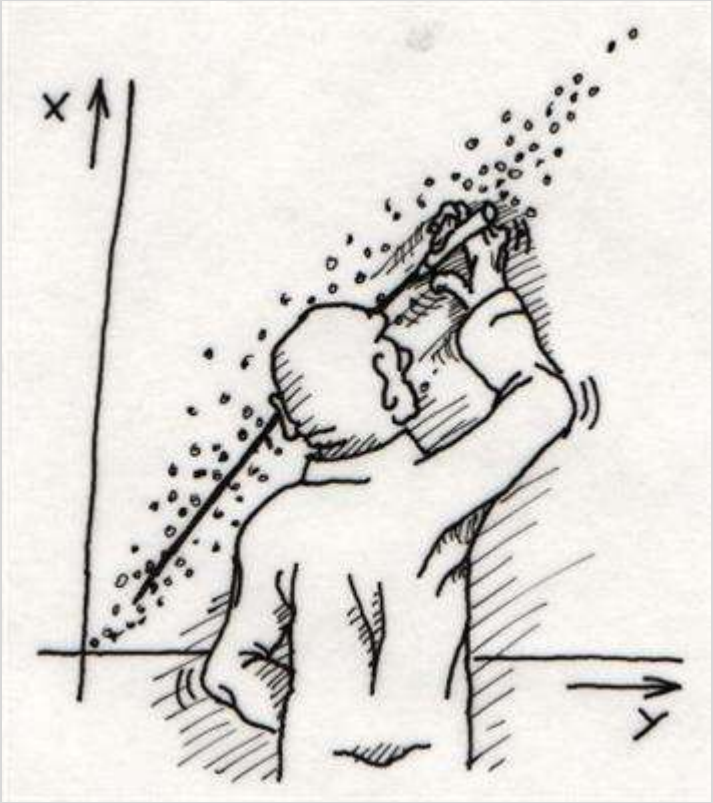


DigitalHouse > Coding School

DATA SCIENCE

MÓDULO 3

Intro a Machine Learning

Agosto de 2017

Machine Learning (ML) es la disciplina donde las **habilidades computacionales y algorítmicas** de la ciencia de datos se encuentran con el **pensamiento estadístico**, y el resultado es una colección de aproximaciones a la inferencia y exploración de datos que no son tanto una propuesta teórica sino más bien **una forma efectiva de computación**.

Mientras estos métodos pueden ser increíblemente poderosos, para ser efectivos deben utilizarse con un **sólido conocimiento de las fortalezas y debilidades de cada método**, así como un entendimiento general de conceptos, entre otros:

- sesgo
- varianza
- sobre-ajuste
- sub-ajuste

OBJETIVOS DE ESTA CLASE

Este material no intenta ser una introducción exhaustiva al campo de machine learning, ni intenta ser un manual completo para el uso del paquete Scikit-Learn. Los objetivos de las próximas clases son:

1. **Introducir el vocabulario y conceptos fundamentales de machine learning.**
2. **Introducir la API de Scikit-Learn y mostrar algunos ejemplos de su uso.**
3. **Estudiar más en profundidad algunos de los algoritmos más importantes de machine learning, y desarrollar una intuición sobre cómo trabajan y cuándo y dónde son aplicables**

Machine learning se suele categorizar como un **sub-campo de la inteligencia artificial**, pero en el contexto de su aplicación en data science es más útil pensar en machine learning como **un medio para construir modelos a partir de los datos**.

Fundamentalmente, machine learning involucra la **construcción de modelos matemáticos para ayudar a entender los datos.**

El aprendizaje (el learning de machine learning) entra en juego cuando le damos a estos modelos parámetros que pueden modificarse y que se adaptan a los datos observados de manera automática; de esta forma, puede considerarse que el algoritmo **aprende de los datos.**

Una vez que estos modelos han sido ajustados a datos observados previamente, pueden ser usados para **predecir y entender aspectos de datos nuevos**.

Comprender **la formulación del problema** en machine learning es esencial para usar estas herramientas efectivamente, por lo que comenzaremos con algunas **categorizaciones generales de los tipos de algoritmos** que estudiaremos en el curso.

Categorías de aproximaciones en Machine Learning

En su nivel más fundamental, machine learning puede ser categorizado en **dos tipos principales:**

El **aprendizaje supervisado** involucra modelar de alguna forma la relación entre características (features) medidas de los datos y alguna etiqueta (label) asociado con los datos; una vez que se determina el modelo, puede ser usado para aplicar nuevas labels a los nuevos datos desconocidos previamente.

Esta categoría se subdivide a su vez en tareas de **clasificación** y tareas de **regresión.** En clasificación, las labels son categorías discretas mientras que en regresión, las labels son cantidades continuas.

El **aprendizaje no supervisado** involucra el modelado de features de un dataset sin referencia a ninguna etiqueta y frecuentemente se describe como “dejar que el dataset hable por sí mismo”.

Estos modelos incluyen tareas tales como **clustering** y **reducción de la dimensionalidad**. Los algoritmos de clustering identifican distintos grupos de datos, mientras que los algoritmos de reducción dimensionalidad buscan la representación más sucinta de los datos.

Ejemplos cualitativos de aplicaciones de Machine Learning

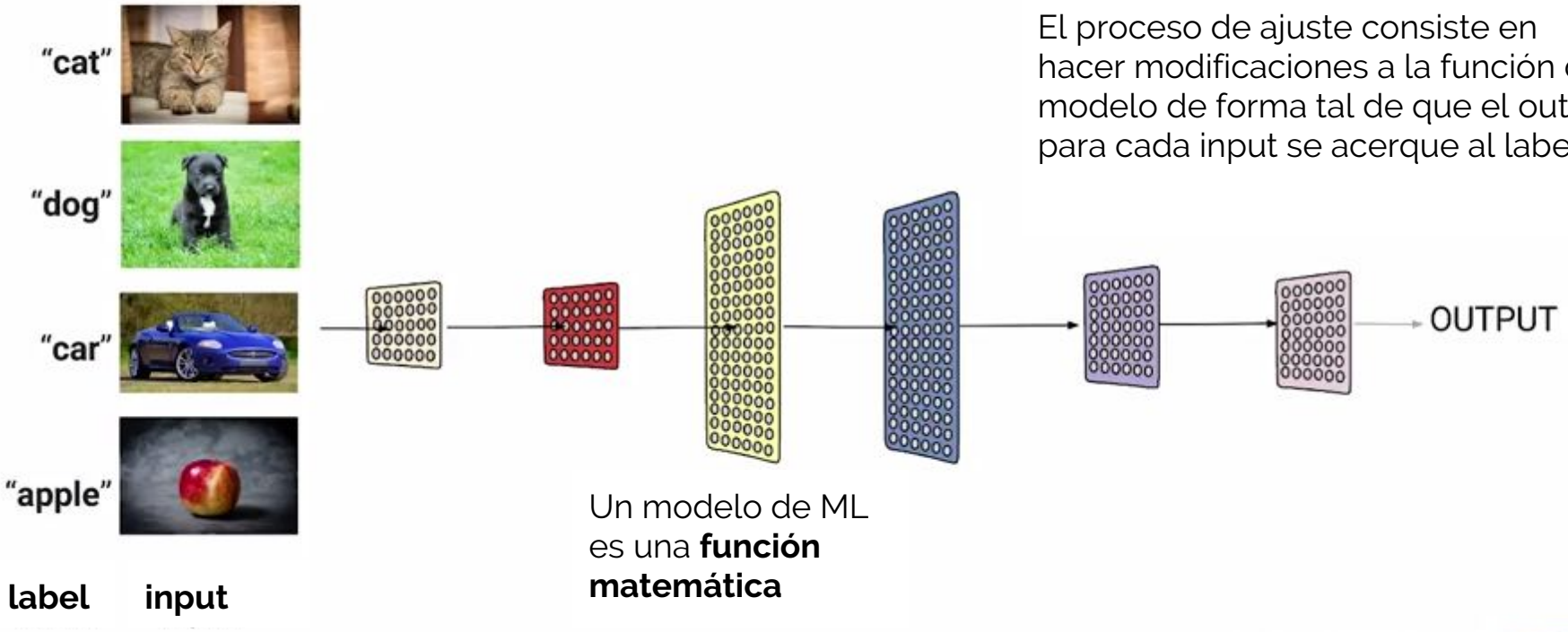
- Aprendizaje Supervisado
  - Clasificación: Predecir labels discretas
  - Regresión: Predecir labels continuas

- Aprendizaje No Supervisado
  - Clustering: Inferir labels en datos no etiquetados
  - Reducción de dimensionalidad: Inferir estructura en datos no etiquetados

IMPORTANTE: La siguiente es sólo una primer aproximación para ayudarnos a construir intuición sobre el tema.

Si algún tema resulta complejo o poco intuitivo, no te preocupes!
Pasaremos el resto del curso estudiando y trabajando con estas técnicas.

Aprendizaje Supervisado

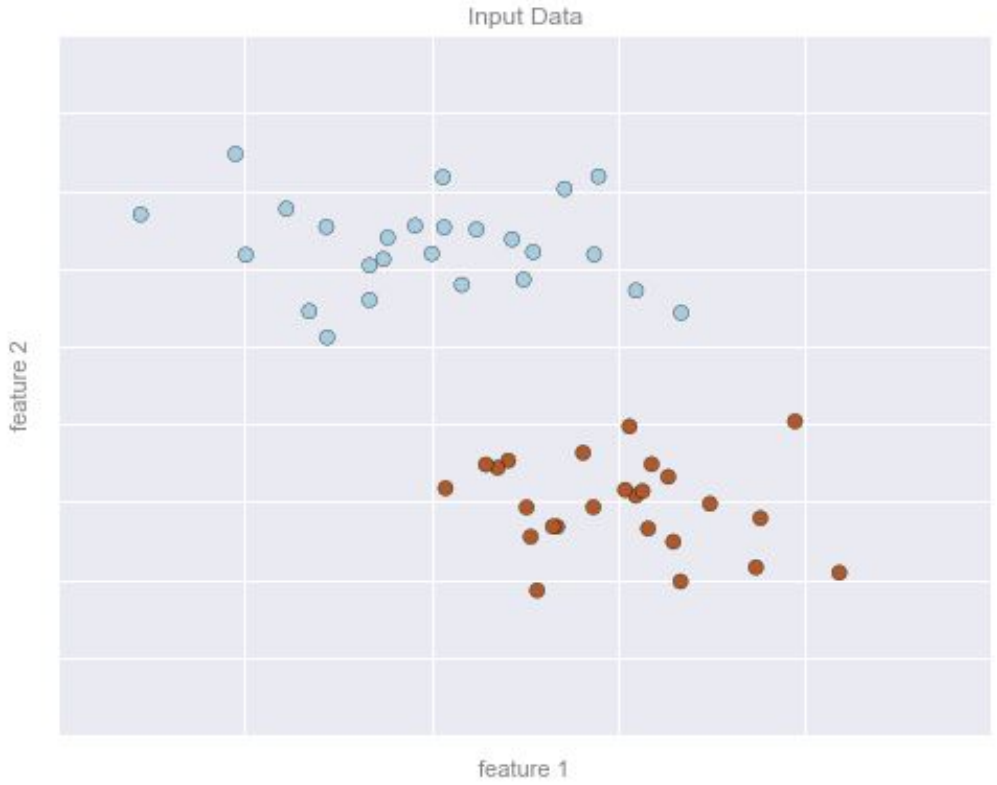


El proceso de ajuste consiste en hacer modificaciones a la función del modelo de forma tal de que el output para cada input se acerque al label

Un modelo de ML es una **función matemática**

Clasificación:

Predecir etiquetas discretas



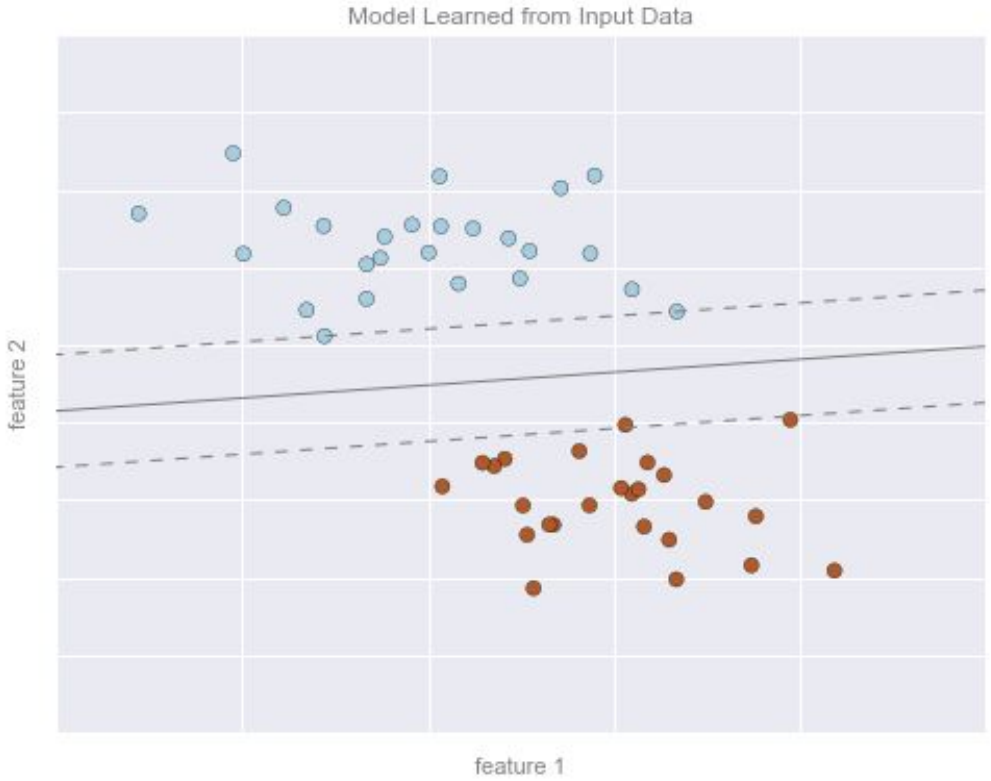
Imaginemos este dataset bi-dimensional. Tenemos dos features por cada punto, representadas por las coordenadas (x, y) en el plano.

Además, tenemos una de dos **etiquetas de clase** por cada punto, representada por el color del punto.

Con esta info queremos **crear un modelo** que nos permita decidir si un **nuevo punto** debería ser etiquetado como azul o rojo.

Hay varios modelos para esta tarea de clasificación, pero elegimos uno extremadamente simple

Asumimos que los dos grupos pueden separarse dibujando una **línea recta** en el plano, tal que los puntos a cada lado de la línea pertenecen al mismo grupo.

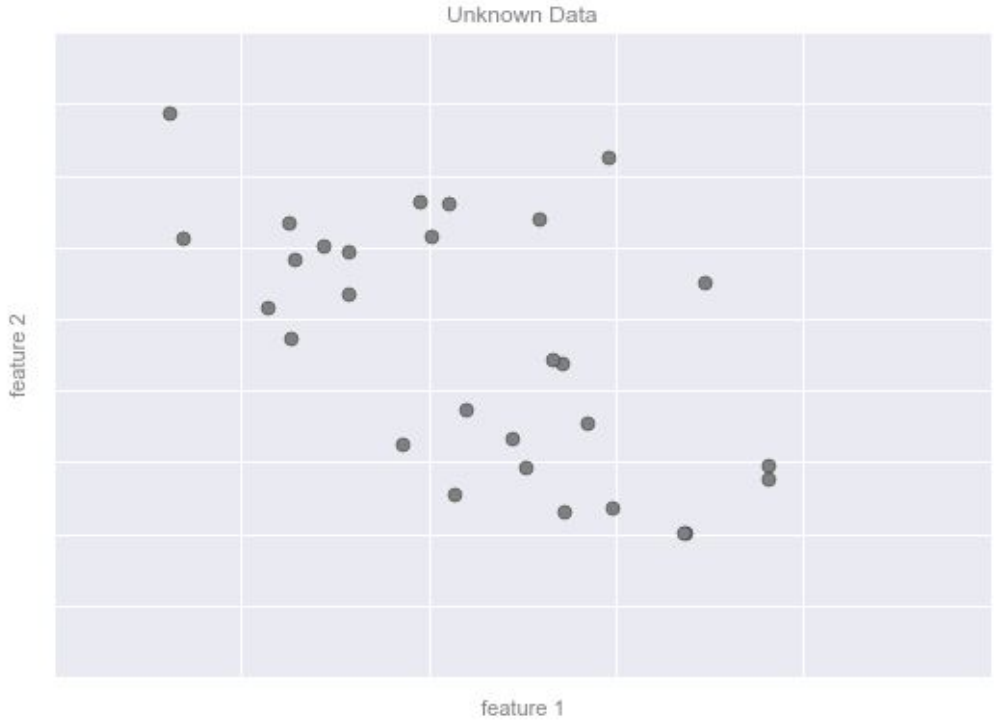


Entonces, el **modelo** es una versión cuantitativa de la afirmación "una línea separa las clases", y los **parámetros del modelo** son los números que describen la ubicación y orientación de esa línea para nuestros datos.

Los **valores óptimos** para estos parámetros del modelo se **aprenden** de los datos, proceso que se conoce usualmente como **entrenar el modelo.**

La figura muestra una representación visual de cómo luce el modelo entrenado para estos datos.

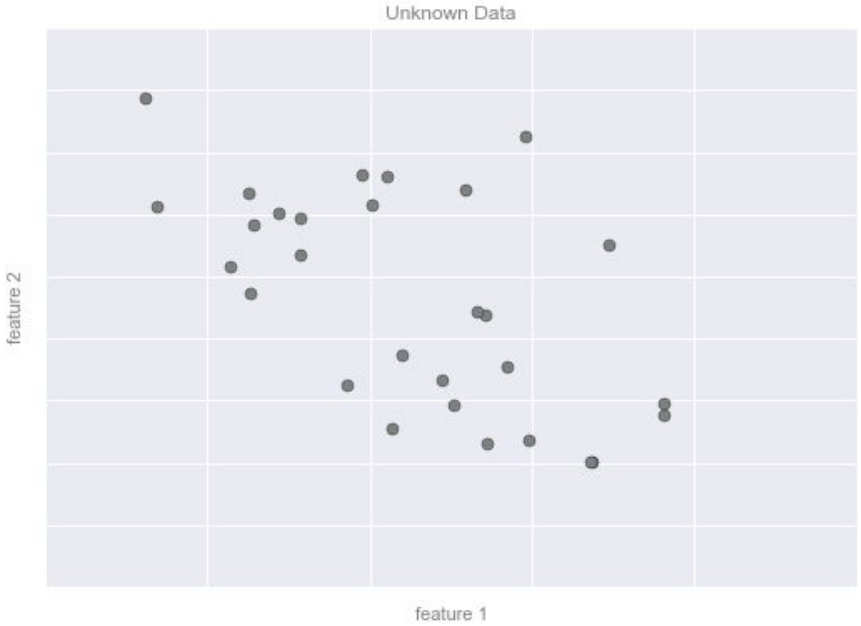
Clasificación: nuevos datos

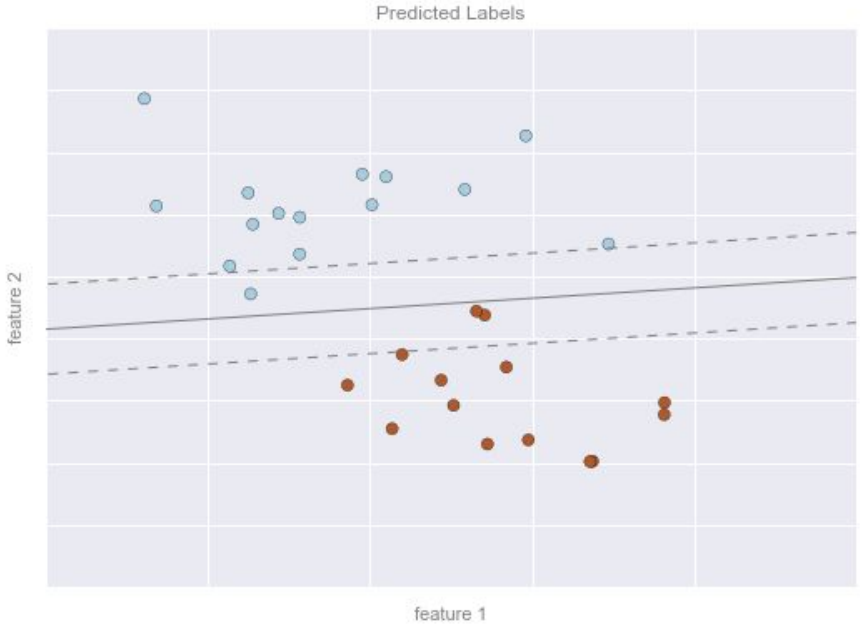


Ahora que este modelo ha sido entrenado, puede generalizarse a nuevos datos sin etiquetas.

Podemos tomar un nuevo dataset, dibujar la línea de nuestro modelo y asignar etiquetas a los nuevos puntos, basándonos en qué lado de la línea quedaron.

Clasificación: PREDICCIÓN





Por ejemplo, lo anterior es similar a la tarea de detección automática de email spam. En este caso podríamos usar la siguiente lista de features y labels:

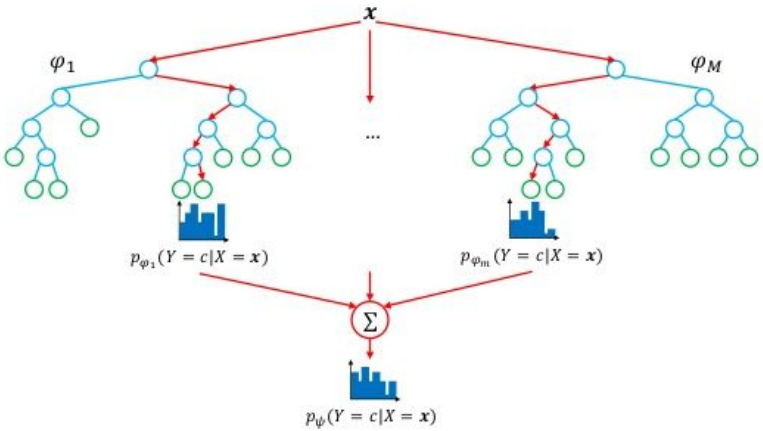
- *feature 1*, *feature 2*, etc. → recuentos normalizados de palabras o frases importantes ("Viagra", "Nigerian prince", etc.)
- *label* --> "spam" o "not spam"

Para el **set de entrenamiento**, estas etiquetas podrían ser determinadas por una inspección individual de una muestra representativa de emails; para el resto de los emails, las labels serían determinadas usando el modelo.

Con un algoritmo de clasificación correctamente entrenado con suficientes features bien construidas (típicamente miles o millones de palabras o frases), este tipo de aproximación puede ser muy efectivo. Veremos un ejemplo de clasificación basada en texto cuando estudiemos **Clasificación usando Naive Bayes.**

Algunos algoritmos de clasificación que discutiremos en detalle son:

- Gaussian naive Bayes
- Support Vector Machines
- Árboles de Decisión y Random Forest



Regresión:

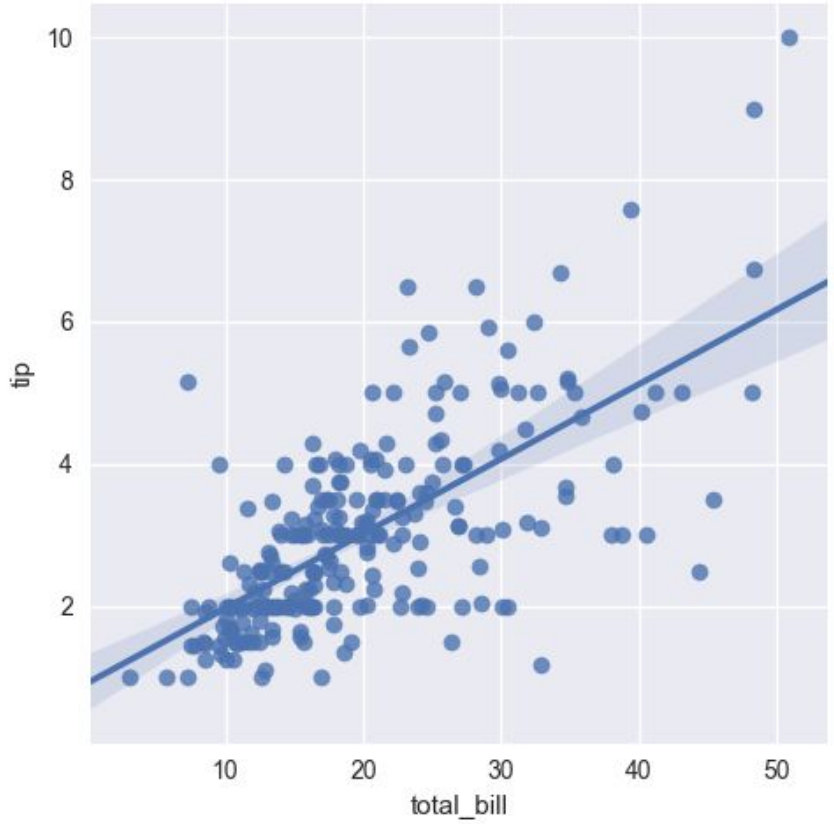
Predecir etiquetas continuas

Regresión: Un ejemplo Simple

Ya vimos algunas regresiones.

Un ejemplo simple con el dataset “tips” incluído en Seaborn:

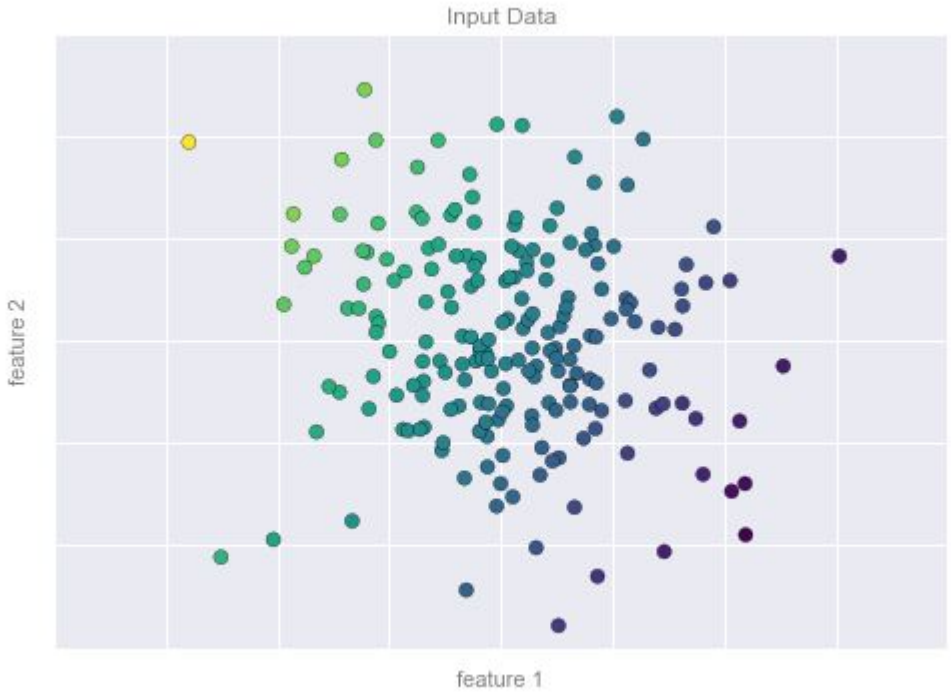
```
tips = sns.load_dataset("tips")

sns.lmplot(x="total_bill", y="tip", data=tips);
```



Regresión: Un ejemplo con dos features

Consideremos el dataset mostrado en la figura. Consiste en un conjunto de puntos, cada uno con una etiqueta continua.

Datos bidimensionales: tenemos dos features describiendo cada punto. El color de cada punto representa el valor contínuo de la etiqueta para esa observación.

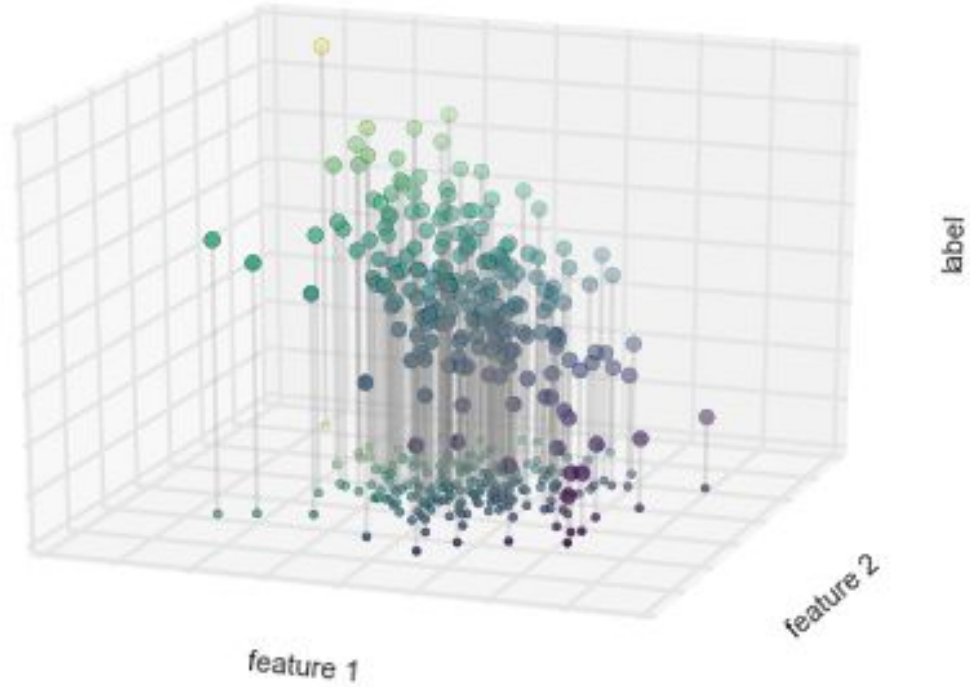


Por ejemplo si queremos predecir performance de alumnos: las variables independientes podrían ser horas de sueño y horas de estudio. La variable dependiente podría ser los scores de un test.

Regresión: Un ejemplo con dos features

Hay varios modelos de regresión posible que podríamos usar para este tipo de datos, pero aquí usaremos una regresión lineal para predecir el valor de las etiquetas de nuevos puntos.

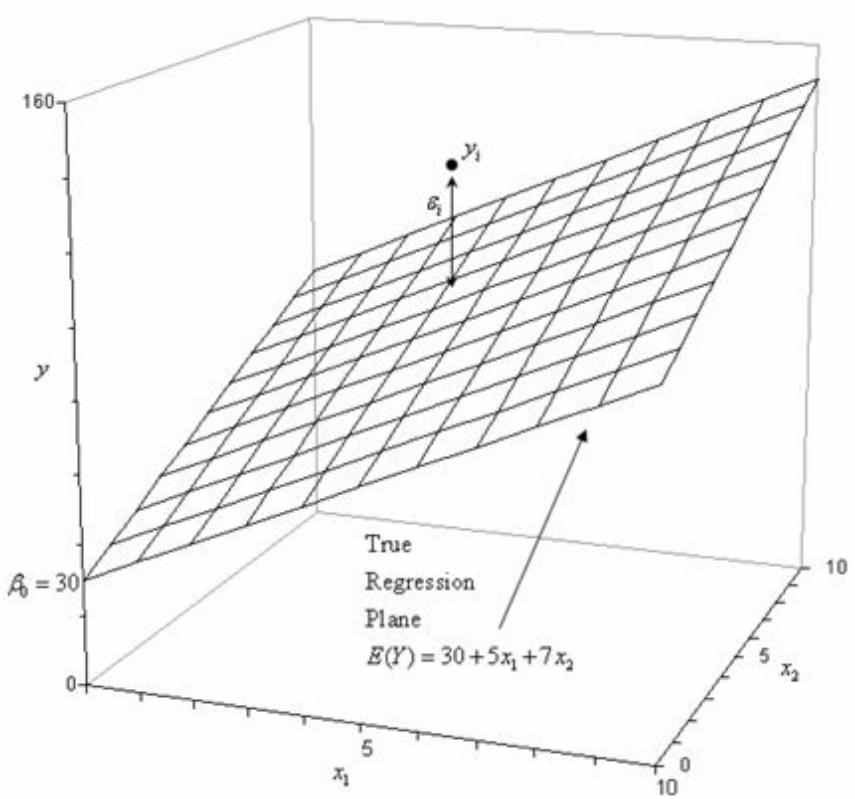
Imaginemos que interpretamos la etiqueta como una tercer dimensión espacial. Entonces podríamos ajustar un plano a nuestros datos.

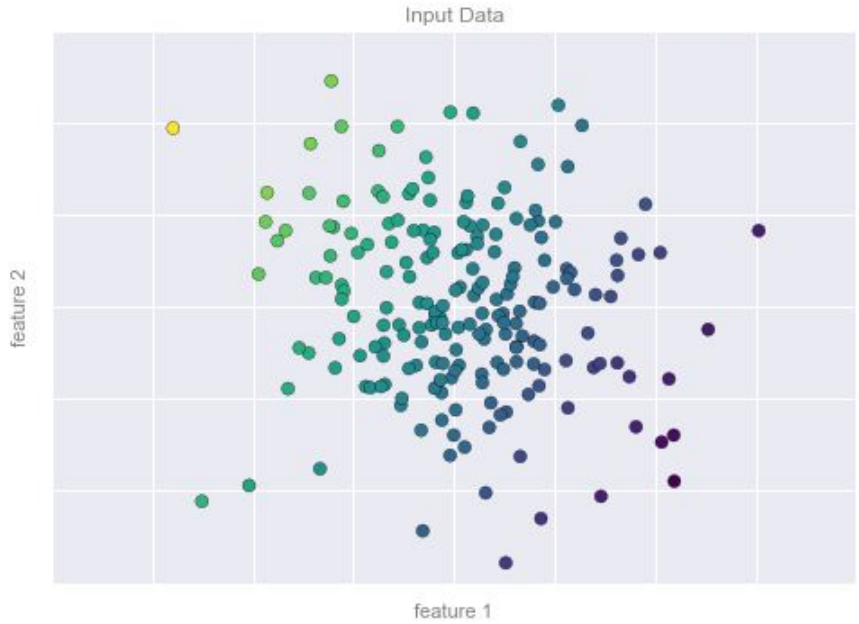


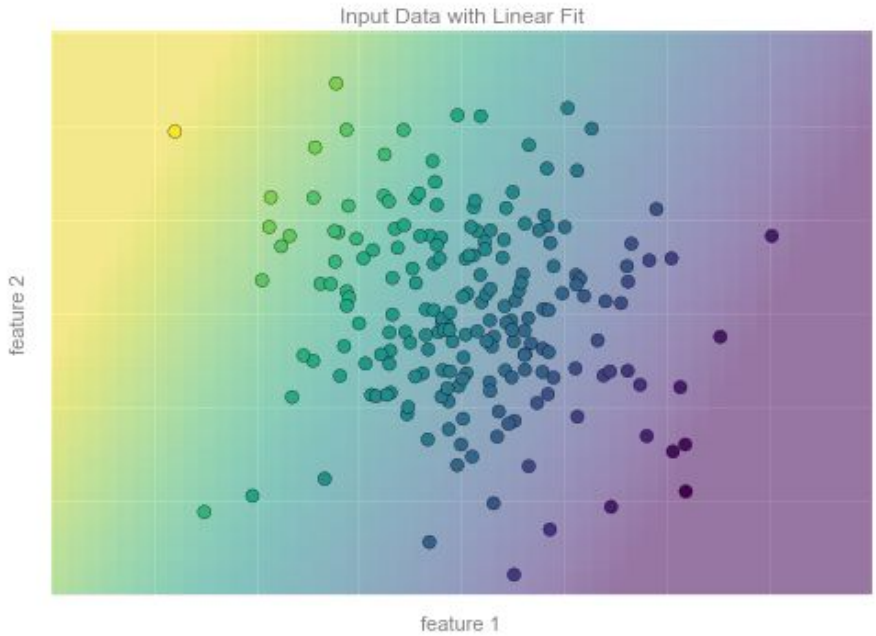
Notar que aquí el plano feature 1 - feature 2 es el mismo que la slide anterior. En este caso, sin embargo, hemos representado el valor contínuo de la etiqueta con un color y su posición en el eje de la tercer dimensión

Parece razonable que ajustar un plano a estos datos en tres dimensiones nos permitirá predecir una etiqueta para cada conjunto de datos nuevos.

Esta es una generalización de alto nivel del conocido problema de ajustar una línea a datos con una feature y una etiqueta.

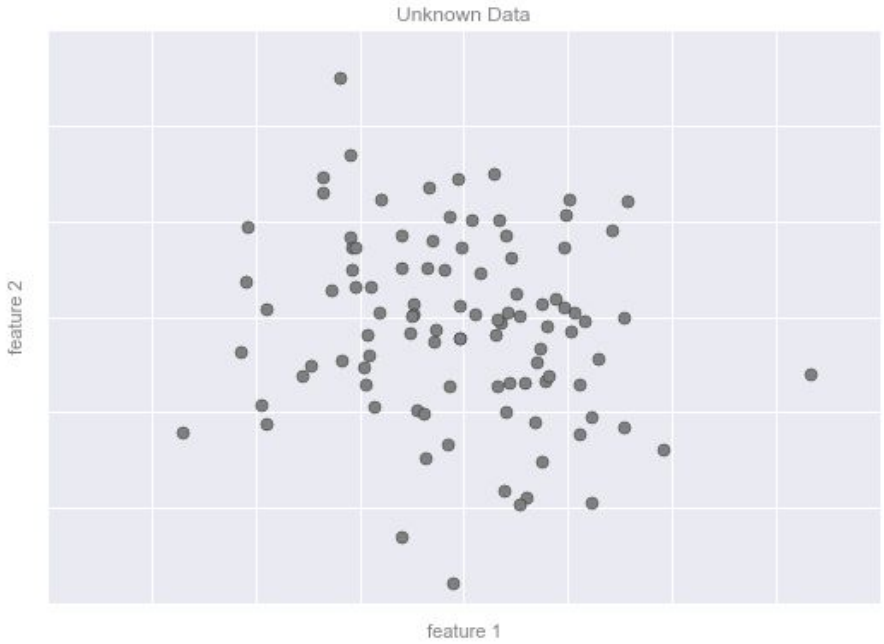


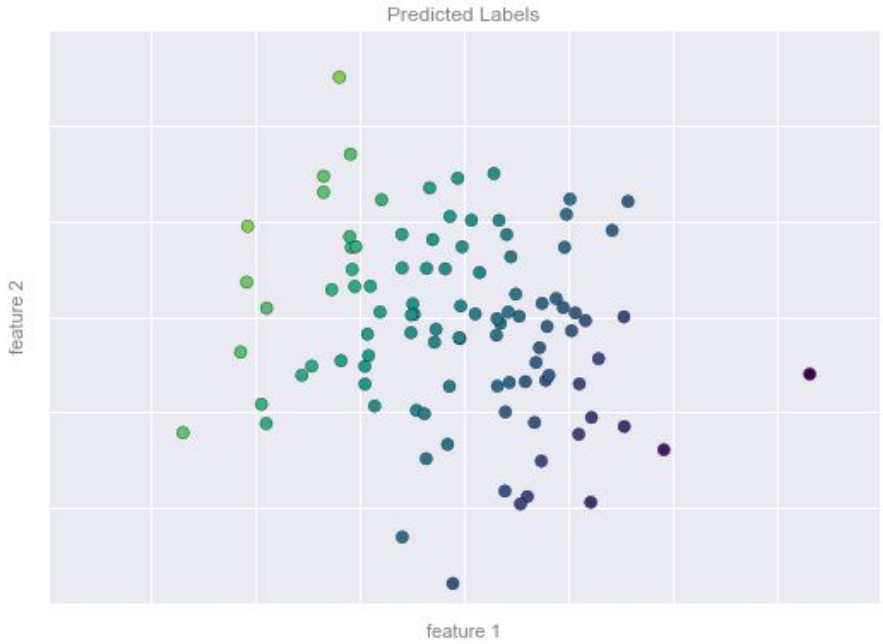




Retornando a la **proyección en dos dimensiones**: vemos los datos y vemos el ajuste lineal, codificado con un contínuo de colores

El plano de ajuste nos da lo que necesitamos para predecir las etiquetas de los nuevos puntos.





Como en el ejemplo de clasificación, esto puede parecer trivial con un número bajo de dimensiones. Pero el poder de estos métodos radica en que pueden ser aplicados y evaluados directamente a casos de datasets con un gran número de dimensiones.

Por ejemplo, podemos computar la **distancia a galaxias** observadas a través de un telescopio - en este caso podemos usar las siguientes features:

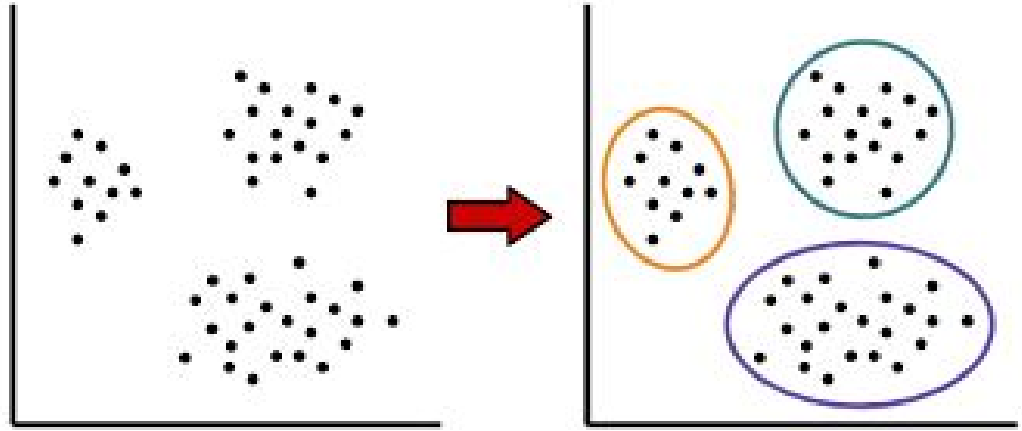
- *feature 1*, *feature 2*, etc. --> brillo de cada galaxia en una de varias longitudes de ondas o colores
- *label* <-- distancia de la galaxia

Las distancias para un pequeño número de galaxias podría determinarse a través de observaciones independientes, usualmente muy costosas. Las distancias al resto de las galaxias podría luego ser estimadas usando un modelo de regresión apropiado, sin la necesidad de emplear los métodos de observación más costosos. En terminología astronómica, esto es conocido como el problema "photometric redshift"

Aprendizaje no supervisado

Las tareas de clasificación y regresión que vimos recién son ejemplos de algoritmos de aprendizaje supervisado, en los cuales intentamos construir un modelo que predice etiquetas para nuevos datos.

El aprendizaje no supervisado involucra modelos que describen los datos **sin referencia a ninguna etiqueta conocida**.



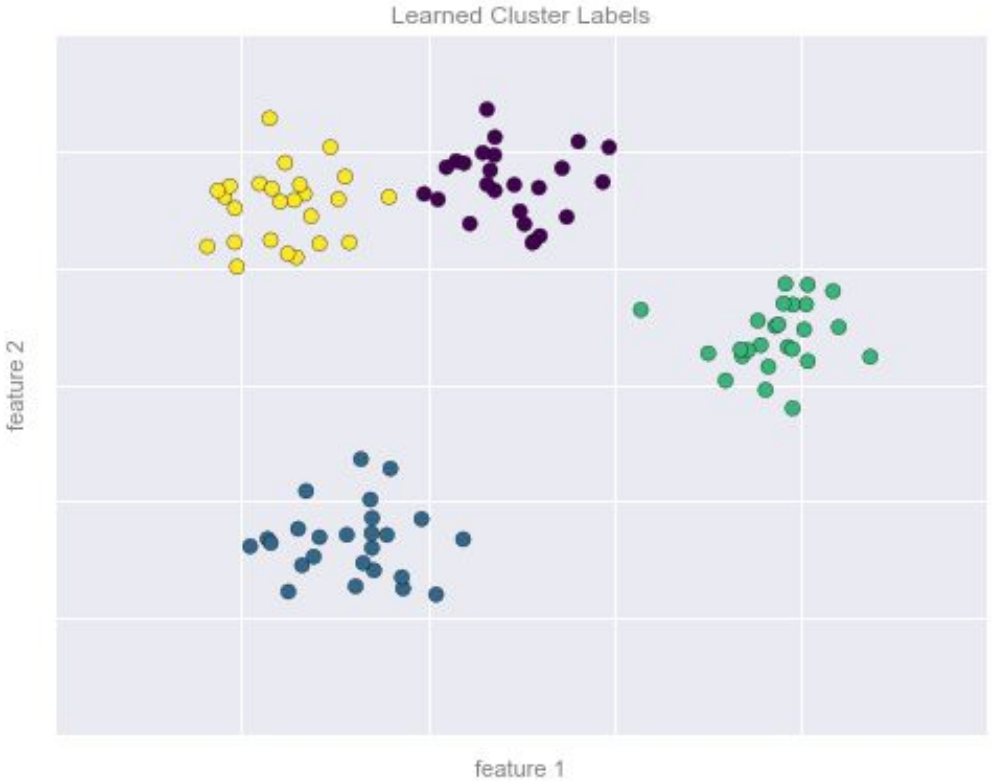
Clustering: Inferir etiquetas en datos no etiquetados

Clustering: Inferir etiquetas en datos no etiquetados

En clustering, a los datos se los asigna automáticamente a algún número de grupos discretos.

Por ejemplo, podríamos tener un dataset bidimensional como el que se muestra en la figura.

Visualmente es evidente que cada uno de estos puntos es parte de un grupo discreto.



Dado este input, un modelo de clustering usará la **estructura intrínseca** de los datos para determinar qué puntos están **relacionados**.

Usando el algoritmo **k-means** (muy rápido e intuitivo), encontramos los cluster mostrados en la figura

k-means ajusta un modelo que consiste en k centros de clusters; se asume que los centros óptimos son aquellos que minimizan la distancia de cada punto a su centro asignado.

Nuevamente, esto parecería ser un ejercicio trivial en dos dimensiones, pero a medida que nuestros datos se vuelven más grandes y complejos, estos algoritmos de clustering pueden ser utilizados para extraer información útil de nuestros datasets.

Vamos a discutir el algoritmo k-means en detalle más adelante. Otros algoritmos importantes que vamos a estudiar son:

- Clustering jerárquico
- DBSCAN

Reducción de dimensionalidad: Inferir estructura en datos no etiquetados

La reducción de dimensionalidad es otro ejemplo de algoritmo no supervisado, en el cual las etiquetas u otra información es inferida de la estructura del propio dataset.

Es una técnica un poco más abstracta que los ejemplos vistos previamente pero, en términos generales, busca obtener, de un dataset complejo, una representación de baja dimensionalidad, que de alguna manera preserve cualidades relevantes del dataset original.

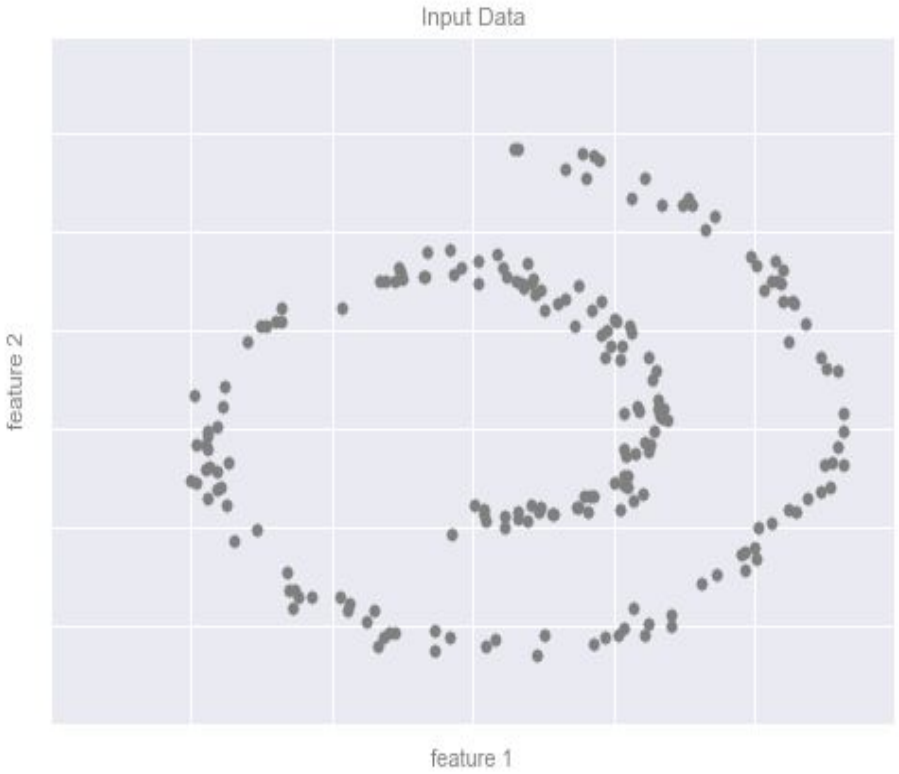
Consideremos este dataset bidimensional.

Visualmente es evidente que hay alguna estructura en estos datos: parece ser una línea unidimensional dispuesta en forma de espiral dentro del espacio bidimensional.

En cierto sentido, podríamos decir que este dataset es ***intrínsecamente* unidimensional.**

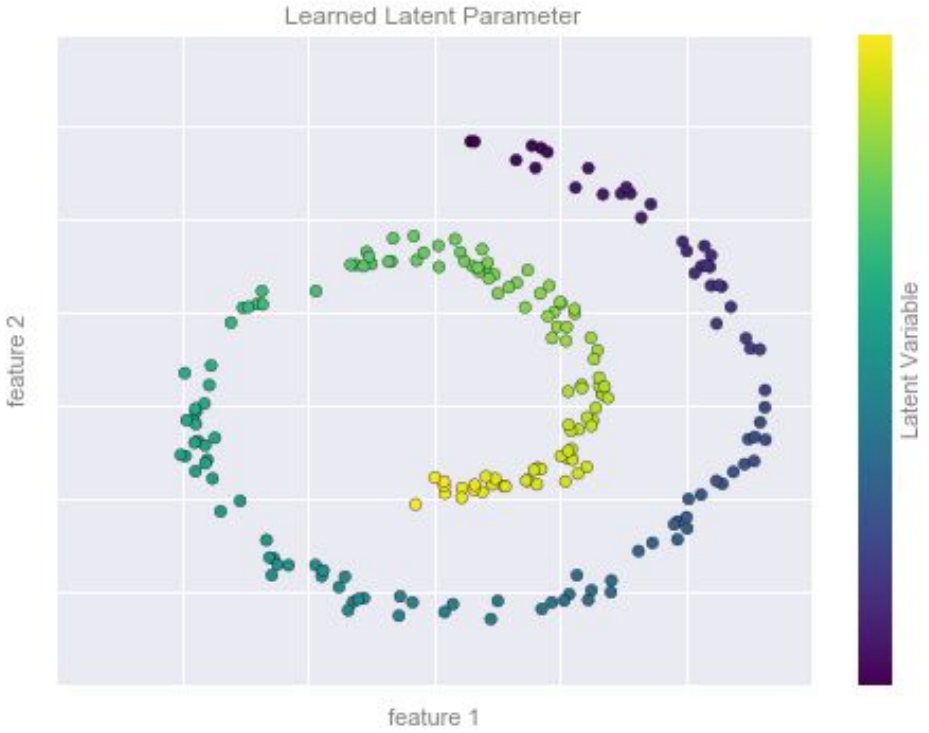
Sin embargo, esta data unidimensional está embebida en **un espacio de mayor dimensionalidad.**

Un modelo apropiado de reducción de dimensionalidad en este caso sería sensible a esta estructura embebida no lineal (nonlinear embedded structure), y sería capaz de extraer una representación de baja dimensionalidad.



La figura muestra una visualización de los resultados del algoritmo Isomap, un ejemplo de un tipo de técnica conocida como manifold learning.

Notar que los colores (que representan la variable latente unidimensional extraída) cambian uniformemente a lo largo de la espiral, lo que indica que el algoritmo detectó efectivamente la estructura observada a simple vista.



Como en los ejemplos previos, el poder de los algoritmos de reducción de dimensionalidad se vuelve evidente en casos de alta dimensionalidad.

Por ejemplo, podríamos querer visualizar relaciones importantes ocultas en datasets que tienen 100 o 1000 features.

Visualizar un dataset con 1000 dimensiones es un desafío y una forma en la que podemos realizar esto más fácilmente es a través de una técnica de reducción de la dimensionalidad para reducir los datos a dos o tres dimensiones

Algunos algoritmos importantes de esta categoría que estudiaremos en el curso son:

- Principal Component Analysis (PCA)
- Manifold Learning
  - Isomap
  - Locally Linear Embedding

Hemos visto algunos ejemplos simple de los tipos básicos de algoritmos de machine learning y qué problemas podrían resolver.

Debe quedar claro que hemos ignorado una enorme cantidad de detalles técnicos que iremos estudiando a lo largo del curso.

En resumen, hemos presentado:

- Aprendizaje Supervisado: Modelos que pueden predecir etiquetas basándose en datos de entrenamiento etiquetados
  - *Clasificación*: Modelos que predicen etiquetas para dos o más categorías discretas
  - *Regresión*: Modelos que predicen etiquetas continuas

- Aprendizaje No Supervisado: Modelos que identifican estructura en datos sin etiquetar.
  - Clustering: Modelos que identifican grupos en los datos
  - Reducción de la dimensionalidad: Modelos que detectan e identifican estructuras de menor dimensión en datos de mayor dimensión.