



**Inteligencia Artificial**  
**Grado en Ingeniería Informática en Sistemas de Información**  
**ENSEÑANZAS PRÁCTICAS Y DE DESARROLLO**  
**EPD2.2: Machine Learning – Regresión Lineal Multivariable**

## Objetivos

- Implementación en Python de un algoritmo de Regresión Lineal con múltiples variables y su aplicación a datos.

## Bibliografía Básica

Machine Learning. Tom Mitchell. MacGraw-Hill, 1997

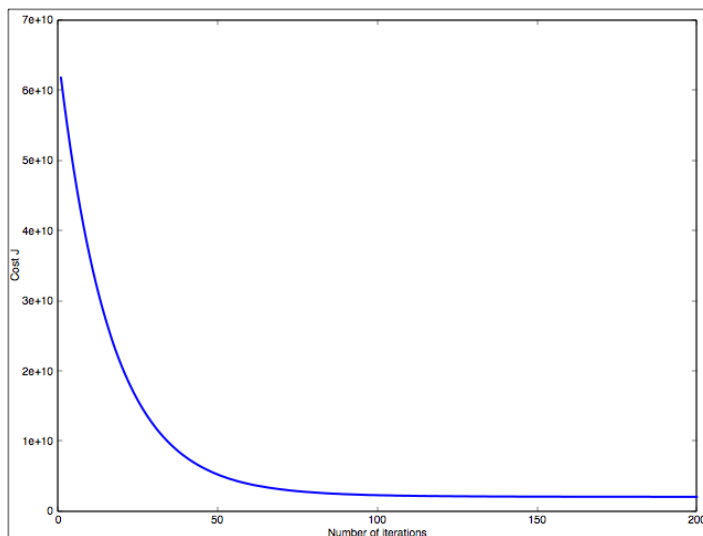
## Ejercicios

Implementar regresión lineal con múltiples variables para predecir los precios de casas. Suponer que queremos vender nuestra casa y queremos saber cuál sería un buen precio de mercado. Una opción sería recolectar información de ventas de casas recientes y crear un modelo de precios de casas.

El fichero `ex1data2.txt` contiene un conjunto de entrenamiento de precios de casas en una ciudad estadounidense. La primera columna corresponde con el tamaño de la casa (en pies cuadrados), la segunda columna indica el número de dormitorios, y la tercera el precio de la casa.

Antes de nada, muestra las diez primeras filas de datos y a continuación:

1. Implementar una función para normalizar los datos de la matriz  $X$  que utilice la desviación estándar.
2. Si la función de coste implementada en la práctica anterior no se puede utilizar para múltiples variables, codificar una nueva. En este caso también hay que añadir una primera columna a la matriz  $X$  con todos sus elementos a 1.
3. Lo mismo ocurre con el método del gradiente, ver si se puede reutilizar o hay que implementar uno nuevo.
4. Mostrar el gráfico de convergencia que se observa en la figura obtenido con `alpha` igual a 0.03 y 200 iteraciones.
5. Predecir el precio de una casa de 1650 pies cuadrados y 3 habitaciones.
6. Calcular los parámetros `theta` mediante el uso de la ecuación normal, y predecir nuevamente el precio de la vivienda con los valores anteriores.





## Problemas

---

**P1.** En el caso de estudio utilizado en los ejercicios anteriores, probar con diferentes valores de `alpha`, como por ejemplo 0.3, 0.1, 0.03, 0.01,... usando la normalización de los datos con desviación estándar (Ejercicio 1). Representar los gráficos de convergencia (Ejercicio 4) y comparar los diferentes resultados obtenidos. Finalmente, utilizar los mejores valores de `theta` para predecir el precio de una casa de 1650 pies cuadrados y 3 habitaciones. Además, se podrían comparar estos resultados con los obtenidos en la ampliación de problemas de la práctica anterior aplicando regresión lineal con una solo variable.

**P2.** Se va a utilizar la librería `scikit learn` (también conocida como `sklearn`) para dividir en dos partes los conjuntos de datos `X` e `y` de `ex1data2.txt`: una parte será el conjunto de entrenamiento y otro de test, contiendo el primero el 70% de los ejemplos elegidos aleatoriamente, mientras que el segundo contendrá el resto. Para ello utilizar la función `train_test_split()`. Al finalizar deberías tener `Xtrain`, `Xtest`, `ytrain` e `ytest`.

A continuación, averiguar los valores de `theta` con los nuevos conjuntos de entrenamiento (partiendo de la inicialización de `theta` a 0 y el `alpha` que se haya considerado mejor en el problema anterior) y comparar los errores obtenidos mediante las medidas error cuadrático medio y correlación sobre los conjuntos de test usando tanto las funciones que se han implementado como la “LinearRegression” de `sklearn`. Se pueden usar las medidas de error que se encuentran en la librería `sklearn`:  
`sklearn.metrics.mean_squared_error(y_test, y_pred)` y  
`sklearn.metrics.r2_score(y_test, y_pred)` respectivamente.