# Python para el análisis de datos- Lectura 9

Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Enero-2022

# Machine learning

El machine learning o aprendizaje de máquinas, es la rama de la ciencia de datos que analiza y procesa los datos para encontrar predicciones sobre su comportamiento.

El ingeniero de datos, predice los comportamientos de un conjunto de datos o genera patrones para distintos tipos de procesos.

Por ejemplo, una analogía de esto, sería el que usted desarrolle el siguiente juego matemático. La idea es encontrar el número que continúa en la secuencia.

**1,1,2,3,5,8,?**

# Machine learning

El resultado, es bastante conocido por todos, pero lo que hicimos, fue analizar la data, y obtener en base a nuestra experiencia previa; es decir a la data que conocíamos; una relación entre los datos, hasta conseguir definir una **regla**:

???

Esta regla la definimos luego de analizar los resultados y contrastar con la data que teníamos.

Un resultado distinto fuese si en lugar de ese set de datos, tuviésemos ahora :

**1,1,2,3,5,8,10,16,17,27,?**.

# Machine learning

Nuestra predicción era muy diferente y además con el segundo dataset nuestro resultado es totalmente distinto.

# Machine learning

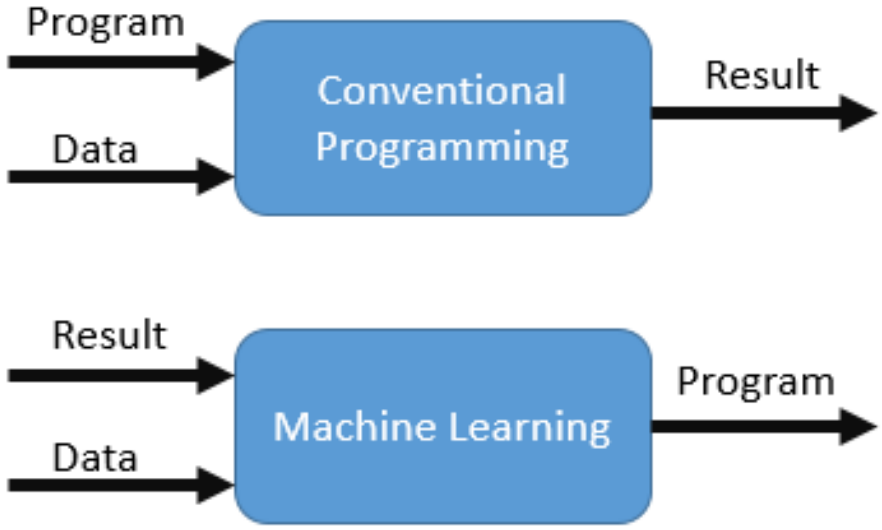
Nuestra predicción era muy diferente y además con el segundo dataset nuestro resultado es totalmente distinto.

Algo similar sucede con el machine learning. Ofrecemos un dataset (conjunto de datos) inicial al computador, que mediante algoritmos de optimización estimará reglas "invisibles" al científico de datos y que determinarán una predección o una relación entre los datos mostrados.

# Machine learning

Nuestra predicción era muy diferente y además con el segundo dataset nuestro resultado es totalmente distinto.

Algo similar sucede con el machine learning. Ofrecemos un dataset (conjunto de datos) inicial al computador, que mediante algoritmos de optimización estimará reglas "invisibles" al científico de datos y que determinarán una predección o una relación entre los datos mostrados.



# Tipos de Machine learning

La clasificación más general de los algoritmos de Machine learning es:

- Aprendizaje supervisado: Aqui se encuentran los algoritmos que involucran relación entre caracterísiticas conocidas y alguna etiqueta o predicción "desconocida". En este grupo se encuentran tareas de clasificación por etiqueta y regresión.

# Tipos de Machine learning

La clasificación más general de los algoritmos de Machine learning es:

- Aprendizaje supervisado: Aqui se encuentran los algoritmos que involucran relación entre caracterísiticas conocidas y alguna etiqueta o predicción "desconocida". En este grupo se encuentran tareas de clasificación por etiqueta y regresión.
- Aprendizaje no supervisado: Involucra algoritmos que no necesitan relacionarse con ninguna etiqueta. Aquí se incluyen tareas como el ordenamiento por grupos y la reducción dimensional.

# Tipos de Machine learning

Otra forma de clasificar al ML es mediante sus métodos de aplicación en el modelo:

- Entrenamiento por Lote entero: También llamado aprendizaje estadístico. Este tipo de entrenamiento, ajusta al modelo cuando se ha entrenado todo el conjunto de datos. Este tipo de estructura es simple, pero muy lenta para conjuntos de datos grandes.

# Tipos de Machine learning

Otra forma de clasificar al ML es mediante sus métodos de aplicación en el modelo:

- Entrenamiento por Lote entero: También llamado aprendizaje estadístico. Este tipo de entrenamiento, ajusta al modelo cuando se ha entrenado todo el conjunto de datos. Este tipo de estructura es simple, pero muy lenta para conjuntos de datos grandes.
- Entrenamiento por mini Lotes: El tipo de entrenamiento usado para poder trabajar con cantidades grandes de datos. En este caso, los parámetros del modelo van cambiando conforme los minilotes se entrenan.

# Tipos de Machine learning

- Entrenamiento Activo: Este tipo de entrenamiento se "ajusta" con decisiones tomadas por el usuario, que entrenan en tiempo real al modelo. Por ejemplo un verificador de correo Spam, mejora cuando un usuario elige qué correo es spam manualmente.

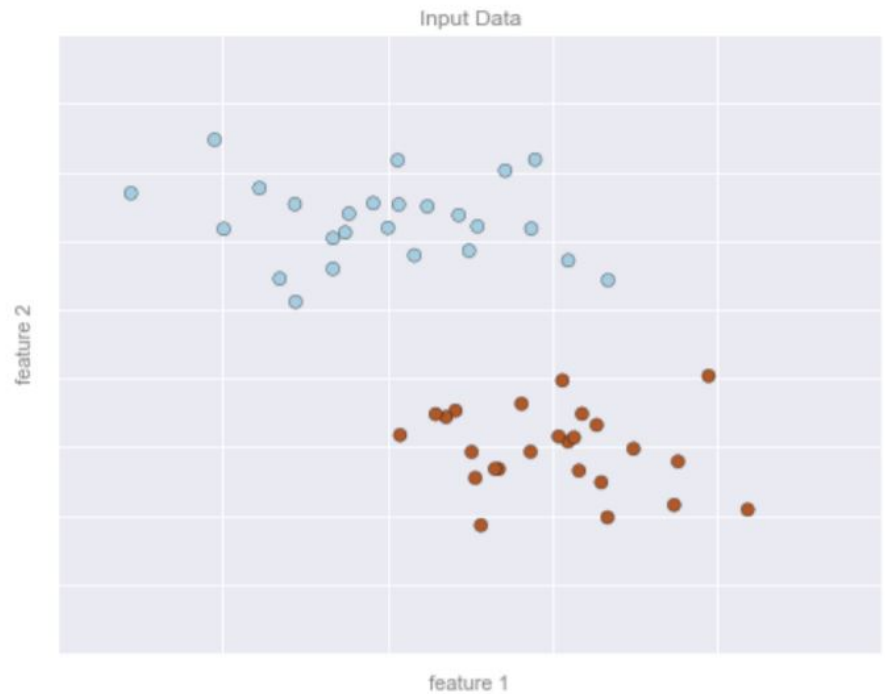
# Tipos de Machine learning

- Entrenamiento Activo: Este tipo de entrenamiento se "ajusta" con decisiones tomadas por el usuario, que entrenan en tiempo real al modelo. Por ejemplo un verificador de correo Spam, mejora cuando un usuario elige qué correo es spam manualmente.
- Entrenamiento por refuerzo: Este tipo de entrenamiento usa su entorno y la interacción con el sistema para mejorar la aproximación. Usando un eficiente bucle de retribución, las acciones cambian según el contexto.

# Aprendizaje supervizado

El aprendizaje supervizado necesita de 2 grupos de datos: Las características o **features** que definen el sistema y las etiquetas o **etiquetas** que son la respuesta del sistema.

Por ejemplo en una tarea de clasificación tenemos 2 o más grupos de datos que requieren clasificarse.



# Aprendizaje supervizado

Una tarea de clasificación supervisada, necesita conocer de antemano, en un conjunto de entrenamiento, las etiquetas correspondientes a cada grupo de datos, para poder hacer una clasificación.

El otro tipo de tarea es de regresión. En este caso, un conjunto de datos dispersos se analizan con el fin de realizar una regresión de datos.

# Aprendizaje supervizado

Tanto la tarea de clasificación como la de regresión tienen los siguientes pasos:

- Limpiar y organizar los datos: Se tiene que tener claramente definido que datos, dentro del conjunto de datos tiene que ser el **target** o **label** y cuáles son los **features** y separarlos. Además de limpiar los datos (Eliminar nulos, verificar normalización, etc.)

# Aprendizaje supervizado

Tanto la tarea de clasificación como la de regresión tienen los siguientes pasos:

- Limpiar y organizar los datos: Se tiene que tener claramente definido que datos, dentro del conjunto de datos tiene que ser el **target** o **label** y cuáles son los **features** y separarlos. Además de limpiar los datos (Eliminar nulos, verificar normalización, etc.)
- Separar el conjunto de datos: Luego de haber definido cuales son nuestros features y nuestros target, el siguiente paso es separar el conjunto de datos en 2 grupos: Un grupo de prueba o entrenamiento y otro para validación.

# Aprendizaje supervizado

Tanto la tarea de clasificación como la de regresión tienen los siguientes pasos:

- Limpiar y organizar los datos: Se tiene que tener claramente definido que datos, dentro del conjunto de datos tiene que ser el **target** o **label** y cuáles son los **features** y separarlos. Además de limpiar los datos (Eliminar nulos, verificar normalización, etc.)
- Separar el conjunto de datos: Luego de haber definido cuales son nuestros features y nuestros target, el siguiente paso es separar el conjunto de datos en 2 grupos: Un grupo de prueba o entrenamiento y otro para validación.

# Aprendizaje supervizado

- Elegir el modelo: El tercer paso es elegir qué modelo o algoritmo de Machine learning vamos a usar. En este caso, podemos elegir entre muchos tipos de algoritmos y normalmente para verificar cuál es el que da mejor resultado, se realiza una comparación entre ellos.

# Aprendizaje supervizado

- Elegir el modelo: El tercer paso es elegir qué modelo o algoritmo de Machine learning vamos a usar. En este caso, podemos elegir entre muchos tipos de algoritmos y normalmente para verificar cuál es el que da mejor resultado, se realiza una comparación entre ellos.
- Verificar resultados: Luego de realizar el entrenamiento se verifican resultados y se analiza cómo mejorar la precisión del modelo, con lo que se retornaría al primer paso. En cambio si el modelo cumple nuestras espectativas, se consideraría para usarse.

# Scikit learn

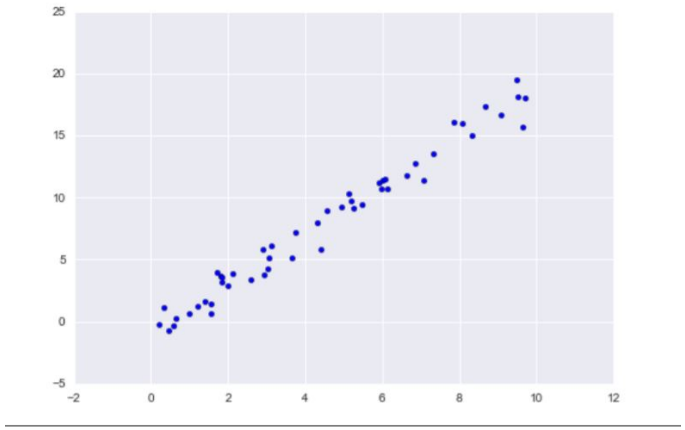
Esta librería permite generar modelos de machine learning y algoritmos teniendo en cuenta solo el conocimiento de los **hiperparámetros** del modelo.

Los hiperparámetros son las "condiciones de frontera" que el ingeniero de datos tiene que colocar al principio del entrenamiento y que se tienen en discusión, de acuerdo al tipo de tarea y de modelo que se requiera. Discutiremos el modelo de regresión lineal.

# Regresión lineal

Discutiremos ahora el modelo de regresión lineal.

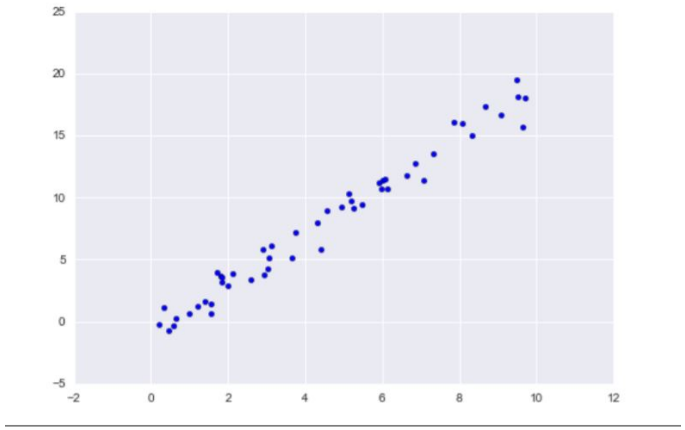
La idea central de este modelo, es generar de acuerdo a un conjunto de entrenamiento de datos una predicción de manera lineal respecto a la data futura.

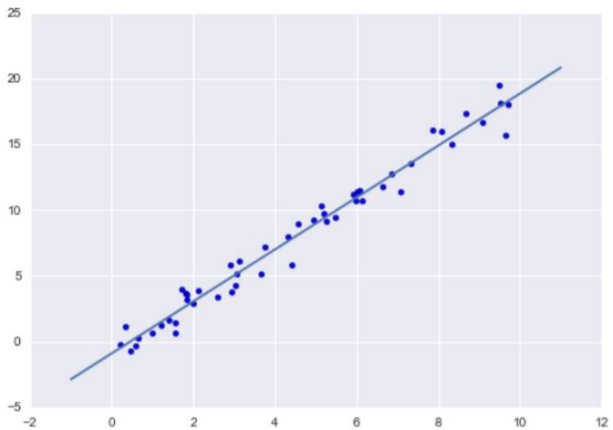


# Regresión lineal

Discutiremos ahora el modelo de regresión lineal.

La idea central de este modelo, es generar de acuerdo a un conjunto de entrenamiento de datos una predicción de manera lineal respecto a la data futura.



# Regresión lineal

El problema en esta tarea es encontrar la mejor línea recta que ajuste a los datos. El objetivo es encontrar una regla de la forma:

**???**

Para ello se puede hacer uso de algunas técnicas como por ejemplo:

**Método de los mínimos cuadrados ordinarios**

Este método funciona a través de una inferencia de target inicial y luego un ajuste optimizando la función de error dada por:

# Regresión lineal

El problema en esta tarea es encontrar la mejor línea recta que ajuste a los datos. El objetivo es encontrar una regla de la forma:

**???**

Para ello se puede hacer uso de algunas técnicas como por ejemplo:

**Método de los mínimos cuadrados ordinarios**

Este método funciona a través de una inferencia de target inicial y luego un ajuste optimizando la función de error dada por:

# Computarización de regresión lineal

Para lectura de los datos y manejo de datos usamos pandas.

`X` será el vector o matriz de **features** y `y` el vector **target**.

Para la separación en conjunto de entrenamiento y prueba, usamos el método de scikit learn train_test_split, un método importante ya que se usa con varias librerías diferentes.

Su sintaxis es:

```
from sklearn.model_selection import train_test_split
Xtrain,Xtest,ytrain,ytest = train_test_split(X, y,
random_state=n)
```

Donde random_state controla el ordenamiento aleatorio de datos. Para varias pruebas, es recomendable tenerlo fijado.

# Computarización de regresión lineal

Para usar el modelo de mínimos cuadráticos se utiliza el constructor:

```
from sklearn import linear_model
model1 = linear_model.LinearRegression()
```

Para ajustar el modelo, es decir, entrenarlo de acuerdo al conjunto de entrenamiento usamos:

```
model1.fit(Xtrain,ytrain)
```

Para hacer la validación de nuestro modelo, podemos hacer predicciones "controladas" con la data de validación:

```
ypre = model1.predict(Xtest)
```

# Resultados

Para conocer los coeficientes y el intercepto del modelo usamos:

```
ypre = model1.coef_
ypre = model1.intercept_
```

Para conocer qué tan bueno es el modelo, usamos los métodos mean_squared_error y r2_score del módulo metrics de sklearn.

```
from sklearn.metrics import mean_squared_error,
r2_score
error = mean_squared_error(ytest, ypred)
puntaje = r2_score(ytest, ypred)
```

Mientras más alto es el puntaje y menor el error, mejor es el modelo encontrado. Ver problema 1