Pràctica 5: Regressió Lineal

Pràctica de Simulació Física

Pau Nonell Isach Login: ls30817
Cristian Gonzàlez Ruz Login: c.gonzalez.2015

Nota: el codi de Python entregat està escrit en la versió 3 (Python 3).

Descripció de la pràctica

En aquesta pràctica hem aplicat els nostres coneixements de regressió lineal en 2 casos pràctics. El primer dels casos és un conjunt de dades de pisos de Boston on agafàvem el número d'habitacions com a atribut i els preus d'aquests pisos com a valor a deduir. És a dir que hem utilitzat el número d'habitacions com a x i el preu com a y. En el segon cas, hem adaptat un projecte de Processing que simula un pes penjat sobre una molla, i n'hem extret les longituds amb diverses masses. La massa del pes és la x i la longitud la y. Per fer-ho, hem utilitzat programació en Python en l'entorn Júpiter.

Classes implementades en Processing

- **SpringOsc:**
  Aquesta classe funciona com a interfície del programa Processing que serveix per extreure les dades de la molla. S'encarrega d'actualitzar i pintar la molla. En el principi de l'execució, calculem una massa aleatòria. Quan el pes de la molla arriba a uns valors mínims d'acceleració i velocitat, suposem que està en repòs i guardem els valors de massa utilitzat i la longitud a la que està respecte l'anclatge. Llavors tornem a calcular una massa aleatòria i tornem a fer el procediment. Les dades les guardem en un fitxer .csv i utilitzem la classe Table i TableRow que ens permet tractar i guardar aquestes dades amb més facilitat.

- **Mover:**
  És el pes o bob de la molla. Conté els atributs de posició, velocitat, acceleració i massa entre altres. A més, té una variable anomenada accAux que permet guardar-nos l'acceleració, ja que el programa posa l'acceleració a 0 a cada frame. D'aquesta manera la classe SpringOsc pot recuperar aquesta informació.

- **Spring:**
  És la classe que representa la molla. Conté els atributs i funcions necessàries per representar i actualitzar la molla del programa.

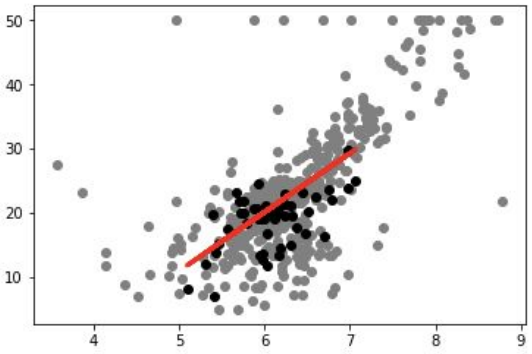
Programa implementat en Python

El programa que hem implementat consisteix en una funció que és utilitzada dues vegades, una per a la primera part de la pràctica (els pisos de Boston) i una altra per a extreure la distància que s'allarga la molla depenent del pes que apliquem amb l'objectiu de predir la constant elàstica de la molla. Per al primer cas, el que cal passar-li a la funció és el dataset d'entrenament obtingut a partir del dataset de Boston (tots els elements excepte els 50 últims) i el dataset de prova obtingut a partir dels últims 50 elements. Per al segon cas, el dataset és obtingut a través de la lectura del fitxer CSV, que ha sigut generat pel programa de Processing. De la mateixa manera, es dediquen tots els elements excepte els 50 últims per al dataset d'entrenament i els 50 últims són per al dataset de prova.

La funció implementada es diu "linearRegression" i s'encarrega de rebre els datasets, calcular la regressió lineal i mostrar algunes gràfiques per tal d'avaluar el resultat obtingut i mesurar la seva fiabilitat a través de l'error. En primer lloc, la funció s'encarrega de normalitzar les dades, tot calculant i mostrant la mitja i la desviació estàndard utilitzades. Després de mostrar una gràfica de les dades d'entrenament ja normalitzades, es procedeix a preparar el càlcul de la regressió lineal, de manera que es concatenen les dades formant matrius que fan possible els càlculs posteriors. A continuació, es procedeix a iniciar l'algoritme del gradient, que busca minimitzar l'error per tal de trobar la millor regressió lineal possible. L'objectiu d'aquest algoritme és que els valors de les theta's es vagin apropant al valor desitjat i que l'error vagi disminuint. Finalment, la funció s'encarrega de mostrar diverses gràfiques, entre les quals destaquem J en funció de les iteracions i dels valors de les theta's.

Resultats

Pel que fa als resultats, podem observar a les gràfiques del programa implementat que s'ha aconseguit reduir l'error en cada iteració, pel que podem confirmar que aprèn. A més, podem observar que els valors de les theta's s'estabilitzen als valors trobats. Per altra banda, podem comparar la regressió lineal obtinguda amb les eines de sklearn (es mostra a continuació). Ambdues regressions no són idèntiques ja que els algoritmes de sklearn incorporen altres optimitzacions però els resultats tenen força similitud, pel que podem confirmar que el nostre programa funciona correctament.

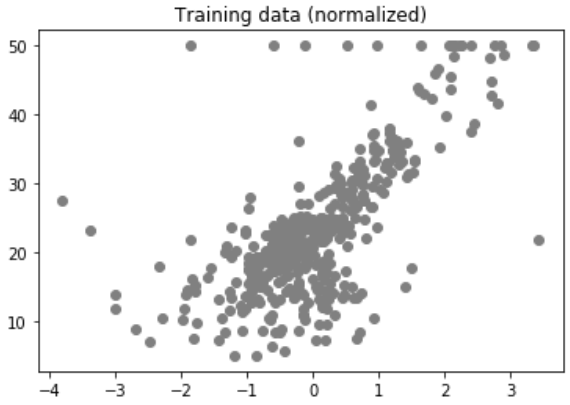


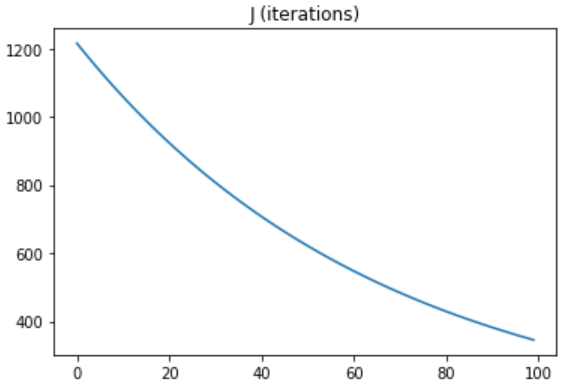
Primera part: regressió lineal obtinguda amb les eines de sklearn

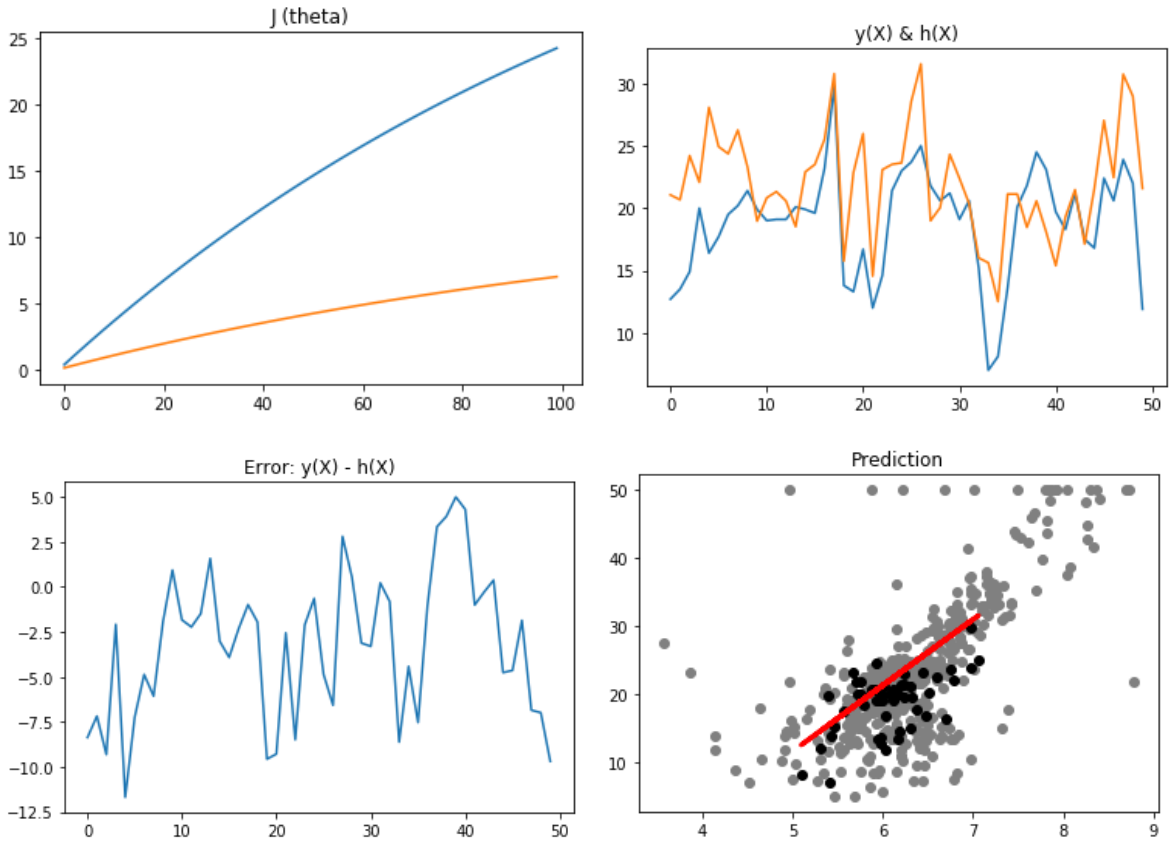
A continuació es mostren les gràfiques del nostre programa.

Primera part: Pisos Boston

Mean: [6.30723465]
Std: [0.72156238]



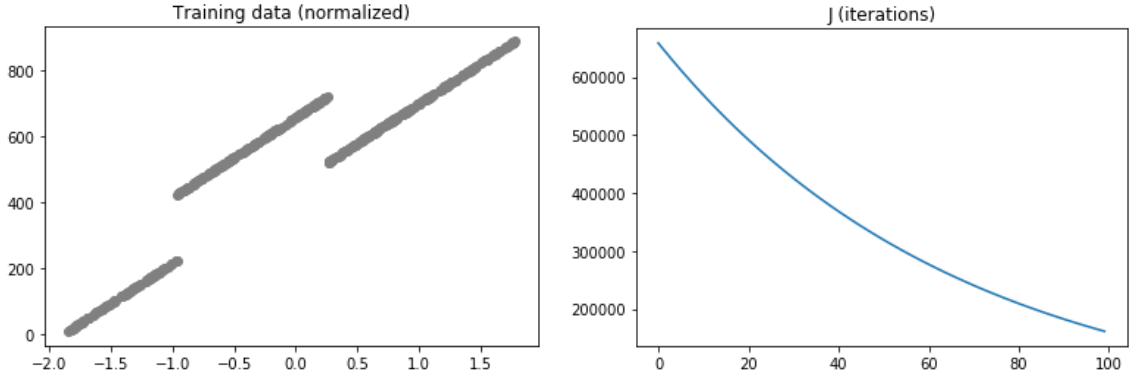


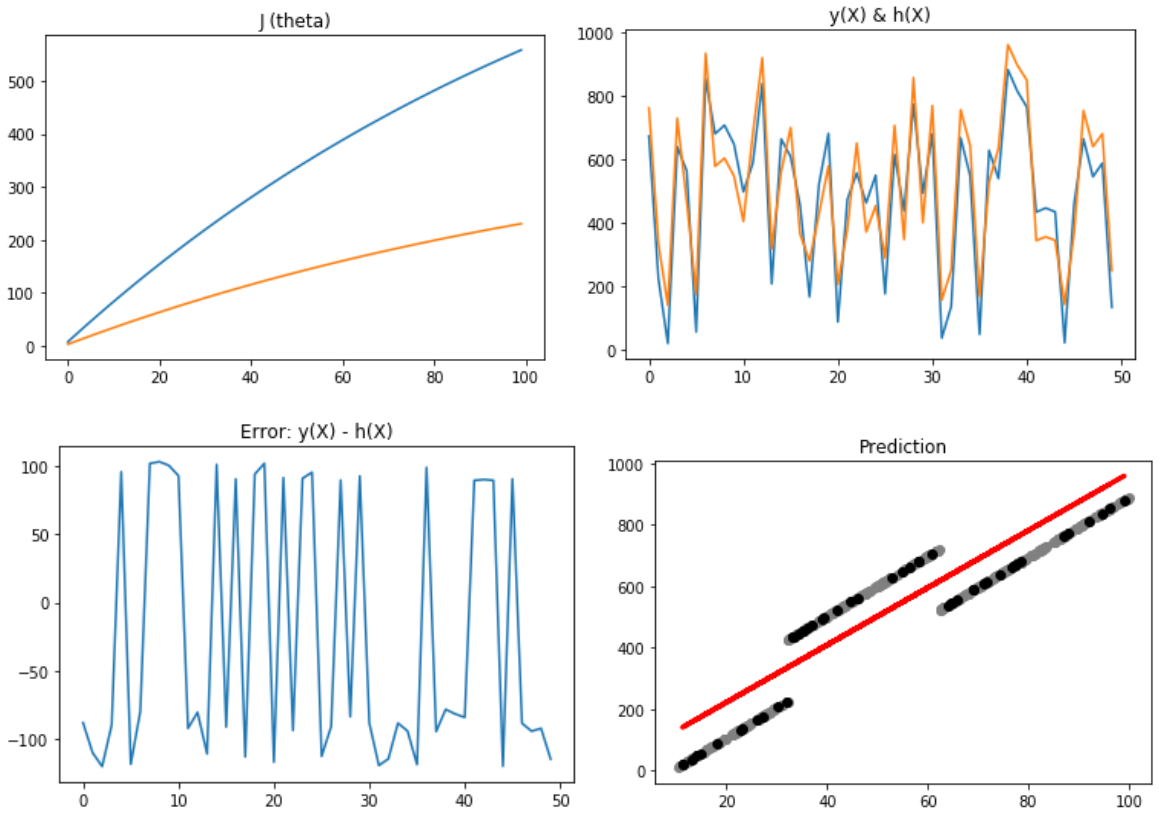


Segona part: Valor de la molla (amb k = 0.2, gravetat = 2 i masses aleatòries entre 20 i 100)

De la mateixa manera que en la primera part de la pràctica, podem observar a les gràfiques del programa implementat s'ha aconseguit reduir l'error en cada iteració, pel que podem confirmar que aprèn. A més, podem observar que els valors de les theta's s'estabilitzen als valors trobats.

Mean: 56.02246451222222
Std: 24.745422685823023

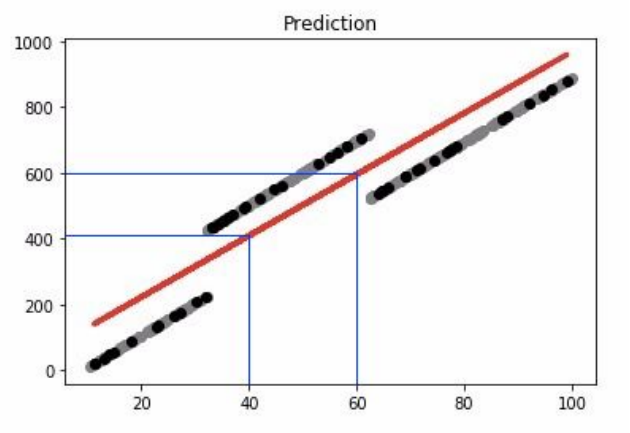




Càlcul de la k:

Per a trobar la constant elàstica de la molla, utilitzem la recta de regressió i la llei de Hooke, és a dir, $F = -k * x$. Per fer-ho agafem dos punts de la recta i en trobem la diferència. Com que el valor de les x és la massa en kg, hem de multiplicar-la per la gravetat (que és 2, tal com es veu en la línia 32 de "SpringOsc"). El resultat és el següent:

$$(60 - 40) * 2 = k * (600 - 410)$$
$$k = \frac{20 * 2}{190} = 0.2105$$



Observem que ens dona k = 0.2105, resultat que s'aproxima molt a la k designada al Processing. Per tant, el resultat és correcte.