LinearRegression como algoritmo de regresión lineal

Empezaremos por el modelo más sencillo , que es ajustar los datos a una línea para pasar luego a ver diferentes modelos de clasificación en orden creciente de complejidad en subsiguientes entradas.

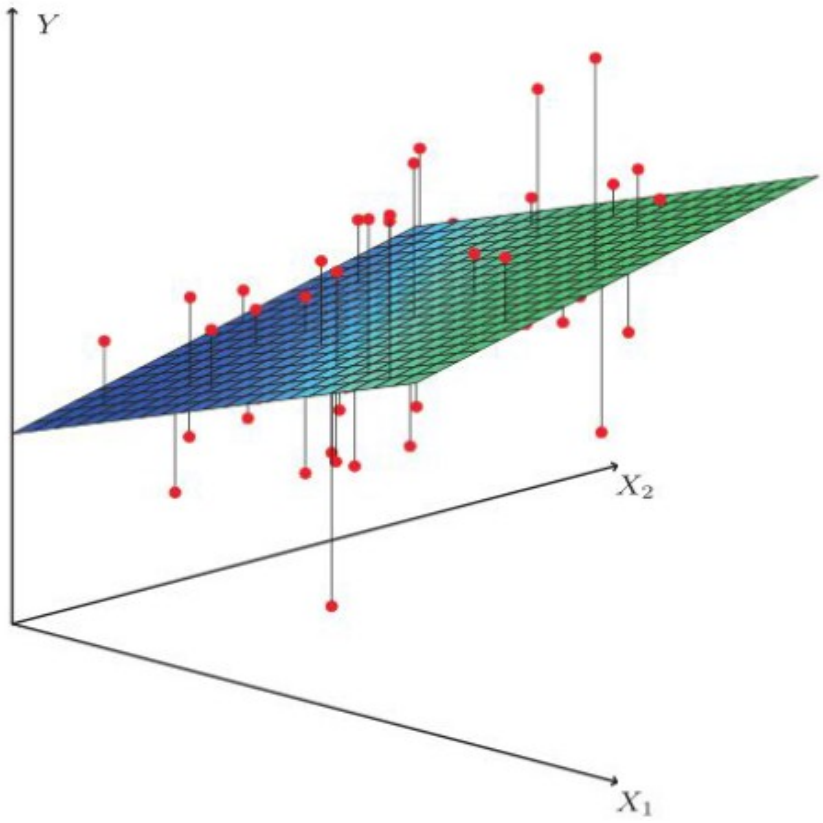
La regresión lineal se basa en buscar la recta que modele la tendencia de los datos y, según ella, predecir cualquier otro dato en el futuro.

El método es bien conocido en varios campos y quizás es el más natural de aprender, ya que la idea de **trazar la recta que mejor describe la relación entre puntos de un plano** parece familiar e intuitivo.

El objetivo es que los datos se adapten a una recta.

La idea de la regresión es tener un conjunto de datos de entrada y explicar o describir las salidas como una combinación lineal de los datos de entrada.

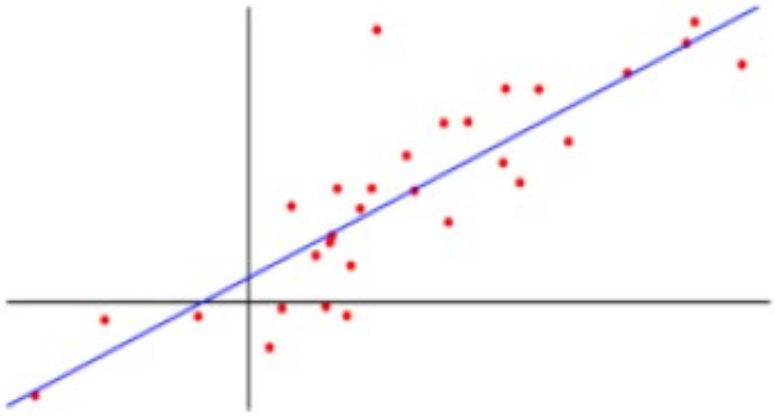
Al final el objetivo de la regresión lineal es encontrar el plano que mejor satisface ciertas condiciones para asegurar que predice o explica lo mejor posible los datos.



El modelo de regresión lineal se utiliza para estimar los valores reales (costo de las viviendas, el número de llamadas, ventas totales, etc.) basados en variables continuas. La idea es tratar de establecer la relación entre las variables independientes y dependientes por medio de ajustar una mejor línea recta con respecto a los puntos. Esta línea de mejor ajuste se conoce como línea de regresión.

La regresión lineal, o mínimos cuadrados ordinarios (OLS), es el método lineal más sencillo y clásico para la regresión. La regresión lineal encuentra los parámetros que minimizan el error cuadrático medio entre las predicciones y los verdaderos objetivos de regresión, y, en el conjunto de entrenamiento.

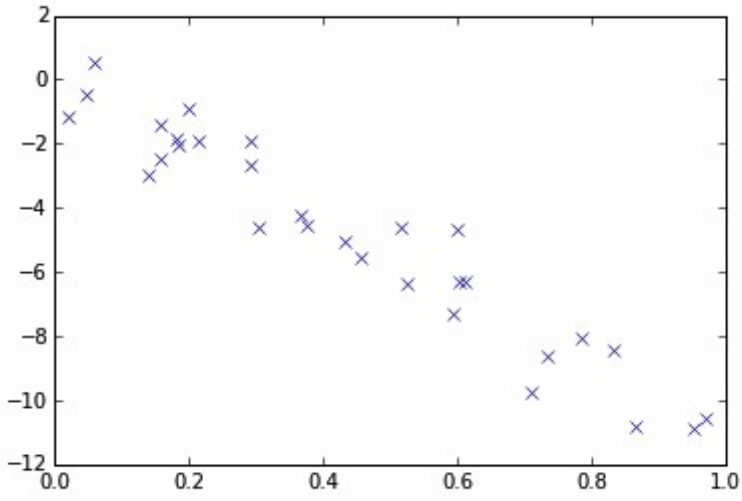
El error cuadrático medio es la suma de las diferencias cuadradas entre las predicciones y los valores reales. La regresión lineal no tiene parámetros, lo cual es una ventaja, pero tampoco tiene forma de controlar la complejidad del modelo.



```
rng = np.random.RandomState(42)
X = rng.uniform(size=(30, 1))
a = rng.normal(scale=10)
b = rng.normal()

y_clean = np.dot(X, a).ravel() + b
y = y_clean + rng.normal(size=len(y_clean))
plt.plot(X[:, 0], y, 'x')
```

```
[<matplotlib.lines.Line2D at 0x16d4f588>]
```




```
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import LinearRegression

lr = LinearRegression().fit(X, y)
y_pred = lr.predict(X)
plt.plot(X[:, 0], y, 'x')
plt.plot(X[:, 0], y_pred)
```

```
[<matplotlib.lines.Line2D at 0x16383eb8>]
```