomendación y muchos más.

También obtendrá un certificado en Machine Learning para probar su competencia, y compartirlo en cualquier lugar que te guste en línea o fuera de línea, tales como perfiles de LinkedIn y redes sociales.

Objetivos de aprendizaje

En este curso aprenderás:

- A Relacionar el Modelado Estadístico con Machine Learning y hacer una comparación entre ellos..
- Ejemplos de la vida real respecto de Machine learning y la forma en que afecta a la sociedad de maneras impensadas!
- En los laboratorios: Utilizar las librerías de Python para Machine Learning, tales como scikit-learn.

Explorar algoritmos y modelos:

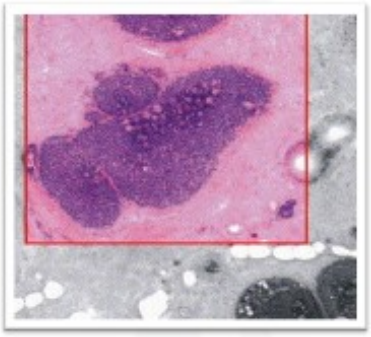
- Algoritmos populares: Regresión, Clasificación, y Clustering
- Sistemas Recomendadores: Basados en Contenido y Content-Based y Filtrado Colaborativo
- Modelos populares: Entrenar/Probar, Descenso por Gradiente, y Error Cuadrático Medio

Módulo 1: Machine Learning

Introducción a Machine Learning

En esta lección, aprenderás sobre:

- Aplicaciones de Machine Learning
- Librerías Python para Machine Learning
- Aprendizaje Supervisado y No Supervisado



Esta es una muestra de célula humana extraída de un paciente. Y esta célula tiene características … por ejemplo, su espesor de aglutinación es 6, la uniformidad de tamaño celular es 1, su adhesión marginal es 1, y así.

Una de las preguntas más interesantes que podemos hacer, en este punto es: "¿Es una célula benigna o maligna?" En contraste con un tumor benigno, un tumor maligno es un tumor que puede invadir el tejido adyacente o diseminarse alrededor del cuerpo, y diagnosticarlo de manera temprana puede ser clave para la supervivencia del paciente.

Fácilmente se podría suponer que sólo un médico con años de experiencia podría diagnosticar ese tumor y determinar si el paciente está desarrollando cáncer o no.
¿Cierto?

Bueno, imagine que ha obtenido un conjunto de datos que contiene características de miles de

muestras de células humanas extraídas de pacientes que se creía que estaban en riesgo de desarrollar cáncer.
El análisis de los datos originales mostró que muchas de las características diferían significativamente entre muestras benignas y malignas. Puedes utilizar los valores de estas características celulares en muestras de otros pacientes para dar una indicación anticipada de si una nueva muestra puede ser benigna o maligna.

Debes limpiar los datos, seleccionar un algoritmo adecuado para construir un modelo de predicción, y entrenar tu modelo para entender los patrones de células benignas o malignas dentro de los datos. Una vez que el modelo ha sido entrenado por medio de datos iterativamente, se puede utilizar para predecir tu célula nueva o desconocida con una precisión bastante alta.
¡Esto es machine learning!

Es la forma en que un modelo de machine learning puede hacer la tarea de un médico o, al menos, ayudarlo para hacer el proceso más rápido. Ahora, permítame darle una definición formal de machine learning.

Machine learning es el subcampo de la ciencia de la computación que "da a las computadoras la habilidad de aprender sin ser programadas explícitamente.”

Permítanme explicar lo que quiero decir cuando digo "sin ser programadas explícitamente."
Supongan que tienen un conjunto de datos de imágenes de animales como, por ejemplo, perros y gatos, y quieran una aplicación o software sea capaz de reconocerlos y diferenciarlos.

Lo primero que tienen que hacer es interpretar las imágenes como un conjunto de conjuntos de características.
Por ejemplo, ¿la imagen muestra los ojos del animal? Si es así, ¿cuál es su tamaño? ¿Tiene orejas? ¿Qué hay de una cola? ¿Cuántas piernas? ¿Tiene alas?

Antes de machine learning, cada imagen se transformaría en un vector de características. Entonces, tradicionalmente, tendríamos que escribir algunas reglas o métodos para que las computadoras sean inteligentes y detecten a los animales.

Pero, fue un fracaso.
¿Por qué? Bueno, como podrás adivinar, se necesitaba muchas reglas, muy dependientes del conjunto de datos actual, y no lo suficiente generalizadas como para detectar casos fuera de la muestra.

Este es el momento en que machine learning entró en escena.
Usar machine learning nos permite construir un modelo que observe todos los conjuntos de características, y su correspondiente tipo de animales, y aprender el patrón de cada animal. Se trata de un modelo construido por algoritmos de machine learning.

Detecta sin haber sido programado explícitamente para hacerlo. En esencia, el aprendizaje automático sigue el mismo proceso que un niño de 4 años utiliza para aprender, entender, y diferenciar a los animales.

Así, los algoritmos de machine learning, inspirados por el proceso de aprendizaje humano, iterativamente aprenden de los datos, y permiten que las computadoras encuentren información

oculta. Estos modelos nos ayudan en una variedad de tareas, tales como el reconocimiento de objetos, resúmenes, recomendaciones, y así.

Machine Learning afecta a la sociedad de una forma muy influyente.
He aquí algunos ejemplos de la vida real.
En primer lugar, ¿cómo creen que Netflix y Amazon recomiendan videos, películas y programas de televisión a sus usuarios? Utilizan machine learning para producir sugerencias que podrías disfrutar! Esto es similar a cómo tus amigos te pueden recomendar un programa de televisión, basado en su conocimiento sobre los tipos de programas que te gusta ver.

¿Cómo cree que los bancos toman una decisión al aprobar una solicitud de préstamo? Utilizan machine learning para predecir la probabilidad de impago para cada solicitante, y luego aprueban o rechazar la solicitud de préstamo en base a esa probabilidad.

Las compañías de telecomunicaciones utilizan los datos demográficos de sus clientes para segmentarlos, o predecir si ellos se retirarán de su compañía el próximo mes. Hay muchas otras aplicaciones de machine learning que vemos todos los días en nuestra la vida, como los chatbots, acceder a nuestros teléfonos o incluso los juegos de ordenador usan el reconocimiento facial.

Cada uno utiliza técnicas y algoritmos diferentes de machine learning. Por lo tanto, vamos a examinar rápidamente algunas de las técnicas más populares.

La técnica de **Regresión/Estimación** se utiliza para predecir un valor continuo, por ejemplo, predicción de cosas como el precio de una casa basada en sus características, o para estimar la emisión de CO2 del motor de un coche.

Se utiliza una técnica de **clasificación** para predecir la clase o categoría, por ejemplo, si una célula es benigna o maligna, o si un cliente se va a retirar o no.

Clustering: Los grupos de casos similares, por ejemplo, pueden encontrar pacientes similares, o pueden ser utilizados para la segmentación de clientes en el campo bancario.

La técnica de **asociación** se utiliza para buscar elementos o sucesos que a menudo se producen conjuntamente, por ejemplo, artículos de comestibles que normalmente son comprados conjuntamente por un cliente en particular.

La **detección de anomalías** se utiliza para descubrir casos anormales e inusuales, por ejemplo, se utiliza para la detección de fraude de tarjetas de crédito.

La **minería secuencial** se utiliza para predecir el siguiente suceso, por ejemplo, la secuencia de pulsación en sitios web.

La **reducción de dimensión** (PCA) se utiliza para reducir el tamaño de los datos.

Y finalmente, los **sistemas de recomendación**; esto asocia las preferencias de la gente con otros que tienen gustos similares, y recomienda nuevos artículos para ellos, como libros o películas.

En este punto, estoy bastante seguro de que esta pregunta ha cruzado su mente, " ¿Cuál es la diferencia entre estas palabras que seguimos escuchando estos días, como inteligencia artificial (o IA), Machine Learning y Deep Learning?”

Bueno, permítanme explicar lo que es diferente entre ellos.

En breve, **IA** trata de hacer las computadoras inteligentes para imitar las funciones cognitivas de los seres humanos. Así que, la Inteligencia Artificial es un campo general con un amplio alcance, incluyendo: Proceso de lenguaje, creatividad y resumen.

Machine Learning es la rama de la IA que cubre la parte estadística de la inteligencia artificial. Enseña a la computadora a resolver problemas al mirar cientos o miles de ejemplos, aprender de ellos, y luego usar esa experiencia para resolver el mismo problema en nuevas situaciones.

Y **Deep Learning** es un campo muy especial de Machine Learning donde las computadoras pueden aprender y tomar decisiones inteligentes por su cuenta. Deep learning involucra un nivel más profundo de automatización en comparación con la mayoría de los algoritmos de machine learning.

Python para Machine Learning

Python es un lenguaje de programación de propósito general popular y potente que surgió recientemente como el idioma preferido entre los científicos de datos.
Pueden escribir su algoritmo de machine learning usando Python, y funciona muy bien.

Sin embargo, hay muchos módulos y bibliotecas ya implementados en Python que pueden hacer su vida mucho más fácil.
Tratamos de introducir paquetes de Python en este curso y usarlos en los laboratorios para dar una mejor experiencia.

El primer paquete es Numpy, una biblioteca de matemática para trabajar con arreglos de n-dimensiones en Python. Esta te permite realizar cálculos de forma eficiente y eficaz.
Es mejor que Python regular debido a sus increíbles capacidades.
Por ejemplo, para trabajar con arreglos, diccionarios, funciones, tipos de datos, y trabajar con imágenes, necesitas conocer Numpy.

SciPy es una colección de algoritmos numéricos y herramientas de dominio especifico, incluyendo procesamiento de señal, optimización, estadísticas y mucho más.SciPy es una buena biblioteca para la computación científica y de alto rendimiento.

Matplotlib es un paquete muy popular que proporciona trazado 2D, así como trazado 3D.

El conocimiento básico acerca de estos 3 paquetes, que están construidos sobre Python, es un buen activo para los científicos de datos que quieran trabajar con problemas del mundo real.

La biblioteca Pandas es una biblioteca de Python de muy alto nivel que proporciona estructuras de datos de alto rendimiento, fáciles de utilizar . Tiene muchas funciones para la importación, manipulación y análisis de datos. En particular, ofrece estructuras de datos y operaciones para la manipulación de tablas numéricas y series de tiempo.

scikit-learn es una colección de algoritmos y herramientas para machine learning, que es nuestro objetivo aquí, y que aprenderás a utilizar en el presente curso.
Como vamos a utilizar scikit-learn en los laboratorios, permítanme explicar más acerca de este y mostrarte por qué es tan popular entre los científicos de datos.

SciKit-learn es una biblioteca gratuita de machine learning para el lenguaje de programación Python. Tiene la mayoría de los algoritmos de clasificación, regresión y agrupamiento, y está diseñada para trabajar con las bibliotecas numéricas y científicas de Python, NumPy y SciPy. Además, incluye una documentación muy buena.

Encima de eso, la implementación de modelos de machine learning con scikit-learn es realmente fácil, con unas pocas líneas de código Python. La mayoría de las tareas que se deben realizar en un pipeline de machine learning se implementan ya en scikit-learn, incluyendo, preprocesamiento de datos, selección de características, extracción de características, división de entrenamiento/prueba, definición de los algoritmos, modelos de ajuste, parámetros de ajuste, predicción, evaluación y exportación del modelo.
Déjenme mostrarles un ejemplo de cómo scikit-learn luce cuando se utiliza esta biblioteca.
No tienes que entender el código por ahora, pero sólo ve lo fácil que puedes construir un modelo con sólo unas pocas líneas de código. Básicamente, los algoritmos de machine learning se benefician de la estandarización del conjunto de datos.

Si hay algunos valores atípicos, o campos de escalas diferentes en su conjunto de datos, tiene que arreglarlos. El paquete de preprocesamiento de scikit-learn proporciona varias funciones comunes de utilidad y las clases de transformación para cambiar los vectores de características en bruto en una forma adecuada de vector para modelado.

Hay que dividir el conjunto de datos en conjuntos de entrenamiento y pruebas para entrenar su modelo, y luego probar la precisión del modelo por separado.

El Scikit-learn puede dividir arreglos o matrices en subconjuntos aleatorios de entrenamiento y pruebas para ti, en una línea de código. Entonces, puedes configurar tu algoritmo.
Por ejemplo, puede crear un clasificador utilizando un algoritmo de clasificación de vector de soporte.

```
from sklearn import preprocessing
X = preprocessing.StandardScaler().fit(X).transform(X)
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)
```

```
from sklearn import svm
clf = svm.SVC(gamma=0.001, C=100.)
```

```
clf.fit(X_train, y_train)
```

Llamamos a nuestra instancia de estimador 'clf', e inicializamos sus parámetros. Ahora, puedes entrenar tu modelo con el conjunto de entrenamiento.
Al pasar nuestro conjunto de entrenamiento al método 'fit', el modelo 'clf' aprende a clasificar casos desconocidos.

```
clf.predict(X_test)
```

A continuación, podemos utilizar nuestro conjunto de pruebas para ejecutar predicciones. Y, el resultado nos dice cuál es la clase de cada valor desconocido.

Además, puede utilizar distintas métricas para evaluar la precisión del modelo, por ejemplo, utilizando una matriz de confusión (confusion_matrix) para mostrar los resultados.

```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test, yhat, labels=[1,0]))
```

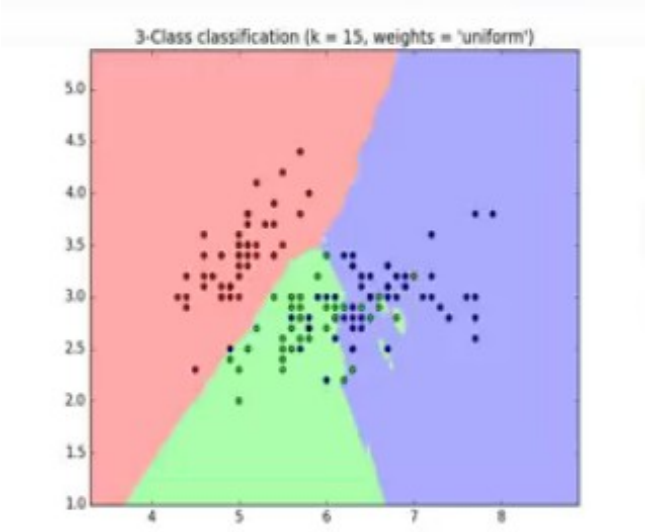
Y finalmente, guarda el modelo.

```
import pickle
s = pickle.dumps(clf)
```

Aprendizaje Supervisado y No Supervisado

Una manera fácil de empezar a entender el concepto de aprendizaje supervisado es observando directamente a las palabras que lo componen.

Supervisado significa observar y dirigir la ejecución de una tarea, proyecto o actividad. Obviamente, no vamos a estar supervisando a una persona...En vez de eso, estaremos supervisando un modelo de machine learning que podría ser capaz de producir regiones de clasificación como la que vemos aquí:



Entonces, ¿cómo supervisamos un modelo de machine learning?
Lo hacemos "educando" el modelo. Es decir, cargamos el modelo con conocimiento para que pueda predecir las instancias futuras.

" ¿Cómo se educa exactamente un modelo?"
Nosotros educamos el modelo entrenándolo con algunos datos de un conjunto de datos con etiquetas. Es importante tener en cuenta que los datos están etiquetados.
¿Y qué aspecto tiene un conjunto de datos etiquetado?
Bueno, puede verse como esto. Este ejemplo se toma del conjunto de datos de cáncer:

ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
1000025	5	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	benign
1015425	3	1	1	1	2	2	3	1	1	malignant
1016277	6	8	8	1	3	4	3	7	1	benign
1017023	4	1	1	3	2	1	3	1	1	benign
1017122	8	10	10	8	7	10		7	1	malignant
1018099	1	1	1	1	2	10	3	1	1	benign
1018561	2	1	2	H	2	1	3	1	1	benign
1033078	2	1	1	1	2	1	1	1	5	benign
1033078	4	2	1	1	2	1	2	1	1	benign

Como pueden ver, tenemos algunos datos históricos de los pacientes, y ya conocemos la clase de cada fila.

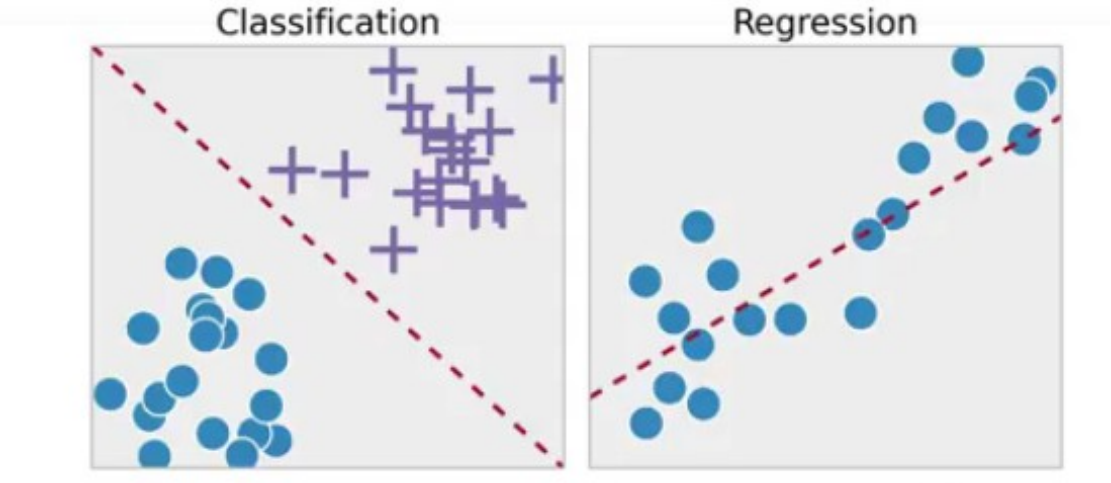
Comencemos por introducir algunos componentes de esta tabla. Los nombres aquí arriba, que se laman Espesor de grupo, Uniformidad del tamaño de la celda, Uniformidad de forma de célula, adhesión marginal, etc., se denominan atributos.

Las columnas se denominan Características, que incluyen los datos. Si grafica estos datos y mira un solo punto de datos en el gráfico, obtendrá todos estos atributos.
Esto haría una fila en este gráfico, a la que también se refiere como una observación.

Mirando directamente el valor de los datos, puede tener dos clases. La primera es numérica. Cuando se trata de machine learning, los datos utilizados con más frecuencia son numéricos.

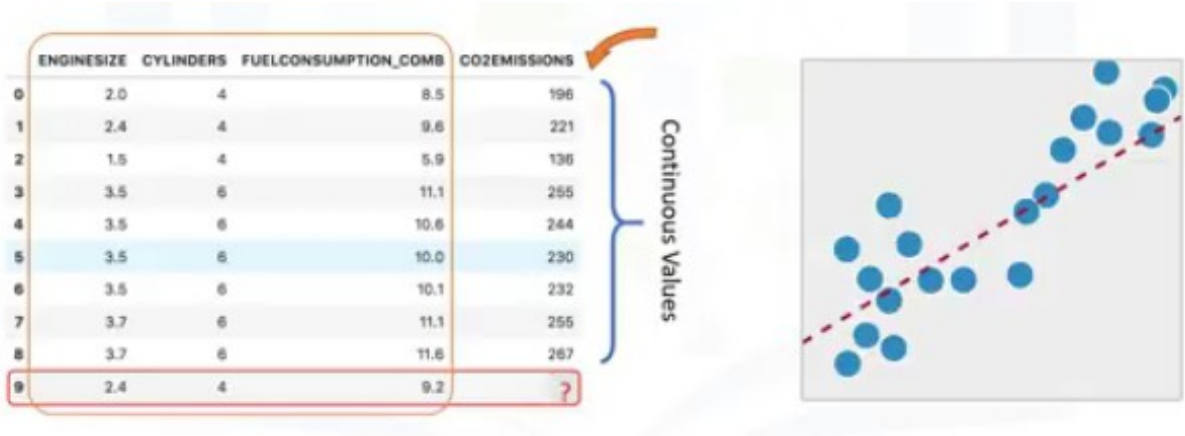
El segundo es categórico ... es decir, no es numérico, porque contiene caracteres en lugar de números. En este caso, es categórico porque este conjunto de datos se hace para Clasificación.
Existen dos tipos de técnicas de Aprendizaje Supervisado.

- Clasificación
- Regresión.



La **clasificación** es el proceso de predecir una categoría o etiqueta de clase discreta.
La **regresión** es el proceso de predicción de un valor continuo en contraposición a la predicción un valor categórico en Clasificación.

Observe este conjunto de datos.
Está relacionado con las emisiones de CO2 de diferentes coches.
Incluye el tamaño del motor, Cilindros, Consumo de combustible y emisión CO2 de varios modelos de automóviles.



Dado este conjunto de datos, puede utilizar regresión para predecir la emisión CO2 de un nuevo coche utilizando otros campos como, por ejemplo, el tamaño del motor o el número de cilindros.

El aprendizaje **no supervisado** es exactamente cómo suena. No supervisamos el modelo, dejamos que trabaje por su cuenta para descubrir información que puede que no ser visible para el ojo humano.

Esto significa que el algoritmo no supervisado entrena con el conjunto de datos, y extrae conclusiones sobre datos sin etiquetar.
En términos generales, el aprendizaje no supervisado tiene algoritmos más difíciles que el aprendizaje supervisado, ya que sabemos poco a nada sobre los datos o los resultados que se esperan.
Reducción de la dimensión, estimación de densidad, Análisis de cesta de mercado y Agrupación en clústeres son las técnicas de machine learning no supervisado usadas mas ampliamente.

La **reducción de dimensión** y/o la selección de características desempeñan un gran papel reduciendo las características haciendo que la clasificación sea más fácil.

El **análisis de la canasta de mercado** es una técnica de modelado basada en la teoría de que si se compra un cierto grupo de artículos, es más probable que compres otro grupo de artículos.

La **estimación de densidad** es un concepto muy simple que se utiliza principalmente para explorar los datosy encontrar alguna estructura interna.

El **agrupamiento** es considerado como una de las técnicas más populares en machine learning no supervisado utilizada para agrupar los puntos de datos u objetos similares de algún modo.
El análisis de clústeres tiene muchas aplicaciones en diferentes dominios, ya sea que un banco quiera segmentar a sus clientes en función de determinadas características, o de ayudar a un individuo a organizar y agrupar sus tipos de música favoritos!

En términos generales, sin embargo, la Agrupación se utiliza principalmente para: Descubrimiento de estructuras, Sintetización, y Detección de anomalías.

Por lo tanto, para recapitular

- La mayor diferencia entre el Aprendizaje supervisado y no supervisado es que el aprendizaje supervisado se ocupa de los datos etiquetados mientras que el aprendizaje no supervisado se ocupa de datos no etiquetados.
- En el aprendizaje supervisado, tenemos algoritmos de machine learning para Clasificación y Regresión.
- En el aprendizaje no supervisado, tenemos métodos como el agrupamiento.
- En comparación con el aprendizaje supervisado, el aprendizaje no supervisado tiene menos modelos y menos métodos de evaluación que se puedan utilizar para asegurarse de que el resultado del modelo es preciso.
- Como tal, el aprendizaje no supervisado crea un entorno menos controlable, ya que la máquina esta creando resultados para nosotros.

Módulo 2: Regresión

Objetivos del módulo

Aprender:

- Algoritmos de Regresión
- Evaluación de Modelo
- Evaluación de Modelo: Overfitting (sobreajuste) y Underfitting (subajuste)
- A Entender Diferentes Modelos de Evaluación
- Regresión Lineal Simple

Introducción al Modelo de Regresión

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

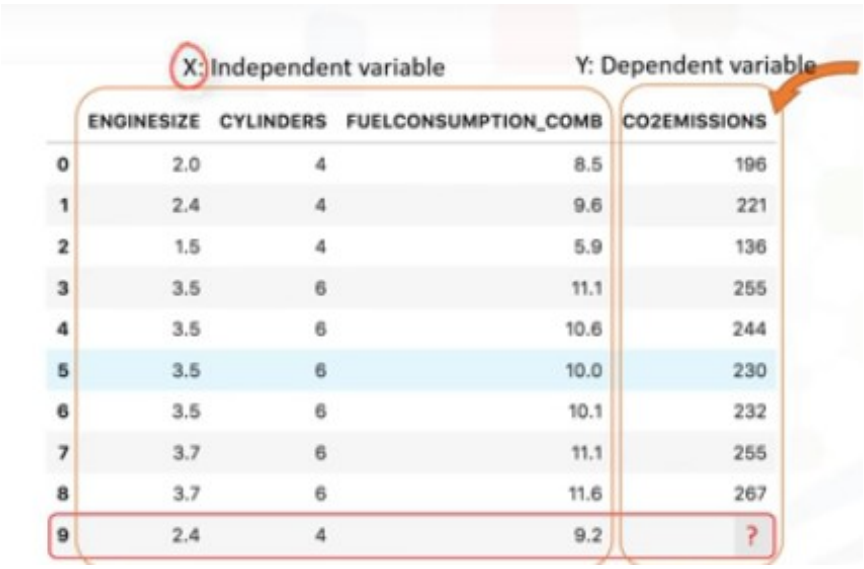
Mire este conjunto de datos. Estos estan relacionadas con las emisiones de CO2 de los diferentes autos. Incluye el tamaño del motor, el número de cilíndros, el consumo de combustible y las emisiones de CO2 de varios modelos automotrices.

La pregunta es, "Dado este conjunto de datos, podemos predecir la emisión CO2 de un coche utilizando otros campos como, por ejemplo, EngineSize o Cylinders?"

Supongamos que tenemos algunos datos históricos de diferentes autos, y suponemos que un auto, como en la fila 9, aún no se ha fabricado, pero estamos interesados en la estimación de su Aproximado para las emisiones de CO2, después de la producción. ¿Es posible?
Podemos utilizar métodos de regresión para predecir un valor continuo, como por ejemplo Emisión de CO2, algunas otras variables.
De hecho, la regresión es el proceso de predicción de un valor continuo.

En la regresión hay dos tipos de variables:

- una variable dependiente y
- una o más variables independientes



La variable **dependiente** se puede ver como el "estado", "objetivo" o "el objetivo final" que estudiamos y tratar de predecir, y las variables **independientes**, también conocidas como variables explicativas, pueden ser se ve como la "causa" de los "estados".

Las variables independientes se muestran convencionalmente por x; y la variable dependiente es denotada por y.

Un modelo de regresión se refiere a y, o a la variable dependiente, a una función de x, es decir, a las variables independientes.
El punto clave de la regresión es que nuestro valor dependiente debe ser continuo, y no puede ser un valor discreto.

Sin embargo, la variable independiente o las variables se pueden medir en categóricas o escala de medición continua.

Por lo tanto, lo que queremos hacer aquí es usar los datos históricos de algunos autos, usando uno o más de sus características, y a partir de esos datos, hacen un modelo.

Utilizamos la regresión para construir un modelo de regresión/estimación.
A continuación, el modelo se utiliza para predecir la emisión de CO2 esperada para un coche nuevo o desconocido.

Básicamente, hay 2 tipos de modelos de regresión:
- regresión simple
- regresión múltiple.

La **regresión simple** es cuando se utiliza una variable independiente para estimar una variable dependiente. Puede ser lineal en no lineales.
Por ejemplo, pronosticar la emisión de CO2 utilizando la variable EngineSize. La linealidad de la regresión se basa en la naturaleza de la relación entre la independencia y la dependencia variables.

Cuando hay más de una variable independiente presente, el proceso se llama **regresión múltiple, el cual puede ser lineal o no lineal**.

Por ejemplo, predicando emisiones de CO2 utilizando EngineSize y el número de Cylinders en cualquier auto dado.

De nuevo, en función de la relación entre las variables dependientes e independientes, puede ser regresión lineal o no lineal.
Examinemos algunas aplicaciones de muestra de regresión.

Esencialmente, utilizamos la regresión cuando queremos estimar un valor continuo.

Por ejemplo, una de las aplicaciones del análisis de regresión podría estar en el área de la previsión de ventas. Puede intentar predecir las ventas anuales totales de un vendedor de variables independientes como, por ejemplo, como la edad, la educación y los años de experiencia.

También se puede utilizar en el campo de la psicología, por ejemplo, para determinar la satisfacción individual basado en factores demográficos y psicológicos.

Podemos utilizar el análisis de regresión para predecir el precio de una casa en un área, basada en su tamaño, número de habitaciones, y así sucesivamente.

Incluso podemos usarlo para predecir el ingreso de empleo para las variables independientes, como las horas de trabajo, educación, ocupación, sexo, edad, años de experiencia, etc.

De hecho, se pueden encontrar muchos ejemplos de la utilidad del análisis de regresión en estos y muchos otros campos o dominios, como las finanzas, la salud, las ventas, y más.

Tenemos muchos algoritmos de regresión:

- Regresion ordinal
- Regresión de Poisson
- Regresión Fast forest quantile
- Lineal, Polinomial, Lasso, Stepwise y regresión Ridge
- Regresión linel Bayesiana
- Regresión por Redes Neuronales
- Decision Forest Regression
- Arboles de decisión Impulsados (*Boosted decision tree regresion*)
- KNN (K-nearest neighbours)

Cada uno de ellos tiene su propia importancia y una condición específica a la que su aplicación es las más adecuado.

Regresión Lineal Simple

Esta introducción de alto nivel les dará suficiente información de fondo sobre la regresión lineal para poder usarlo de forma efectiva en sus propios problemas.

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

Echemos un vistazo a este conjunto de datos. Está relacionado con la emisión CO2 de diferentes coches. Incluye el tamaño del motor, Cilindros, Consumo de combustible y las emisiones de CO2 para los distintos modelos de coches.

La pregunta es: Dados este conjunto de datos, podemos predecir la emisión de CO2 de un coche, utilizando otro campo como, por ejemplo, el tamaño del motor? ¡Sencillamente, sí! Podemos utilizar regresión lineal para predecir un valor continuo como, por ejemplo, Emisión de CO2, utilizando otras variables.

La regresión lineal es la aproximación de un modelo lineal que se utiliza para describir la relación entre dos o más variables.
En la regresión lineal simple, hay dos variables:

- una variable dependiente
- una variable independiente.

El punto clave en la regresión lineal es que **nuestro valor dependiente debe ser continuo** y no puede ser un valor discreto. Sin embargo, las variables independientes pueden ser medidas en una escala de medida categórica o continua.

Existen dos tipos de modelos de regresión lineal. Son: regresión simple y regresión múltiple.
La **regresión lineal simple** es cuando se utiliza una variable independiente para estimar una variable dependiente. Por ejemplo, predicando la emisión de CO2 utilizando la variable EngineSize.

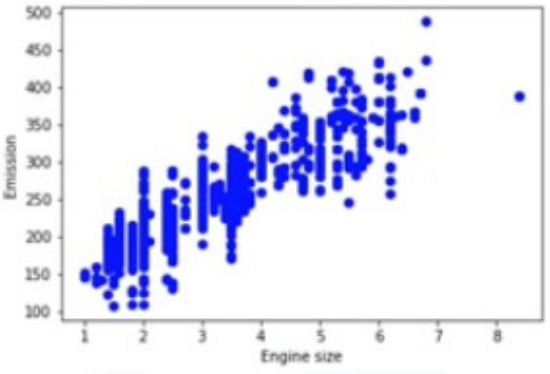
Cuando se utiliza más de una variable independiente ahora, el proceso se denomina **regresión lineal múltiple**.

Por ejemplo, prediciendo las emisiones de CO2 utilizando EngineSize y Cylinders de los coches.
Nuestro enfoque en este video es en una regresión lineal simple.

Ahora, veamos cómo funciona la regresión lineal. Bien, vamos a ver nuestro conjunto de datos de nuevo.

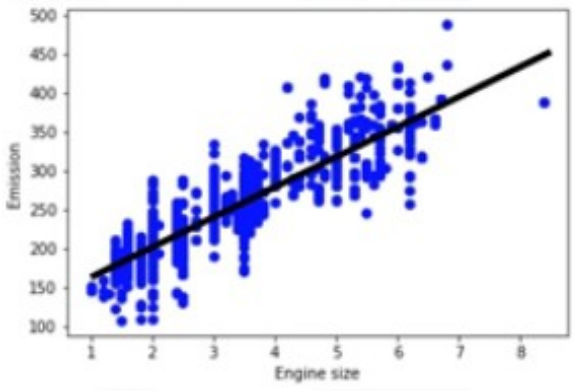
Para entender la regresión lineal, podemos trazar nuestras variables aquí. Mostramos el tamaño del motor como una variable independiente, y Emisión como el valor objetivo que nosotros queremos predecir. Un diagrama de dispersión muestra claramente la relación entre variables en las que los cambios de una variable "explican" o, posiblemente, "hacen que" cambie la otra variable.

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?



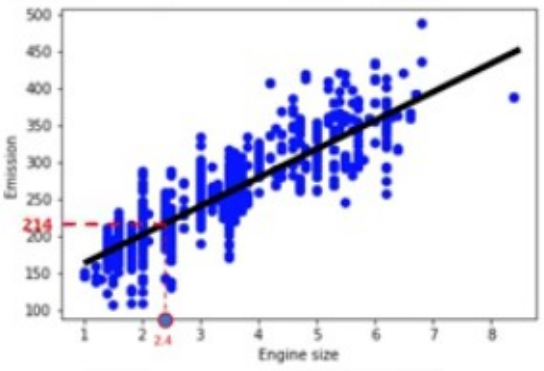
Además, indica que estas variables están relacionadas linealmente. Con la regresión lineal, puede ajustar una línea a través de los datos.
Por ejemplo, a medida que aumenta el EngineSize, también las emisiones. Con la regresión lineal, se puede modelar la relación de estas variables.

Un buen modelo se puede utilizar para predecir cuál es la emisión aproximada de cada coche.

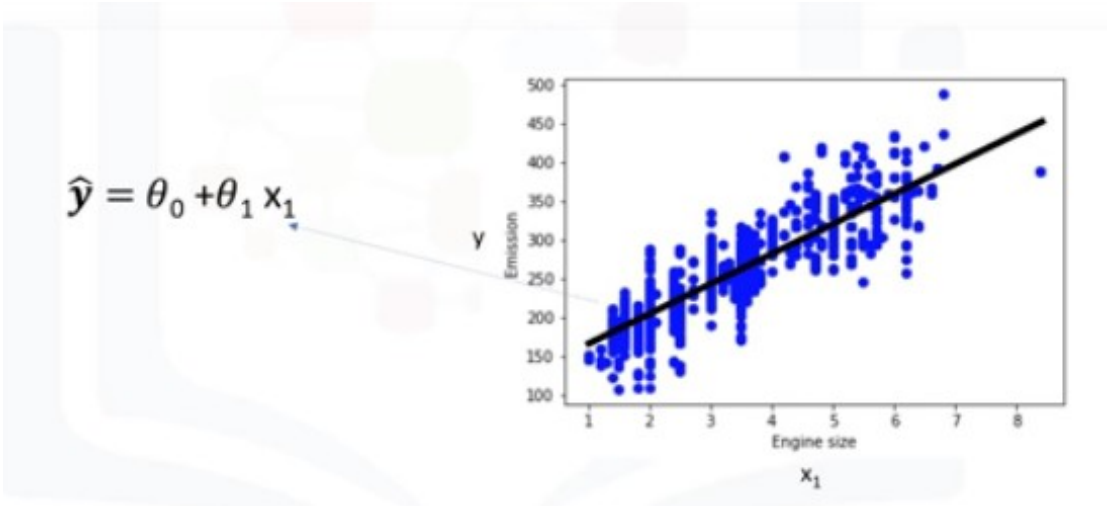


¿Cómo se utiliza esta línea para la predicción ahora?
Supongamos, por un momento, que la línea es un buen ajuste de los datos. Podemos usarlo para predecir la emisión de un coche desconocido. Por ejemplo, para un coche de muestra, con motor tamaño 2.4, se puede encontrar la emisión es 214.

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?



Ahora, vamos a hablar de lo que esta línea de ajuste es en realidad.
Vamos a predecir el valor objetivo, y. En nuestro caso, utilizando la variable independiente, "Tamaño de motor", representada por x1. La línea de ajuste se muestra tradicionalmente como un polinomio. En un problema de regresión simple (un solo x), la forma del modelo sería la $\theta 0 + \theta 1\ x1$.



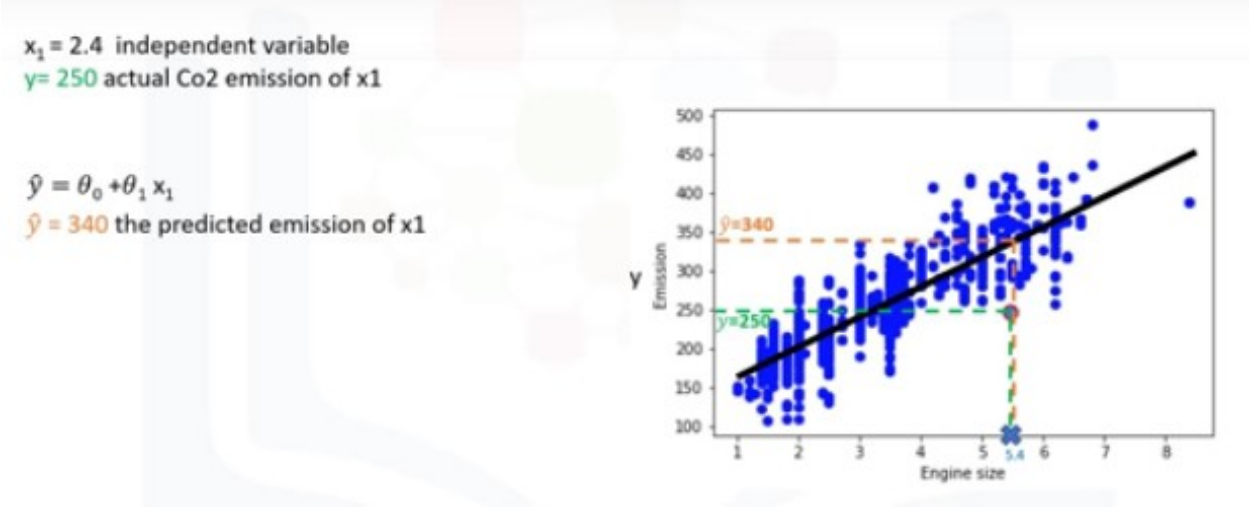
En esta ecuación, y es la variable dependiente o el valor pronosticado, x1 es la variable independiente; los parámetros 0 y 1 son los parámetros de la línea que debemos ajustar. θ1 es conocida como el "pendiente" o "gradiente" de la línea de ajuste y el valor θ0 se conoce como el "intercepto" θ0 y θ1 son los coeficientes de la ecuación lineal.

Se puede interpretar esta ecuación como si y fuera una función de x1, o una función que dependa de x1.

Ahora las preguntas son: "¿Cómo se dibujaria una línea a través de los puntos?" Y, "¿Cómo se determina cuál de las líneas "encaja mejor"?"
La regresión lineal estima los coeficientes de la línea. Esto significa que debemos calcular θ0 y θ1 para encontrar la mejor línea para "ajustar" los datos. Esta línea estimaría mejor la emisión de los data points desconocidos.

Vamos a ver cómo podemos encontrar esta línea, o para ser más precisos, cómo podemos ajustar los parametros para hacer que la línea sea el mejor ajuste para los datos.

Por un momento, supongamos que ya hemos encontrado la mejor línea de ajuste para nuestros datos. Ahora, vamos a pasar por todos los puntos y comprobar lo bien que se alinean con esta línea. El mejor ajuste, aquí, significa que si tenemos, por ejemplo, un coche con un tamaño de motor x1 = 5.4, y CO2 = 250, su CO2 debe aproximarse muy cerca del valor real, que es y=250, basado en datos históricos.
Pero, si usamos la línea de ajuste, o mejor dicho, usando nuestro polinomio con parámetros conocidos para predecir la emisión de CO2, devolverá y = 340.



Ahora, si comparamos el valor real de la emisión del coche con lo que predijimos utilizando nuestro modelo, descubrirá que tenemos un error de 90 unidades (error= y – ŷ = 250 - 340). Esto significa que nuestra línea de predicción no es precisa. Este error también se denomina **error residual**.
Por lo tanto, podemos decir que el error es la distancia desde el punto de datos hasta la línea de regresión ajustada.

La media de todos los errores residuales muestra lo mal que encaja la línea con todo el conjunto de datos.

Matemáticamente, puede ser demostrado por la ecuación, el error de cuadrado medio, mostrado como (MSE). Nuestro objetivo es encontrar una línea en la que se minimice la media de todos estos errores.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

En otras palabras, el error medio de la predicción utilizando la línea de ajuste debe minimizarse. Vamos a decirlo más técnicamente. El objetivo de la regresión lineal es minimizar la ecuación de MSE, y para minimizarla, deberíamos encontrar los mejores parámetros, para θ0 y θ1.

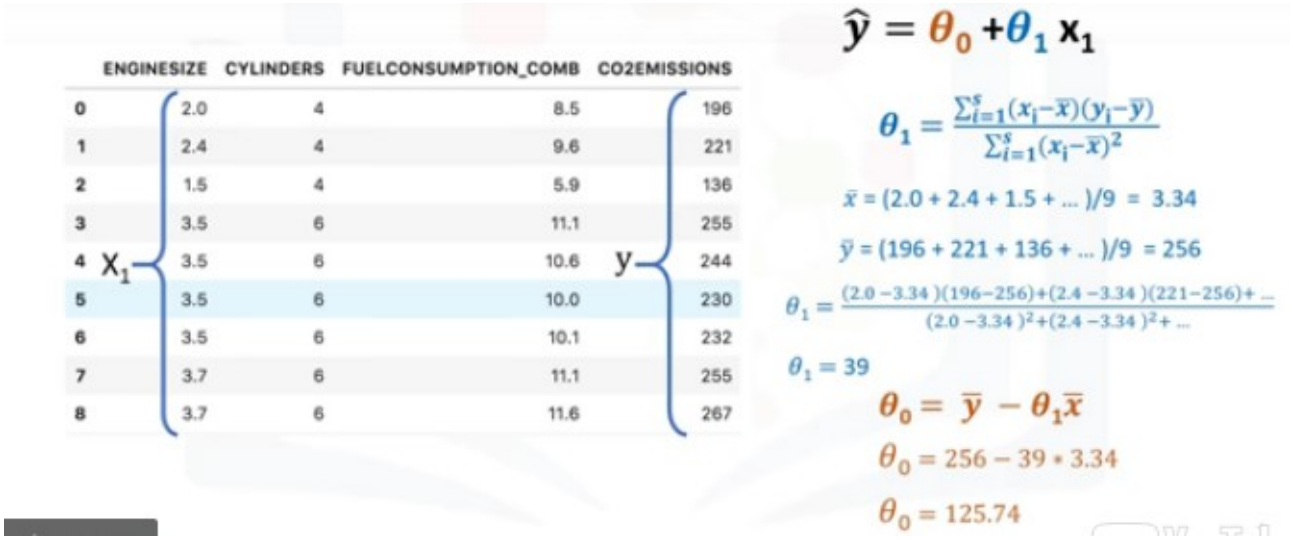
Ahora, la pregunta es, ¿cómo encontrar θ0 y θ1 de tal manera que se minimiza este error? ¿Cómo podemos encontrar una línea tan perfecta? O, dicho de otro modo, ¿cómo deberíamos encontrar los mejores parámetros para nuestra línea? ¿Deberíamos mover la línea de forma aleatoria y calcular el valor de MSE cada vez, y elegir el mínimo?

¡En realidad no! En realidad, tenemos dos opciones aquí:

1. Enfoque Matemático
2. Optimización

Vamos a ver cómo podemos utilizar fácilmente una fórmula matemática para θ0 y θ1.
Enfoque Matemático
Como se ha mencionado anteriormente, θ0 y θ1, en la regresión lineal simple, son los coeficientes de la línea de ajuste. Podemos usar una simple ecuación para estimar estos coeficientes. Esto es, dado que es un regresión lineal simple, con sólo 2 parámetros, y sabiendo que θ0 y θ1 son el intercepto y la pendiente de la línea, podemos estimarlos directamente a partir de nuestros datos.

Se requiere que calculemos la media de las columnas independientes y dependientes objetivo, del conjunto de datos.
Tenga en cuenta que todos los datos deben estar disponibles para atravesar y calcular los parámetros. Se puede mostrar que la intercepto y la pendiente se puede calcular utilizando estas ecuaciones.



Podemos empezar por estimar el valor de θ1.
Esta es la forma en la que se puede encontrar la pendiente de una línea basado en los datos. $\bar{x}$ es el valor promedio para el tamaño del motor en nuestro conjunto de datos.

Por favor, considere que tenemos 9 filas aquí, fila de 0 a 8. En primer lugar, calculamos el promedio de x1 y promedio de y. Luego lo agregamos en la ecuación de la pendiente, para encontrar θ1. Los xi y yi de la ecuación se refieren al hecho que tenemos que repetir estos cálculos a través de todos los valores de nuestro conjunto de datos y se refiere en el valor i'avo de x o y. Aplicando todos los valores, encontramos θ1 = 39; es nuestro segundo parámetro.

Se utiliza para calcular el primer parámetro, que es el intercepto de la línea. Ahora, podemos conectar θ1 a la ecuación de la línea para encontrar el valor de θ0.
Se calcula fácilmente el valor θ0 = 125.74. Así que, estos son los dos parámetros para la línea, donde θ0 es también llamado coeficiente de derivación y θ1 es el coeficiente para la columna de Emisión CO2.
Entonces: $\hat{y}$ = 125.74+ 39 x

Como nota aparte, realmente no necesitas recuerdar la fórmula para el cálculo de estos parámetros, como la mayoría de las librerias usadas para el aprendizaje automático en Python, R, y Scala pueden encontrar fácilmente estos parámetros para usted.
Pero siempre es bueno entender cómo funciona. Ahora, podemos escribir el polinomio de la línea. Así que sabemos cómo encontrar el mejor ajuste para nuestros datos, y su ecuación.

Ahora la pregunta es: "¿Cómo podemos usarlo para predecir la emisión de un nuevo coche basado en su tamaño del motor?"

Después de que encontramos los parámetros de la ecuación lineal, hacer predicciones es tan simple como solucionar la ecuación para un conjunto específico de entradas.

Imaginemos que estamos prediciendo CO2 Emission (y) de EngineSize (x) para el Automóvil en el registro número 9.
Representación de modelo de regresión lineal
para este problema sería: $y = \theta 0 + \theta 1\ x1$.
O si lo correlacionamos con nuestro conjunto de datos, sería $CO2Emission = \theta 0 + \theta 1$ EngineSize.
Como hemos visto, podemos encontrar $\theta 0$, $\theta 1$ usando las ecuaciones de las que acabamos de hablar. Una vez encontrados, podemos conectarlos a la ecuación del modelo lineal.

Por ejemplo, vamos a utilizar $\theta 0 = 125$ y $\theta 1 = 39$. Por lo tanto, podemos volver a escribir el modelo lineal como CO2Emission=125 + 39EngineSize.

Ahora, conectemos la novena fila de nuestro conjunto de datos y calcule la Emisión CO2 para un coche con un EngineSize de 2.4. Así que $CO2Emission = 125 + 39 \times 2.4$.

Por lo tanto, podemos predecir que la Emisión CO2 para este coche específico sería 218,6.

Vamos a hablar un poco sobre por qué la regresión lineal es tan útil. Sencillamente, es la regresión más básica que hay que utilizar y entender. De hecho, una razón por la que la regresión lineal es tan útil es que **es rápida**! También **no requiere ajuste de parámetros**. Por lo tanto, algo parecido a ajustar el parámetro K en los K-Vecinos mas cercanos o la tasa de aprendizaje en las Redes Neuronales no es algo sobre que preocuparse. La regresión lineal también **es fácil de entender y altamente interpretables**.

Regresión Lineal Multiple

La regresión lineal simple es cuando una variable independiente se utiliza para estimar una variable dependiente. Por ejemplo, predecir la emisión de CO2 utilizando la variable EngineSize.

En realidad, hay multiples variables que predicen la emisión de CO2.

Cuando hay múltiples variables independientes presentes, el proceso se llamado **"regresión lineal múltiple"**. Por ejemplo, predecir la emisión de CO2 utilizando EngineSize y el número de Cilindros en el motor del coche.

Lo bueno es que la regresión lineal múltiple es la extensión de el modelo de regresión lineal simple.

¿Qué tipo de preguntas podemos responder usándolo?
Básicamente, hay dos aplicaciones para la regresión lineal múltiple.

1. En primer lugar, puede ser utilizado cuando nos gustaría identificar la fuerza del efecto que as variables independientes tienen en una variable dependiente. Por ejemplo, el tiempo de revisión, la ansiedad de la prueba, la asistencia a clase y el género, ¿tiene algún efecto en el examen de rendimiento de los estudiantes?
2. En segundo lugar, se puede utilizar para predecir el impacto de cambios en las variables independientes sobre la dependiente. Es decir, para entender cómo cambia la variable dependiente cuando cambiamos las variables independientes. Por ejemplo, si estuviéramos revisando los datos de una persona, una regresión lineal múltiple puede indicarte cuánto aumenta (o disminuye) la presión arterial de esa persona por cada aumento (o disminución) unitario en el índice de masa corporal (IMC) de un paciente, manteniendo constante otros factores.

Como es el caso con regresión lineal simple, la regresión lineal múltiple es un método de predecir una variable continua. Utiliza múltiples variables, llamadas independientes variables, o predictores, que mejor predicen el valor de la variable objetivo, que es también llamado la variable dependiente.

En la regresión lineal múltiple, el valor objetivo, y, es una **combinación lineal de variables** independientes, x.
Por ejemplo, puede predecir la cantidad de CO2 que un coche puede emitir debido a variables independientes, como el tamaño del motor del coche, el número de cilíndros y el consumo de combustible.

La regresión lineal múltiple es muy útil porque puede examinar qué variables son predictores significativos de la variable de resultados. Además, puede averiguar cómo impacta cada característica la variable de resultados. Y de nuevo, como es el caso en la regresión lineal simple, si se consigue crear un modelo de regresión, se puede utilizar para predecir la cantidad de emisión de un caso desconocido.
Por lo general, el modelo tiene el formato: $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ y así en, hasta ... + $\theta_n x_n$.
Matemáticamente, podemos mostrarlo como una forma de vector también.

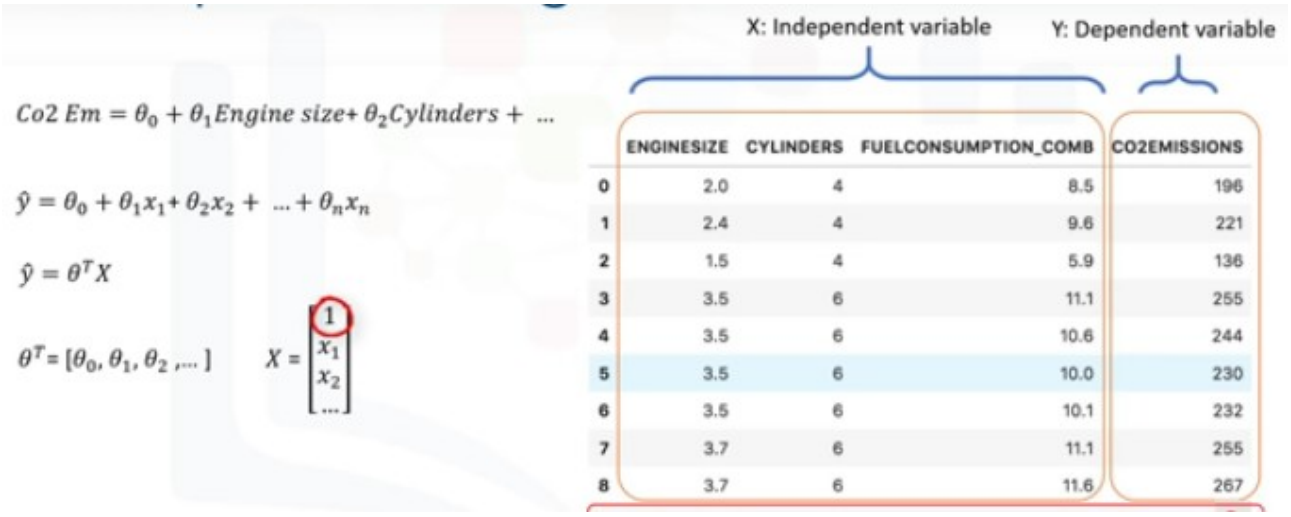
Esto significa que se puede mostrar como un producto de punto de 2 vectores: el vector de parámetros y el vector de conjunto de características.

Generalmente, podemos mostrar la ecuación de un espacio multidimensional como $\theta \wedge T x$, donde θ es un vector de n por un parámetros desconocidos en un espacio multidimensional, y x es el vector

de los conjuntos de características, como θ es un vector de coeficientes, y se supone que se multiplica por x. Convenientemente, se muestra como transponer θ.

θ es también llamado los parámetros, o, vector de peso de la ecuación de regresión ... ambos estos términos se pueden utilizar indistintamente.
Y x es el conjunto de características, que representa un coche. Por ejemplo, x1 para el tamaño del motor, o x2 para los cilindros, etc.

El primer elemento del conjunto de características se establecería en 1, porque convierte el θ_0 en el parámetro de intercepción o sesgo cuando el vector se multiplica por el vector de parámetro. Por favor, observe que θ ^ T x en un espacio dimensional, es la ecuación de una línea.



	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267

Es lo que usamos en regresión lineal simple. En dimensiones más altas, cuando tenemos más de una entrada (o x), la línea se denomina plano o hiper-plano. Y esto es lo que usamos para la regresión lineal múltiple. Por lo tanto, la idea es encontrar el mejor hiper-plano para nuestros datos.

Para este fin, y como es el caso en la regresión lineal, debemos estimar los valores para el vector θ que mejor predice el valor del campo objetivo en cada fila.

Para lograr este objetivo, tenemos que minimizar el error de la predicción.

Ahora, la pregunta es: "¿Cómo podemos encontrar los parámetros optimizados?"
Para encontrar los parámetros optimizados para nuestro modelo, primero debemos entender cuáles son los parámetros optimizados. A continuación, encontraremos una forma de optimizar los parámetros. En resumen, los parámetros optimizados son los que conducen a un modelo con los errores menos graves.

Supongamos, por un momento, que ya hemos encontrado el vector de parámetros de nuestro modelo. Significa que ya conocemos los valores del vector θ. Ahora, podemos utilizar el modelo, y el conjunto de características de la primera fila de nuestro dataset para predecir la emisión CO2 para el primer coche, correcto?

Si insertamos los valores del conjunto de características a la ecuación del modelo, encontraremos y. Digamos, por ejemplo, que devuelve 140 como valor predicho para esta fila específica. ¿Cuál es el valor real? y=196. ¿Que tan diferente es el valor pronosticado del valor real de 196? Bueno, podemos calcularlo simplemente, 196-140, que por supuesto = 56. Este es el error de nuestro modelo, sólo para una fila, o un coche, en nuestro caso.

Como es el caso en regresión lineal, podemos decir que el error aquí es la distancia desde el dato puntual hasta el modelo de regresión ajustada.

La media de todos los errores residuales muestra qué tan malo es el modelo representa el dataset. Se denomina mean squared error, o MSE.
Matemáticamente, la MSE puede ser mostrada por una ecuación.

$\hat{y} = \theta^T X$

$\hat{y}_i = 140$ the predicted emission of x_i

$y_i = 196$ actual value of x_i

$y_i - \hat{y}_i = 196 - 140 = 56$ residual error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267

Si bien esta no es la única forma de exponer el error de un modelo de regresión lineal múltiple, es una de las formas más populares de hacerlo.
El mejor modelo para nuestro dataset es el que tiene un error mínimo para todos los valores de predicción.
Por lo tanto, el objetivo de la regresión lineal múltiple es minimizar la ecuación de MSE.

Para minimizarlo, deberíamos encontrar los mejores parámetros θ, pero ¿cómo?
De acuerdo, " ¿Cómo se encuentra el parámetro o coeficientes para la regresión lineal múltiple?"

Hay muchas maneras de estimar el valor de estos coeficientes. Sin embargo, los métodos más comunes son los **mínimos cuadrados ordinarios** y el **enfoque de optimización**.

Minimos cuadrados ordinrios
Los mínimos cuadrados ordinarios tratan de estimar los valores de los coeficientes minimizando el "Mean Square Error".Este enfoque utiliza los datos como una matriz y utiliza operaciones de álgebra lineal para estimar los valores óptimos para la theta.

El problema con esta técnica es la complejidad del tiempo para calcular las operaciones matriciales, ya que puede tardar mucho tiempo en terminar. Cuando el número de filas del dataset es menos de 10.000 se puede pensar en esta técnica como una opción, sin embargo, para valores mayores, Deberías probar otros enfoques más rápidos.

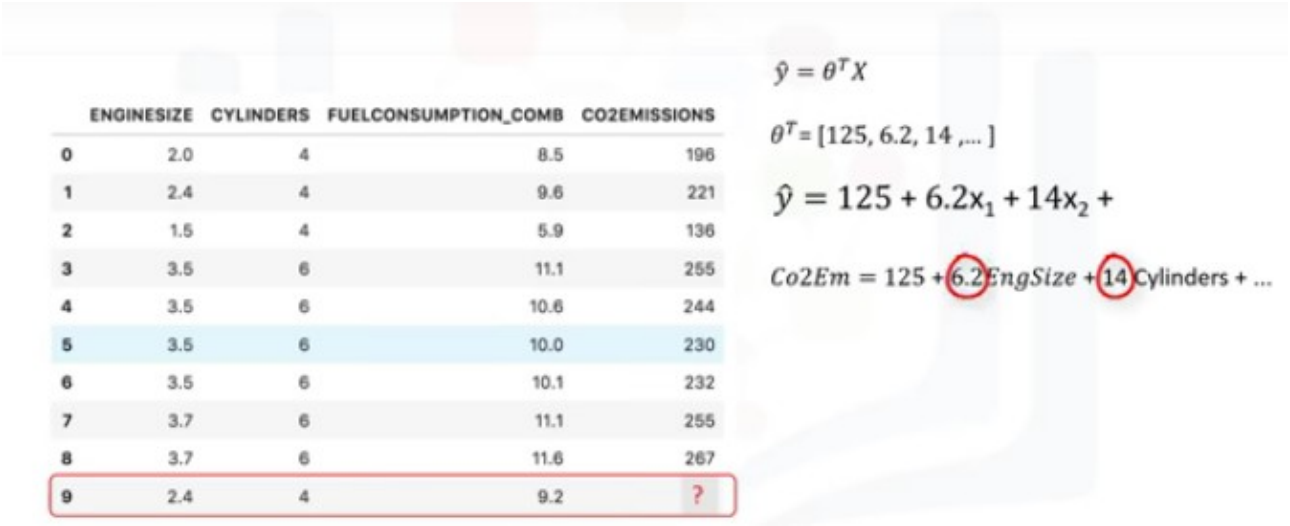
Algoritmos de optimización
La segunda opción es utilizar un algoritmo de optimización para buscar los mejores parámetros. Es decir, se puede utilizar un proceso de optimización de los valores de los coeficientes al minimizar de manera iterativa el error del modelo en sus datos de formación.

Por ejemplo, puede utilizar **Gradient Descent**, que inicia la optimización con valores aleatorios para cada coeficiente. A continuación, calcula los errores, e intenta minimizarlo cambiando sabiamente los coeficientes en múltiples iteraciones. Gradient descent es un enfoque adecuado si se tiene un dataset grande.

Por favor, entienda, sin embargo, que hay otros enfoques para estimar los parámetros de la regresión lineal múltiple que se puede explorar por su cuenta.

Fase de Predicción
Después de que encontramos los parámetros de la ecuación lineal, hacer predicciones es tan simple como resolver la ecuación para un conjunto específico de entradas. Imaginemos que estamos prediciendo emisiones de CO2 (o y) de otras variables para el automóvil
en el registro número 9.

Nuestra representación de modelo de regresión lineal para este problema sería: y = θ ^ T x. Una vez que encontremos los parámetros, podemos conectarlos en la ecuación del modelo lineal.
Por ejemplo, vamos a utilizar θ0 = 125, θ1 = 6.2, θ2 = 14, y entonces así. Si lo mapeamos con nuestro dataset, podemos reescribir el modelo lineal como "CO2 Emission=125 más 6.2 multiplicado por el tamaño del motor más 14 multiplicado por cilindro", y así.



	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

Como pueden ver, regresión lineal múltiple estimar la importancia relativa de los predictores. Por ejemplo, muestra que cilindro tiene mayor impacto en las cantidades de emisiones de CO2 en comparación con el tamaño del motor.

Ahora, conectemos la novena fila de nuestro dataset y calculemos la emisión de CO2 para un coche con el tamaño del motor de 2.4. Así que la Emisión de CO2 =125 + 6.2 × 2.4 + 14 × 4 ... y así. Podemos predecir la emisión de CO2 para este tipo de coche sería 214,1.

Concerns
Ahora permítanme abordar algunas preocupaciones que podría tener con respecto a la regresión lineal múltiple. Como ha visto, se puede utilizar múltiples variables independientes para predecir un valor objetivo en una regresión lineal múltiple.

A veces resulta un mejor modelo en comparación con el uso de una regresión lineal simple, que utiliza sólo una variable independiente para predecir la variable dependiente.

Ahora, la pregunta es: "¿Cuántas variables independientes se deberíamos utilizar para la predicción? ¿Deberíamos utilizar todos los campos de nuestro dataset? ¿Agregar variables independientes a un modelo regresión lineal múltiple siempre aumenta la precisión del módelo?

Básicamente, agregar demasiadas variables independientes sin ninguna justificación teórica puede resultar un **modelo de sobreajuste**. Un modelo de sobreajuste es un problema real porque es demasiado complicado para tu dataset y no es lo suficientemente general como para ser utilizado para la predicción.
Por lo tanto, se recomienda evitar el uso de muchas variables para la predicción.

La siguiente pregunta es: " ¿Deben las variables independientes ser continuas?" Básicamente, las variables independientes categóricas pueden incorporarse en un modelo de regresión al convirtiéndolas en variables numéricas. Por ejemplo, dada una variable binaria como tipo de coche, el código es "0" para "Manual" y 1 para los autos "automáticos".

Como último punto, recuerde que la "regresión lineal múltiple" es un tipo específico de regresión lineal. Por lo tanto, tiene que haber una relación lineal entre la variable dependiente y cada una de las variables independientes.
Hay una serie de formas de comprobar la relación lineal. Por ejemplo, puede utilizar diagramas de dispersión y, luego, comprobar visualmente la linealidad.

Si la relación mostrada en el diagrama de dispersión no es lineal, entonces, debe utilizar regresión no lineal.

Evaluación del Modelo en Modelos de Regresión
El objetivo de la regresión es crear un modelo para predecir con precisión un caso desconocido.
Con este fin, tenemos que realizar una evaluación de regresión después de crear el modelo.
Hay dos tipos de enfoques de evaluación que puedan ser utilizado para lograr este objetivo.

Estos enfoques son: el entrenamiento y la prueba en el mismo dataset, y la división de training/y data set.

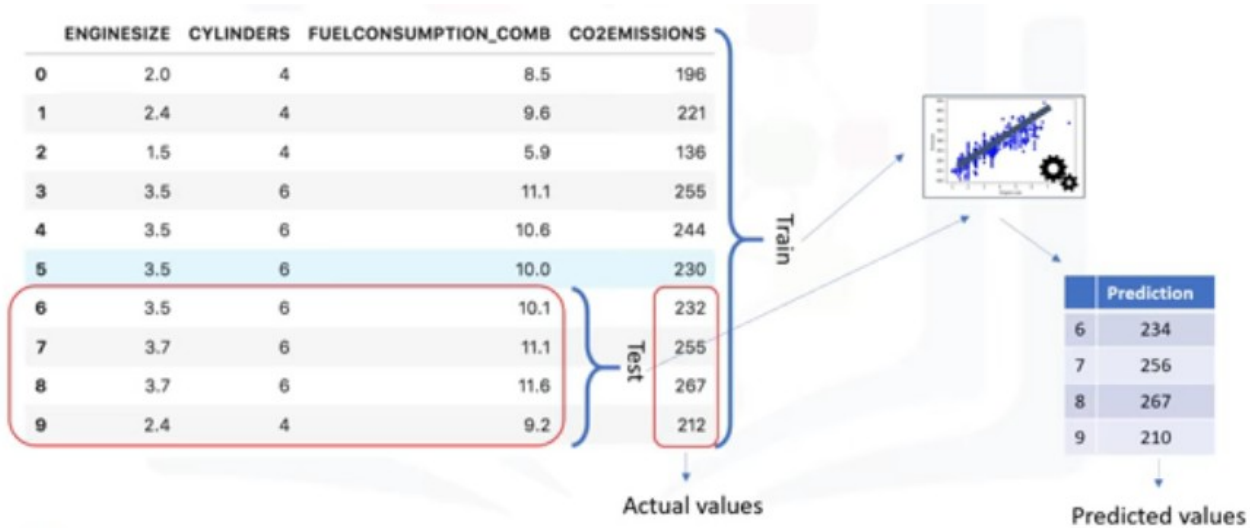
Entrenamiento y testeo en el mismo dataset
Al considerar los modelos de evaluación, es evidente que queremos elegir el que nos dará el resultados más precisos. Así que, la pregunta es, ¿cómo podemos calcular la precisión de nuestro modelo? En otras palabras, ¿cuánto podemos confiar en este modelo para la predicción de una muestra desconocida, utilizando un determinado dataset y habiendo construido un modelo como la regresión lineal.

Una de las soluciones consiste en seleccionar una parte de nuestro dataset para la realización de pruebas.
Por ejemplo, supongamos que tenemos 10 registros en nuestro dataset. Usamos todo el dataset para el entrenamiento, y construimos un modelo usando este conjunto de entrenamiento.

Ahora, seleccionamos una pequeña porción del dataset, tales como los números de fila 6 a 9, pero sin etiquetas. Este conjunto, se denomina un conjunto de pruebas, que tiene las etiquetas, pero las etiquetas no son utilizadas para la predicción, y se utiliza sólo como ground truth.

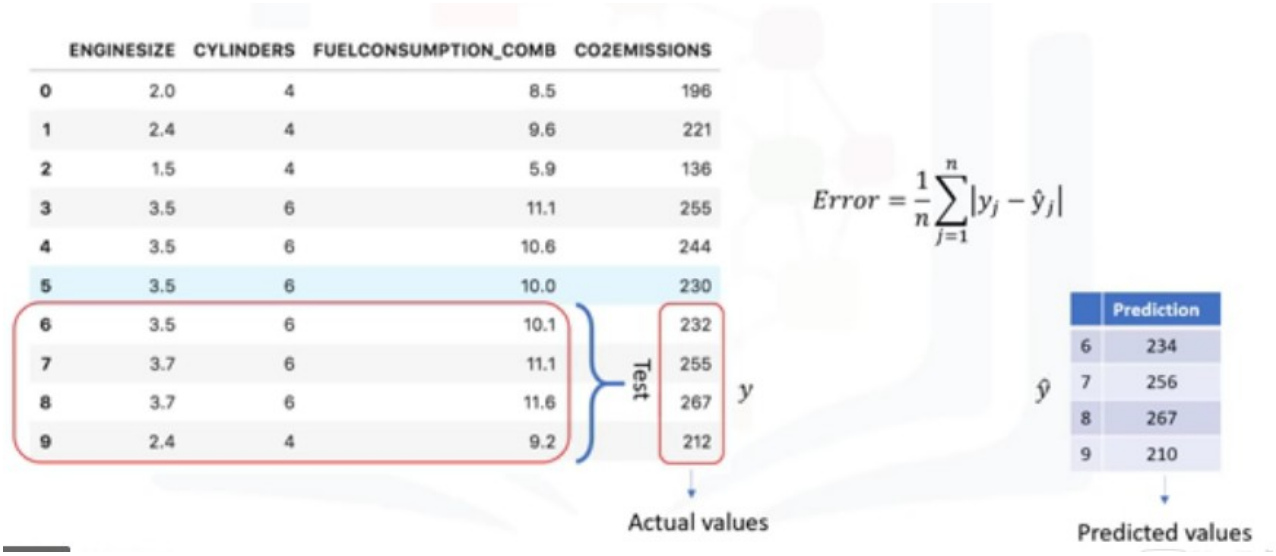
Las etiquetas se denominan "Valores reales" del conjunto de pruebas.
Ahora, pasamos el conjunto de características de la parte de prueba a nuestro modelo construido, y predicen el valores de destino.

Por último, comparamos los valores pronosticados por nuestro modelo con los valores reales de la prueba establecida. Esto indica que tan preciso es actualmente nuestro modelo.



Hay diferentes métricas para informar de la precisión del modelo, pero la mayoría de ellos trabajan generalmente, basado en la similitud de los valores pronosticados y reales.

Una de las métricas más simples para calcular la precisión de nuestro modelo de regresión es calcular el error del modelo como la diferencia promedio entre los valores predicho y los reales para todas las filas.



Podemos escribir este error como una ecuación. Entonces, el primer enfoque de evaluación del que acabamos de hablar es el más simple: entrenar y probar en el MISMO dataset.

Esencialmente, el nombre de este enfoque lo dice todo ... se entrena el modelo en todo el dataset, luego lo prueba utilizando una parte del mismo dataset.

En un sentido general, cuando se prueba con un dataset en el que se conoce el valor objetivo para cada dato puntual, es capaz de obtener un porcentaje de predicciones exactas para el modelo.

Este enfoque de evaluación probablemte tendría una alta "precisión de entrenamiento" y una baja "precisión fuera de la muestra", ya que el modelo conoce todos los data points de prueba de la entrenamiento.

¿Qué es la precisión de entrenamiento y la precisión de fuera de la muestra?
Hemos dicho que la entrenamiento y las pruebas en el mismo dataset produce una alta precisión de entrenamiento, pero ¿qué es exactamente la "precisión de entrenamiento"?

La **precisión de la entrenamiento** es el porcentaje de predicciones correctas que hace el modelo cuando se utiliza el dataset de prueba.
Sin embargo, una alta precisión en la entrenamiento no es necesariamente algo bueno. Por ejemplo, tener una alta precisión de entrenamiento puede dar como resultado un 'sobreajuste' de los datos. Esto significa que el modelo está demasiado formado en el dataset, que puede capturar el estruendo y producir un modelo no generalizado.

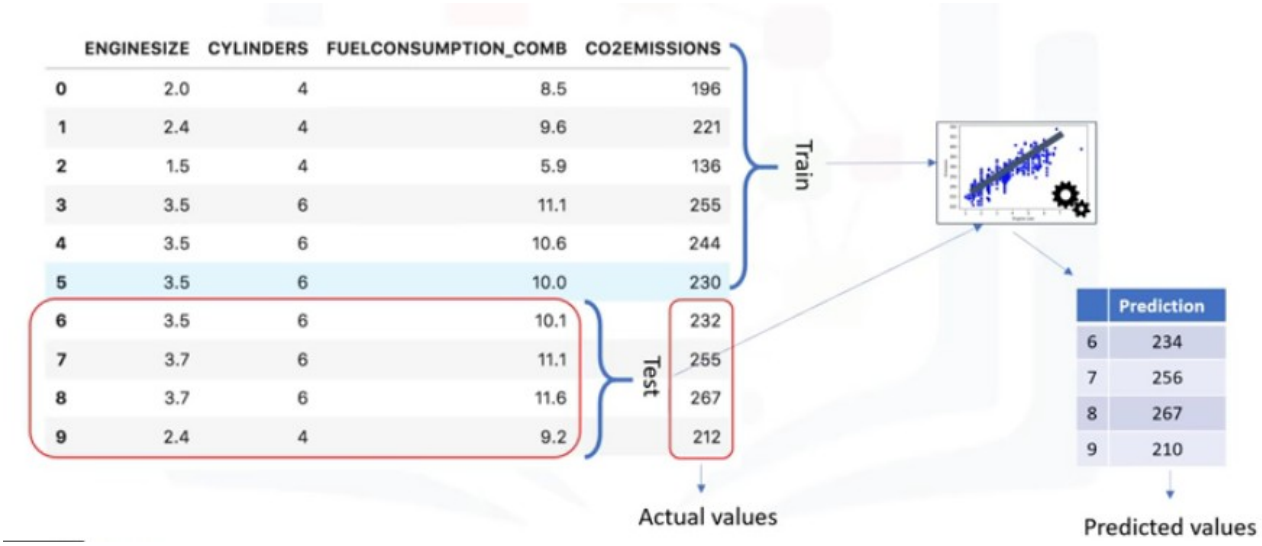
La **precisión fuera de la muestra** es el porcentaje de las predicciones correctas en las que el modelo realiza sobre datos en los que NO ha sido formado el modelo.

Hacer un "entrenamiento y prueba" en el mismo dataset probablemente tendrá una precisión baja fuera de la muestra debido a la probabilidad de estar en sobreajuste.
Es importante que nuestros modelos tengan una precisión alta, fuera de la muestra, porque el propósito de nuestro modelo es, por supuesto, hacer predicciones correctas sobre datos desconocidos.

Entonces, ¿cómo podemos mejorar la precisión fuera de la muestra?

Train/Test Split
En este enfoque, seleccionamos una parte de nuestro dataset para entrenamiento, por ejemplo, filas de 0 a 5. Y el resto se utiliza para probar, por ejemplo, las filas 6 a 9.

El modelo se basa en el conjunto de entrenamiento. Entonces, el conjunto de características de prueba se pasa al modelo para la predicción. Y finalmente, los valores pronosticados para el conjunto de pruebas se comparan con los valores reales del dataset.



Train/Test Split implica la división del dataset en conjuntos de entrenamiento y pruebas, respectivamente, que son mutuamente excluyentes, después de lo cual, se forman con el conjunto de entrenamiento y se prueba con el conjunto de pruebas.

Esto proporcionará una evaluación más precisa sobre la precisión de fuera de la muestra, ya que la prueba del dataset NO forma parte del dataset que se ha utilizado para formar los datos.
Es más realista para los problemas del mundo real.
Esto significa que conocemos los resultado de cada dato puntual en este dataset, haciéndolo genial para probar!
Y dado que estos datos no se han utilizado para formar el modelo, el modelo no tiene conocimiento de el resultado de estos data points. Así que, en esencia, es realmente una prueba fuera de la muestra.

Sin embargo, hay que asegurarse de armar el modelo con el conjunto de pruebas después, pues no es deseable perder datos potencialmente valiosos.

El problema con la train/test split es que es altamente dependiente de los dataset en los que los datos fueron formados y probados. La variación de esto hace que el train/test split tenga una mejor predicción fuera de la muestra que formando y probando en el mismo dataset, pero aún tiene algunos problemas debido a esta dependencia.

K-fold cross-validation

Otro modelo de evaluación, denominado "k-fold cross-validation", soluciona la mayoría de estos problemas.

¿Cómo se arregla una alta variación que resulta de una dependencia?

Se hace promedio.

El concepto básico de "k-fold cross-validation" es el siguiente:

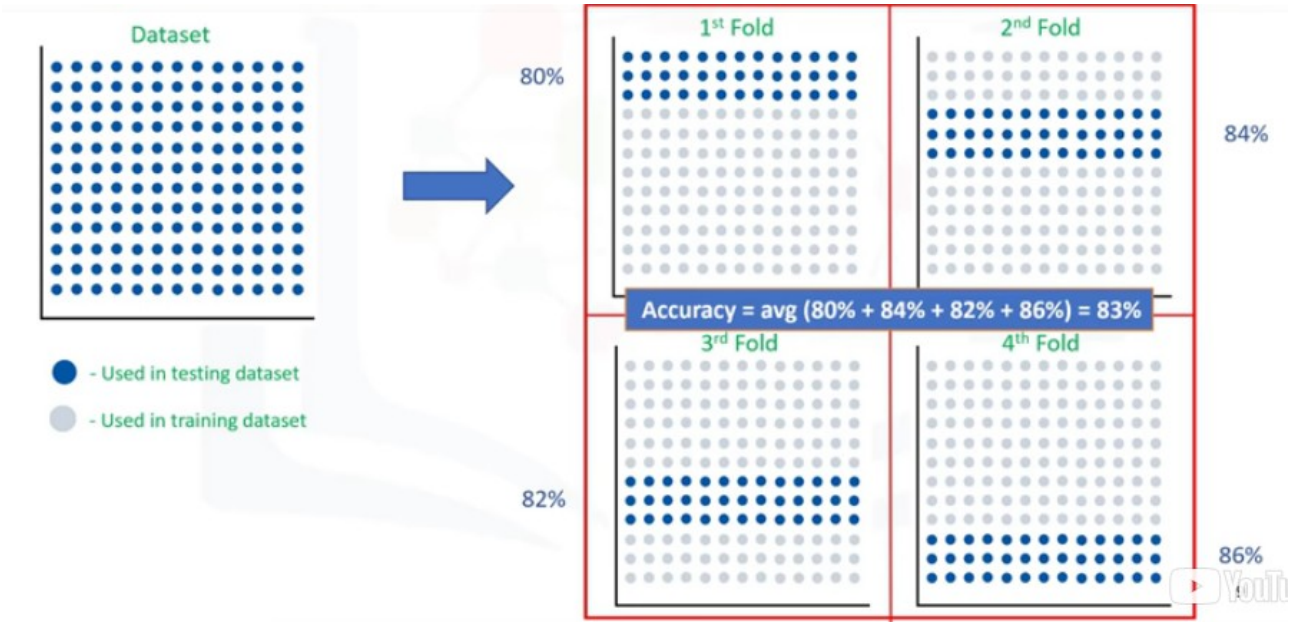
Todo el dataset es representado mediante los puntos de la imagen en la parte superior izquierda. Si tenemos k= 4 pliegues, entonces separamos este dataset tal como se muestra aquí.

En el primer pliegue, por ejemplo, usamos el primer 25 por ciento del dataset para las pruebas, y el resto para la entrenamiento. El modelo se crea utilizando el conjunto de entrenamiento, y se evalúa utilizando el conjunto de pruebas.

A continuación, en la siguiente ronda (o en el segundo pliegue), el segundo 25 por ciento del dataset se utiliza para las pruebas y el resto para la entrenamiento del modelo. Una vez más, se calcula la precisión del modelo y así se sigue por todos los pliegues.

Por último, se promedia el resultado de las cuatro evaluaciones.

Es decir, la precisión de cada pliegue es entonces promediada, teniendo en cuenta que cada pliegue es distinto, donde no se usa ningún dato de entrenamiento en un pliegue en otro.



K-fold cross-validation, en su forma más sencilla, realiza multiples train/test splits utilizando el mismo dataset en el que cada división es distinta. Entonces, el resultado es promediado para producir una precisión más consistente fuera de la muestra.

Métricas de Evaluación en Regresión

Las métricas de evaluación son usadas para explicar el rendimiento de un modelo.
Vamos a hablar más acerca de las métricas de evaluación de modelos que son usadas para la regresión.
Como se mencionó, básicamente, podemos comparar los valores actuales y predecir los valores a calcular la precisión de un modelo de regresión.

Las métricas de evaluación proporcionan un papel fundamental en el desarrollo de un modelo, ya que proporciona una percepción con respecto a las áreas que requieren mejoras.
Estaremos revisando una serie de modelos de evaluación métricas, incluyendo:

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)

Pero, antes de poder definir estos, tenemos que definir lo que realmente es un error.
En el contexto de la regresión, el **error del modelo** es la diferencia entre los datos puntuales y la línea de tendencia generada por el algoritmo.

Dado que hay multiples data points, un error puede ser determinado de multiples maneras.
Mean absolute error es la media del valor absoluto de los errores. Mean absolute error es la medida más fácil de entender, ya que es sólo el error promedio.

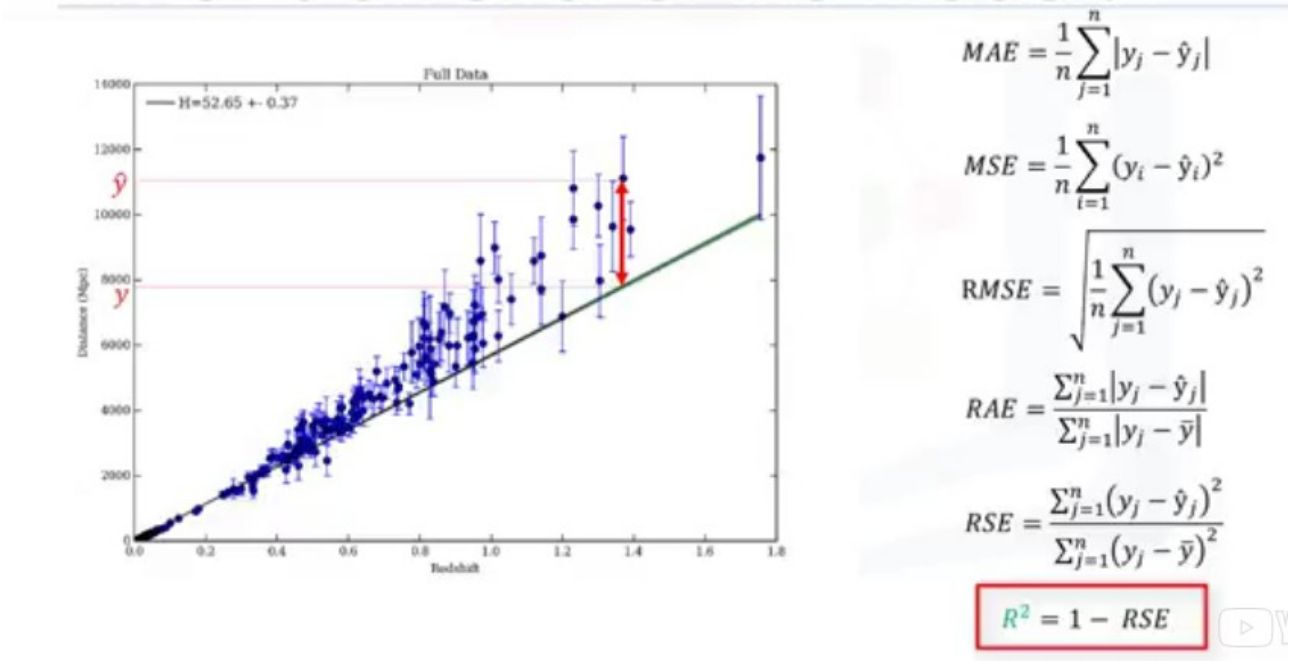
Media Squared Error (MSE) es la media del error al cuadrado. Es más popular que Mean absolute error porque el enfoque se orienta más hacia grandes errores. Esto se debe a que el término al

cuadrado aumenta exponencialmente los errores más grandes en comparación con los más pequeños.

Root Mean Squared Error (RMSE) es la raíz cuadrada de el error del cuadrado medio. Esta es una de las métricas más populares de las métricas de evaluación porque Root Mean Squared Error es interpretable en las mismas unidades como el vector de respuesta (o unidades "y") haciéndolo fácil de corelacionar la información.

Relative Absolute Error (RAE), también conocido como la suma residual de cuadrado, donde la barra de y es un valor medio de y, toma el error absoluto total y la normaliza dividiendo por el error absoluto total del predictor simple.

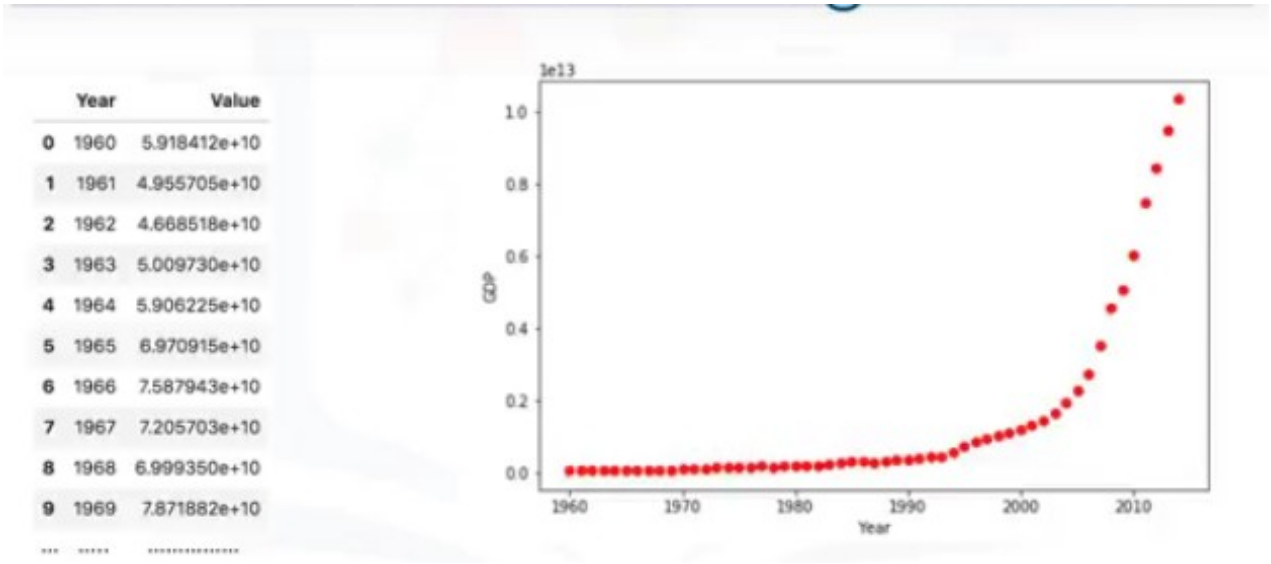
Relative Squared Error (RSE) es muy similar a "Relative absolute error", pero es ampliamente adoptado por la comunidad de data science, como es usado para calcular R-squared. R-squared no es un error, si no, es una métrica popular para la precisión de su modelo. Representa qué tan cerca los valores de los datos se encuentran en la línea de regresión ajustada. Cuanto más alto sea el R cuadrado, mejor encaja el modelo a tus datos.



Cada una de estas métricas se puede utilizar para cuantificar la predicción.
La elección de la métrica depende completamente del tipo de modelo, el tipo de datos y el dominio del conocimiento.

Regresión No-Lineal

Estos data points corresponden al producto bruto interno (PBI) de China de 1960 a 2014.
La primera columna, son los años, y la segunda, el ingreso interno bruto anual correspondiente de china en dólares de estadounidenses para ese año. Este es el aspecto de los data points.



	Year	Value
0	1960	5.918412e+10
1	1961	4.955705e+10
2	1962	4.668518e+10
3	1963	5.009730e+10
4	1964	5.906225e+10
5	1965	6.970915e+10
6	1966	7.587943e+10
7	1967	7.205703e+10
8	1968	6.999350e+10
9	1969	7.871882e+10
...	.....	.....

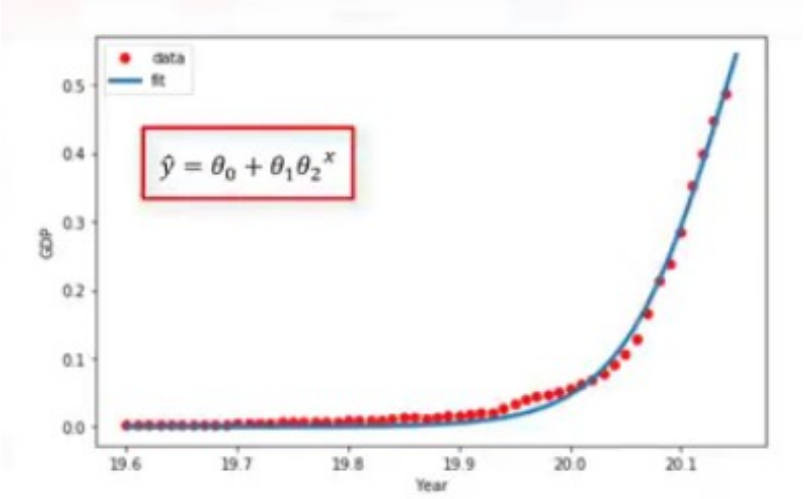
Ahora, tenemos un par de preguntas interesantes.
En primer lugar, " ¿Se puede predecir el PIB en función del tiempo?"
Y en segundo lugar, " ¿Podemos usar una regresión lineal simple para modelarlo?"

De hecho, si los datos muestran una tendencia curvada, entonces la regresión lineal no producirá un resultado muy preciso cuando se compara con una regresión no lineal, simplemente porque, como su nombre indica, la regresión lineal presume que los datos son lineales.

El diagrama de dispersión muestra que parece haber una fuerte relación entre el PIB y el tiempo, pero la relación no es lineal.
Como pueden ver, el crecimiento comienza lentamente, a partir de 2005 en adelante, el crecimiento es muy significativo.

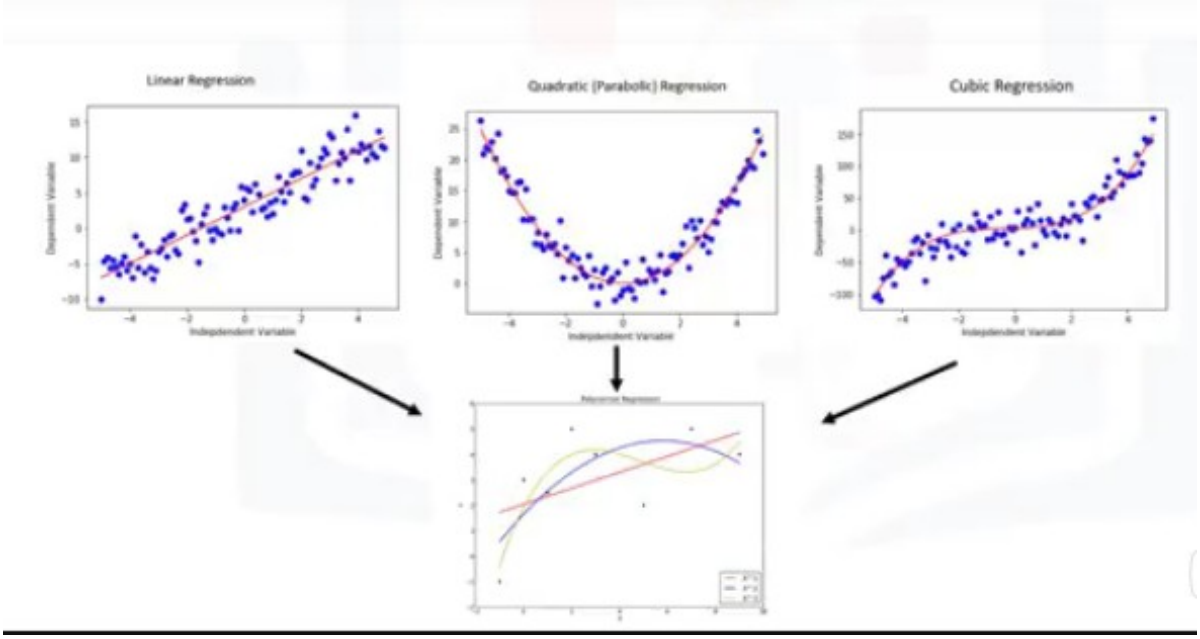
Y, por último, se desacelera ligeramente en los años 2010. Se parece a una función logística o exponencial. Por lo tanto, requiere un método especial de calculación para el procedimiento de regresión no lineal.

Por ejemplo, si asumimos que el modelo para estos data points son funciones exponenciales, como por ejemplo, y = θ_0 + θ_1 [θ_ 2] ^ x, nuestro trabajo consiste en estimar los parámetros del modelo, es decir, θ, y utiliza el modelo ajustada para predecir el PIB para casos desconocidos o futuros.



Regresiones polinómicas

Existen muchas regresiones diferentes que se pueden utilizar para adaptarse a cualquier aspecto del dataset. Aquí Puedes ver una línea de regresión cuadrática y cúbica, y puede continuar y continuar hasta el grados infinitos.



En esencia, podemos llamar a todos estos "regresión polinómica" , donde la relación entre la variable independiente x y la variable dependiente y se modelan como un polinomio grado n en x.

Con muchos tipos de regresión para elegir, hay una buena probabilidad de que uno se ajuste bien a su dataset. Recuerde, es importante elegir una regresión que se adapte mejor a los datos.

Entonces, ¿qué es la Regresión polinómica?
La regresión polinómica se ajusta a una línea curvada de sus datos.

Un ejemplo simple de polinomial, con el grado 3, se muestra como y = θ_0 + θ_1x + θ_2x ^ 2 + θ_3x ^ 3 o a la potencia de 3, donde θ son parámetros a estimar que hacen que el modelo se ajuste perfectamente a los datos subyacentes.

A pesar de que la relación entre x y y es no lineal aquí, y la regresión polinómica puede ajustarse a ellas, un modelo de regresión polinomial puede ser expresado como regresión lineal.

Dada la ecuación polinómica de tercer grado, mediante la definición de x1 = x y x2 = x ^ 2 ó x a la potencia de 2 y así, el modelo se convierte en una regresión lineal simple con nuevas variables, como y = θ0 + θ1 x1 + θ2 x2 + θ3 x3. Este modelo es lineal en los parámetros al ser estimado, ¿no?

Por lo tanto, esta regresión polinómica se considera un caso especial de regresión lineal múltiple tradicional.
Por lo tanto, puede utilizar el mismo mecanismo que la regresión lineal para resolver un problema de este tipo. Por lo tanto, los modelos de regresión polinomial pueden caber utilizando el modelo de mínimos cuadrados.

Los mínimos cuadrados es un método para estimar los parámetros desconocidos en una modelo de regresión lineal, minimizando la suma de los cuadrados de las diferencias entre la variable dependiente observada en el dataset determinado y las predicciones de la función lineal.

Entonces, ¿qué es la "regresión no lineal" exactamente?
En primer lugar, la regresión no lineal es un método para modelar una relación no lineal entre la variable dependiente y un conjunto de variables independientes.
En segundo lugar, para que un modelo sea considerado no lineal, y debe ser una función no lineal de los parámetros θ, no necesariamente las características x.

Cuando se trata de la ecuación no lineal, puede ser la forma de exponencial, logarítmico, y logística, o muchos otros tipos.

Como se puede ver, en todas estas ecuaciones, el cambio de y depende de los cambios en los parámetros θ, no necesariamente en x solamente. Es decir, en la regresión no lineal, un modelo no es lineal por parámetros.

En contraste con la regresión lineal, no podemos utilizar el método de "mínimos cuadrados" ordinarios para ajustar los datos en regresión no lineal, y en general, la estimación de los parámetros no es fácil.

" ¿Cómo puedo saber si un problema es lineal o no lineal de una manera fácil?"
Para responder a esta pregunta, tenemos que hacer dos cosas:
La primera es averiguar visualmente si la relación es lineal o no lineal.
Lo mejor es trazar gráficos bivariados de las variables de salida con cada variable de entrada.

Además, puedes calcular el coeficiente de correlación entre variables independientes y dependientes, y si para todas las variables es 0.7 o superior hay una tendencia lineal, y, por lo tanto, no es apropiado ajustar una regresión no lineal.

La segunda cosa que tenemos que hacer es usar regresión no lineal en vez de regresión lineal cuando no podemos modelar con precisión la relación con los parámetros lineales.

La segunda pregunta importante es: " ¿Cómo debo modelar mis datos, si se muestran no lineales en un diagrama de dispersión?"
Bueno, para hacer frente a esto, tienes que usar una regresión polinómica, usar un modelo de regresión no lineal, o "transformar" los datos.

Módulo 3: Clasificación

Objetivos del módulo

En ésta lección se va a ver:

- K-Vecinos más Próximos (KNN)
- Árboles de Decisión
- Máquinas de Soporte Vectorial (SVM)
- Regresión Logística

Introducción a la clasificasión

En Machine Learning, **la clasificación es un enfoque de aprendizaje supervisado**,que puede ser pensado como un medio de categorización o "clasificación" de algunos elementos desconocidos en un set discreto de "clases".

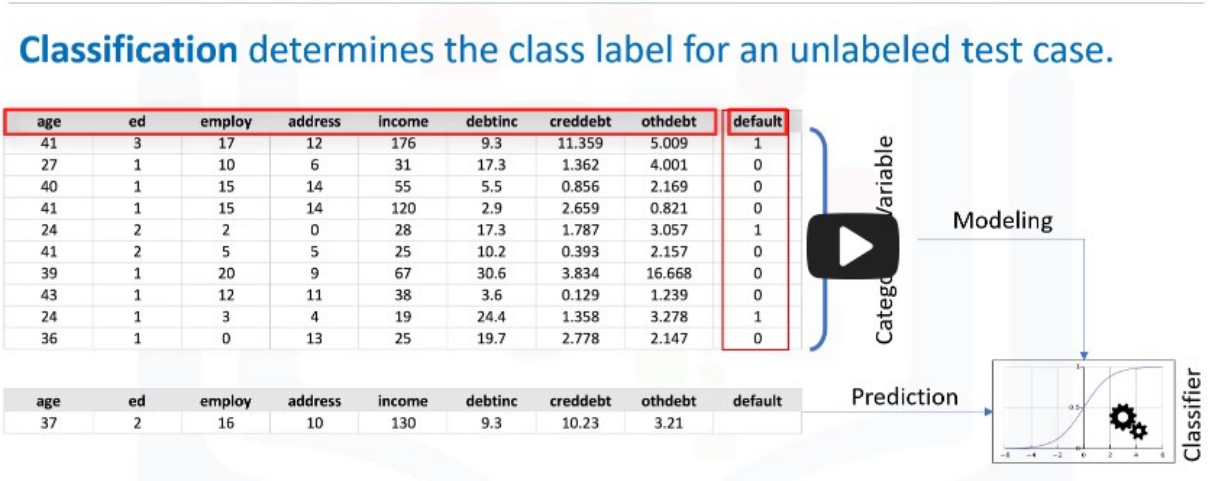
La clasificación intenta aprender la relación entre un set de variables feature o características de un objeto y una variable objetivo de interés.

El atributo objetivo en clasificación es una variable categórica con valores discretos.

¿cómo funcionan las clasificaciones y los clasificadores?
Dado un set de data points de entrenamiento,junto con las etiquetas objetivo,la clasificación determina la etiqueta de clase para un caso de prueba no etiquetado.

Clasificación binaria
Ejemplo:
Una buena muestra de clasificación es la predicción del incumplimiento de pago de un préstamo. Suponga que un banco está preocupado por el potencial de incumplimiento de pago de sus préstamos.
Si se puede usar datos previos de préstamos no pagados para predecir qué clientes podrían tener problemas pagando sus préstamos,estos clientes de "riesgo" pueden, o tener su aplicación de préstamo declinada, o se les puede ofrecer productos alternativos.

El objetivo de un predictor de incumplimiento de préstamo es utilizar los datos de incumplimiento de préstamo existentes,que es información sobre los clientes (como edad,ingresos,educación,etc.), para construir un clasificador, pasar a un nuevo cliente o un futuro potencal no pagador al modelo, y luego etiquetarlo (es decir,los datapoints) como "no pagador"; o "pagador",o por ejemplo,0 o 1.



Así es como un clasificador predice un caso de prueba sin etiquetas.

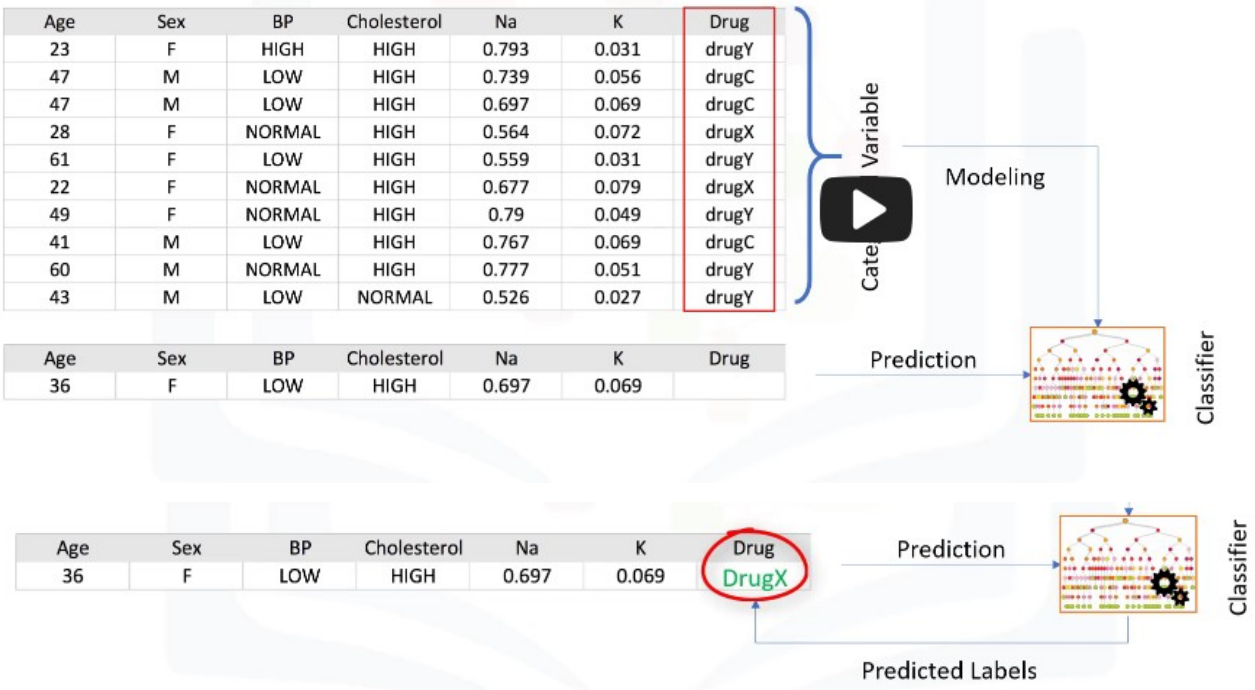
Tenga en cuenta que en este ejemplo específico era de **un clasificador binario con dos valores**.

Clasificación con multiclases
Ejemplo:
Imagine que ha recopilado datos acerca de un set de pacientes, los cuales sufrieron todos de la misma enfermedad.

Durante el curso de su tratamiento,cada paciente respondió a uno de tres medicamentos.

Puede utilizar este dataset etiquetado,con un algoritmo de clasificación,para construir un modelo de clasificación.

De esta manera puedes usarlo para averiguar qué medicamento podría ser apropiado para un futuro paciente con la misma enfermedad.
Como puede ver,se trata de una muestra de clasificación multi-clases.



Usos de la clasificación
La clasificación tiene también diferentes casos de uso empresarial, por ejemplo:
Para predecir la categoría a la que pertenece un cliente.
Para la detección del ratio de abandono,donde predecimos si un cliente cambia a otro proveedor o marca; o para predecir si un cliente responde o no a una campaña publicitaria en particular.
La clasificación de datos tiene varias aplicaciones en una gran variedad de industrias.

Esencialmente,muchos problemas pueden ser expresados como asociaciones entre el feature y la variable objetivo,especialmente cuando datos etiquetados están disponibles.

Esto proporciona una amplia gama de aplicabilidad para la clasificación.
Por ejemplo,la clasificación se puede utilizar para el filtrado de correos electrónico (de lo que va en la carpeta de archivos basura o no), reconocimiento de voz, reconocimiento de escritura a mano, identificación bio-métrica,clasificación de documentos,y mucho más.

Aquí tenemos los tipos de algoritmos de clasificación en machine learning.

- Árboles de decisión
- Naïve Bayes

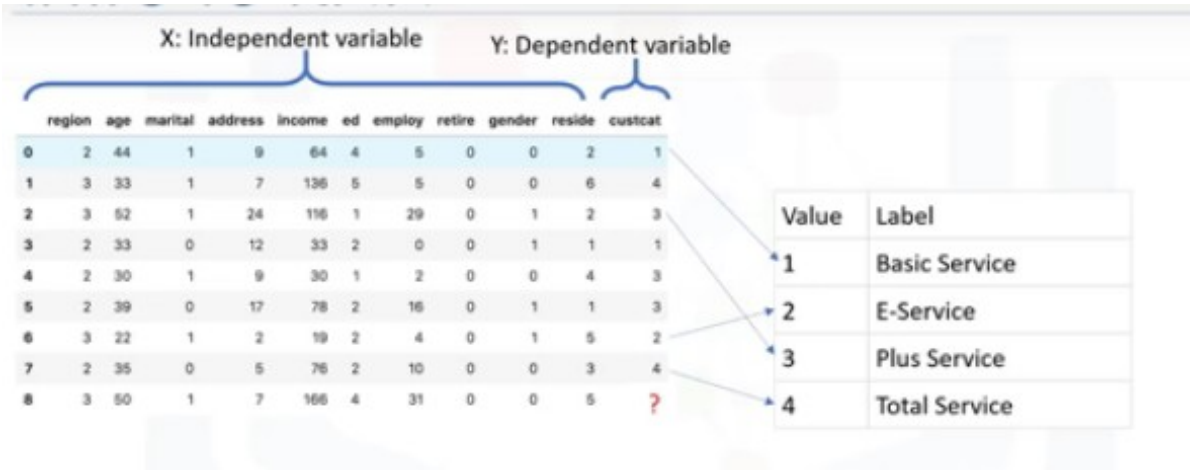
- Análisis Discriminante Linear
- KN-nearest neighbor
- Regresión logística
- Neural Networks
- Support Vector Machines.

Hay muchos tipos de algoritmos de clasificación.

KNN- Vecinos más cercanos:

Ejemplo:

Imagine que un proveedor de telecomunicaciones ha segmentado su base de clientes según el patrón de uso de servicios, categorizando a los clientes en cuatro grupos.



Si los datos demográficos se pueden utilizar para predecir la pertenencia a un grupo, la empresa puede personalizar ofertas para clientes potenciales. Se trata de un problema de clasificación.

Es decir, dado el conjunto de datos, con las etiquetas predefinidas, tenemos que construir un modelo que se utilizará para predecir la clase de un caso nuevo o desconocido.

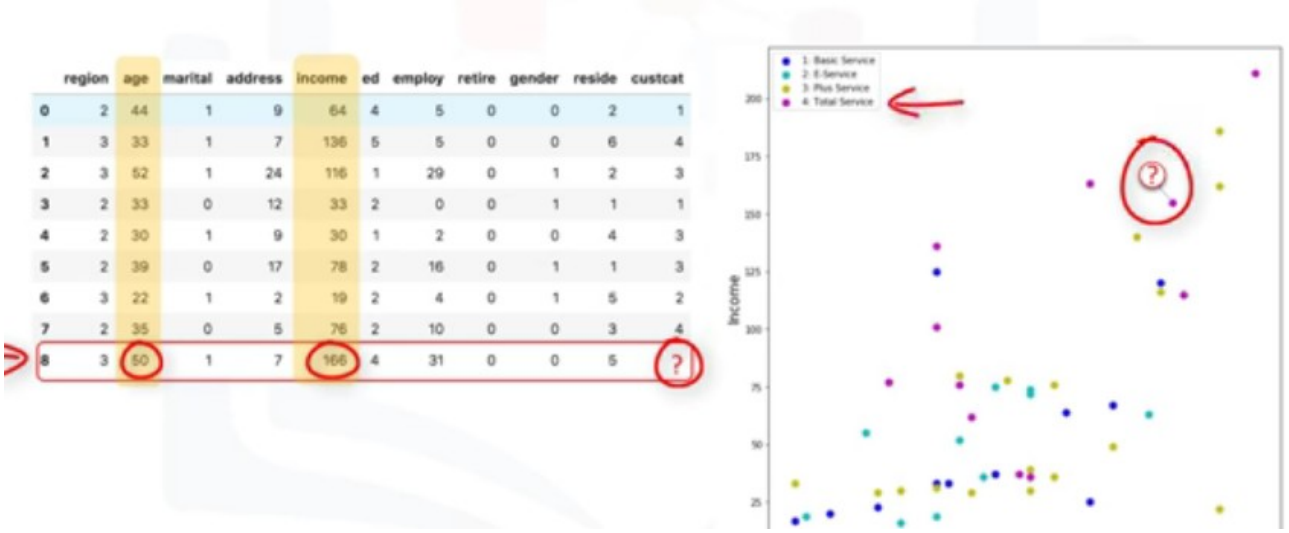
El ejemplo se centra en el uso de datos demográficos, como la región, la edad y el estado civil, para predecir patrones de uso.
El campo objetivo, llamado custcat, tiene cuatro valores posibles que corresponden a los cuatro grupos de clientes, como se indica a continuación: Servicio básico, servicio electrónico, servicio adicional y servicio total.

Nuestro objetivo es construir un clasificador, por ejemplo usando las filas 0 a 7, para predecir la clase de la fila 8.

Utilizaremos un tipo específico de clasificación llamada vecino más próximo.

Solo por el bien de la demostración, vamos a utilizar sólo dos campos como predictores, la Edad y el Salario, y luego trazar a los clientes en función de su membresia de grupo.

Ahora, digamos que tenemos un nuevo cliente, por ejemplo, el registro número 8 con una edad y salario conocido.



	region	age	marital	address	income	ed	employ	retire	gender	reside	custcat
0	2	44	1	9	64	4	5	0	0	2	1
1	3	33	1	7	136	5	5	0	0	6	4
2	3	52	1	24	116	1	29	0	1	2	3
3	2	33	0	12	33	2	0	0	1	1	1
4	2	30	1	9	30	1	2	0	0	4	3
5	2	39	0	17	78	2	16	0	1	1	3
6	3	22	1	2	19	2	4	0	1	5	2
7	2	35	0	5	76	2	10	0	0	3	4
8	3	50	1	7	166	4	31	0	0	5	?

¿Cómo podemos encontrar la clase de este cliente? ¿Podemos encontrar uno de los casos más próximos y asignar la misma etiqueta de clase a nuestro nuevo cliente? ¿Podemos también decir que la clase de nuestro nuevo cliente es probablemente el grupo 4 (es decir, total service) porque su vecino más próximo es también de la clase 4?

	region	age	marital	address	income	ed	employ	retire	gender	reside	custcat
0	2	44	1	9	64	4	5	0	0	2	1
1	3	33	1	7	136	5	5	0	0	6	4
2	3	52	1	24	116	1	29	0	1	2	3
3	2	33	0	12	33	2	0	0	1	1	1
4	2	30	1	9	30	1	2	0	0	4	3
5	2	39	0	17	78	2	16	0	1	1	3
6	3	22	1	2	19	2	4	0	1	5	2
7	2	35	0	5	76	2	10	0	0	3	4
8	3	50	1	7	166	4	31	0	0	5	?

Sí, podemos.
De hecho, es el primer vecino más próximo.
Ahora, la pregunta es: " ¿Hasta qué punto podemos confiar en nuestro juicio, que se basa en el el primer vecino más próximo?"

Podría ser un mal juicio, especialmente si el primer vecino más próximo es muy específico, o un valor atípico, ¿correcto?

Ahora, echemos un vistazo a nuestro diagrama de dispersión nuevamente.

En lugar de elegir al primer vecino más próximo, ¿qué tal si elegimos a los cinco vecinos más próximos, y realizó un voto mayoritario entre ellos para definir la clase de nuestro nuevo cliente?

En este caso, nos gustaría ver que tres de los cinco vecinos más próximos nos digan que vayamos a la clase 3, que es "Plus Service" ¿Esto no tiene más sentido? ¡ Sí, de hecho, sí!
En este caso, el valor de K en el algoritmo de vecinos k más próximo es 5.
Este ejemplo pone de relieve la intuición detrás del algoritmo del vecino más próximo.

El algoritmo en si
El algoritmo del vecino más próximo es un algoritmo de clasificación que toma un montón de puntos marcados y los utiliza para aprender a etiquetar otros puntos.
Este algoritmo clasifica los casos basados en su similitud con otros casos. En los vecinos más próximos, los puntos de datos que están cerca entre sí se dicen que son "vecinos".
El algoritmo se basa en este paradigma: " Casos similares con las mismas etiquetas de clase están cerca el uno al otro."Por lo tanto, la distancia entre dos casos es una medida de su disimilitud.

Existen diferentes maneras de calcular la similitud, o viceversa, la distancia o la la disimilitud de dos puntos de datos, por ejemplo, esto se puede realizar utilizando la distancia Euclidiana.
Ahora, vamos a ver cómo funciona realmente el algoritmo del vecino más próximo.

En un problema de clasificación, el algoritmo del vecino más próximo funciona de la siguiente forma:

1. Elija un valor para K.
2. Calcular la distancia desde el nuevo caso (exclusión de cada uno de los casos del conjunto de datos).
3. Buscar las "k" observaciones en los training data que son "más próximos" al punto de datos desconocido.

4. Predice la respuesta del punto de datos desconocido utilizando el valor de respuesta más popular de
5. los vecinos más próximos.

Hay dos partes en este algoritmo que pueden ser un poco confusas.
Problemas

1. ¿cómo seleccionar la K correcta?
2. ¿cómo calcular la similaridad entre los casos?

Cómo calcular la similaridad:
Empecemos primero con la segunda preocupación, es decir, ¿cómo podemos calcular la similaridad entre dos puntos de datos?

Suponemos que tenemos dos clientes, el cliente 1 y el cliente 2.
Y, por un momento, asuma que estos 2 clientes tienen sólo una característica, Edad.
Podemos utilizar fácilmente un tipo específico de distancia Minkowski para calcular la distancia de estos 2 clientes., la cual es, en efecto un caso de distancia euclidiana:

La distancia de x1 desde x2 es la raíz de 34 menos 30 a la potencia de 2, que es 4.

¿Qué pasa si tenemos más de una característica, por ejemplo Edad y Salario?
Si tenemos salario y edad para cada cliente, podemos seguir utilizando la misma fórmula, pero ahora lo usamos en un espacio bidimensional.

También podemos usar la misma matriz de distancia para los vectores multidimensionales.

Por supuesto, tenemos que normalizar nuestro conjunto de características para obtener la medida de disimilitud precisa.

Hay otras medidas de disimilitud que pueden ser utilizadas para este propósito, pero, tal como se menciona, es altamente dependiente del tipo de datos y también el dominio que la clasificación está hecho por él.
Como se ha mencionado, la K en el vecino más próximo, es el número de vecinos más próximos a examinar.
Se supone que debe ser especificado por el usuario.

Elección del parámetro k
Suponemos que queremos encontrar la clase del cliente que se indica como signo de interrogación en el gráfico.
Un valor bajo de K causa también un modelo muy complejo, lo que podría resultar en un exceso de ajuste del modelo.
Esto significa que el proceso de predicción no se generaliza lo suficiente como para ser utilizado para casos fuera de la muestra.

Los datos de fuera de la muestra son datos que están fuera del conjunto de datos utilizado para entrenar el modelo.
En otras palabras, no se puede confiar en que sea utilizado para la predicción de muestras desconocidas.

Es importante recordar que el exceso de ajuste es malo, ya que queremos un modelo general que funcione para cualquier dato, no sólo los datos utilizados para el entrenamiento

Ahora, en el lado opuesto del espectro, si elegimos un valor muy alto de K, tal como K=20, entonces el modelo se vuelve demasiado generalizado.
Así que, ¿cómo podemos encontrar el mejor valor para K?

La solución general es:

1. reservar una parte de sus datos para probar la precisión del modelo.
2. elige k =1, y luego usa la parte de entrenamiento para modelaje, y calcula la precisión de la predicción utilizando todos los ejemplos en el conjunto de pruebas.
3. Repita este proceso, incrementando el k, y vea cuál de los k es el mejor para su modelo.

El análisis del vecinos más próximo también se puede utilizar para calcular valores para un objetivo continuo.
En esta situación, se utiliza el valor de objetivo promedio o promedio de los vecinos más próximos para obtener el valor predicho para el nuevo caso.
Por ejemplo, suponga que está prediciendo el precio de un inicio basado en su conjunto de características, tales como el número de habitaciones, las imágenes cuadradas, el año en que fue construido, etc.
Usted puede encontrar fácilmente las tres casas vecinas más cercanas, por supuesto, no sólo en base a la distancia, pero también basado en todos los atributos, y luego predice el precio de la casa, como el promedio de los vecinos.