01-Aprendizaje supervisado

November 18, 2019

0.0.1 Aprendizaje Supervisado

Más adelante definiremos de manera formal qué es el aprendizaje supervisado, comencemos ahora con un ejemplo: Supongamos que quieres predecir precios de casas, y tenemos el siguiente conjunto de datos.

In [1]:
```
from pylab import *
```

In [2]:
```
# Tamaño de las casas en pies cuadrados
SIZE = array([533, 540, 800, 990, 1250, 1460, 1600, 1700, 1850, 1990, 2250])

# Precio de las casas en miles ($)
PRICE = array([100, 150, 210, 230, 296, 275, 310, 298, 330, 320, 310])
```

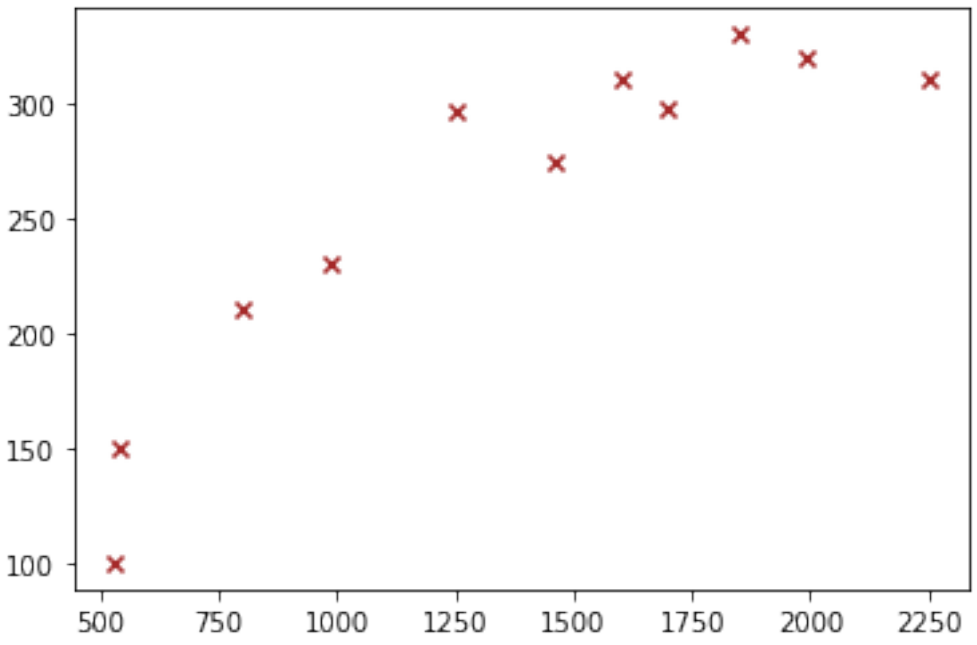
In [3]:
```
print(SIZE.shape)
```

```
(11,)
```

In [4]:
```
print(PRICE.shape)
```

```
(11,)
```

In [5]:
```
scatter(SIZE, PRICE, marker="x", c="brown")
show()
```



Tenemos en el eje horizontal, el tamaño de distintas casas en pies cuadrados, y en el eje vertical su respectivo precio en miles. Entonces con estos datos digamos que tienes una casa de 750 pies cuadrados y quieres saber cuánto dinero se puede pedir por ella. Entonces, £cómo puede ayudarte un algoritmo de aprendizaje? Algo que prodría hacer el algoritmo de aprendizaje es trazar una línea recta a través de los datos, y en base a la ecuación de dicha recta podemos predecir el precio de la casa.

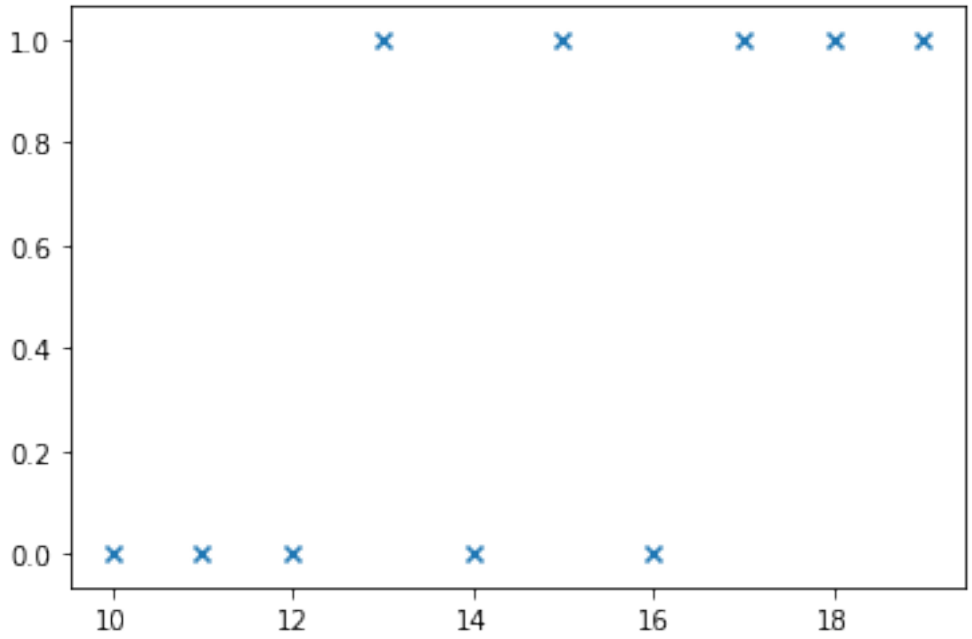
Pero tal vez este no es el único algoritmo de aprendizaje que puedes utilizar. Puede haber uno mejor. Por ejemplo, en lugar de trazar una línea recta en los datos, podríamos decidir que es mejor insertar una función que describa mejor la relación entre el tamaño y el precio de la casa. Pudiera ser una función cuadrática o un polinomio de segundo grado.

Este es un ejemplo de algoritmo de aprendizaje supervisado. Y el término aprendizaje supervisado se refiere al hecho de que le dimos al algoritmo un conjunto de datos donde se daban las "respuestas correctas". Es decir, le dimos un conjunto de datos de casas en los que para cada ejemplo del conjunto de datos, se dijo cuál era el precio correcto, el precio real de esa casa y la tarea del algoritmo solo fué generar más "respuestas correctas", como para predecir el precio de tu casa.

Para definir un poco más la terminología, esto también se denomina un problema de regresión y con esto nos referimos a que queremos predecir un resultado de valor continuo. Es decir, el precio.

Ahora pasemos a otro ejemplo de aprendizaje supervisado. Digamos que quieres revisar expedientes médicos y tratar de predecir cáncer de mama como maligno o benigno.

```
In [6]: tumor = array([0, 0, 0, 1, 0, 1, 0, 1 ,1 , 1])

In [7]: scatter(range(10, tumor.shape[0]+10), tumor, marker="x")
        show()
```



Supongamos que nuestro conjunto de datos se ve de la manera anterior, donde en el eje horizontal tenemos el tamaño del tumor y esta marcando 0 si es benigno y 1 si es maligno. Así que en este ejemplo tenemos cinco tumores benignos que se muestran abajo y cinco tumores malignos que se muestran con un valor del eje vertical de 1. La pregunta para un algoritmo de machine learning sería: £puedes calcular cuál es la probabilidad, de que un tumor sea maligno vs benigno? Este es un ejemplo de un problema de clasificación, el término clasificación se refiere al hecho de que intentamos predecir un resultado de valor discreto: cero o uno, maligno o benigno. Un problema de clasificación va más allá de tan solo dos posibles casos, para este ejemplo pudieramos encontrarnos con la situación de que existan varios tipos de cáncer, entonces tal vez tendríamos 0 para benigno y de $1 - n$ para todos los posibles tipos de cáncer que se estén analizando.

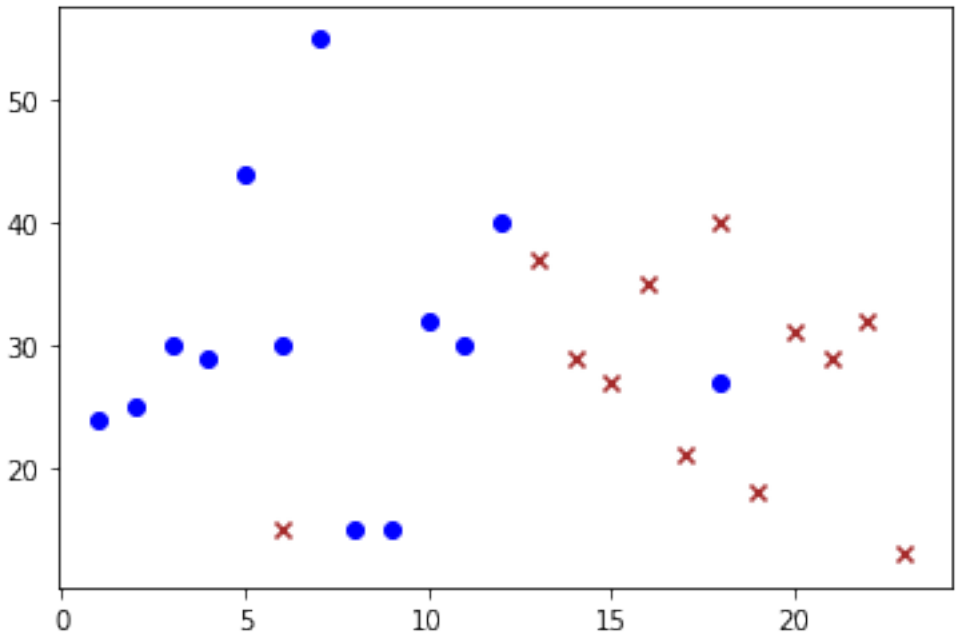
En otros problemas de aprendizaje en muchas ocasiones tenemos más de una característica, más de un atributo. Por ejemplo digamos que en lugar de solo saber el tamaño del tumor, también conocemos la edad del paciente y el tamaño del tumor. En ese caso tal vez tu conjunto de datos se verá así

```
In [8]: edad = array(10 * np.random.randn(24), dtype='int') + 30

        print(edad, edad.shape)
```

```
[24 25 30 29 44 30 55 15 15 32 30 40 37 29 27 35 21 40 18 31 29 32 13 32] (24,)
```

```
In [9]: scatter(range(1, 7), edad[:6], marker='o', c="blue")
        scatter(6, edad[7], marker='x', c="brown")
        scatter(range(7, 13), edad[6:12], c="blue")

        scatter(range(13, 19), edad[12:18], marker='x', c="brown")
```

```
scatter(18, edad[14], marker='o', c="blue")
scatter(range(19, 24), edad[18:23], marker='x', c="brown")
show()
```



Con un conjunto de datos como este, lo que puede hacer el algoritmo de aprendizaje es trazar la línea recta en los datos para tratar de separar los tumores malignos de los benignos.

Con algoritmos más avanzados de machine learning podemos tratar con problemas que tienen una infinita cantidad de características.