

Práctica 5:

Regresión lineal regularizada: sesgo y varianza

Parte 1: Regresión lineal regularizada

```
datafile = 'ex5data1.mat'  
mat = loadmat(datafile)
```

```
X = mat.get("X")  
y = mat.get("y")
```

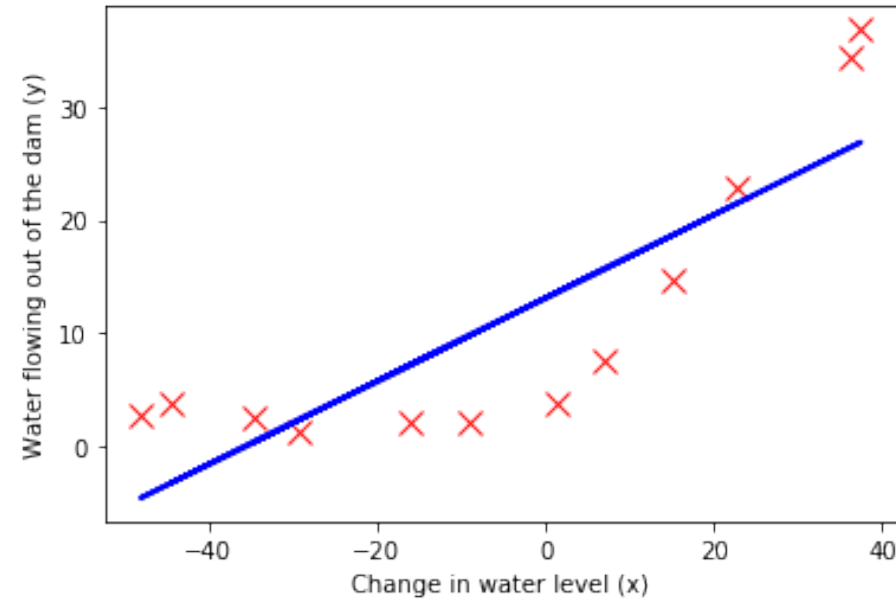
```
Xval = mat.get("Xval")  
yval = mat.get("yval")
```

```
print(X.shape, y.shape)  
print(Xval.shape, yval.shape)
```

```
print(np.mean(X), np.std(X))  
print(np.mean(y), np.std(y))
```

```
(12, 1) (12, 1)  
(21, 1) (21, 1)  
-5.085426348834809 28.68873075847896  
11.217589325366376 12.492955274415026
```

Parte 1: Regresión lineal regularizada



Para un valor de $\lambda = 1$ y $\Theta = [1; 1]$
coste = 303,993
gradiente = $[-15,303; 598,167]$.

Parte 2: Curvas de aprendizaje

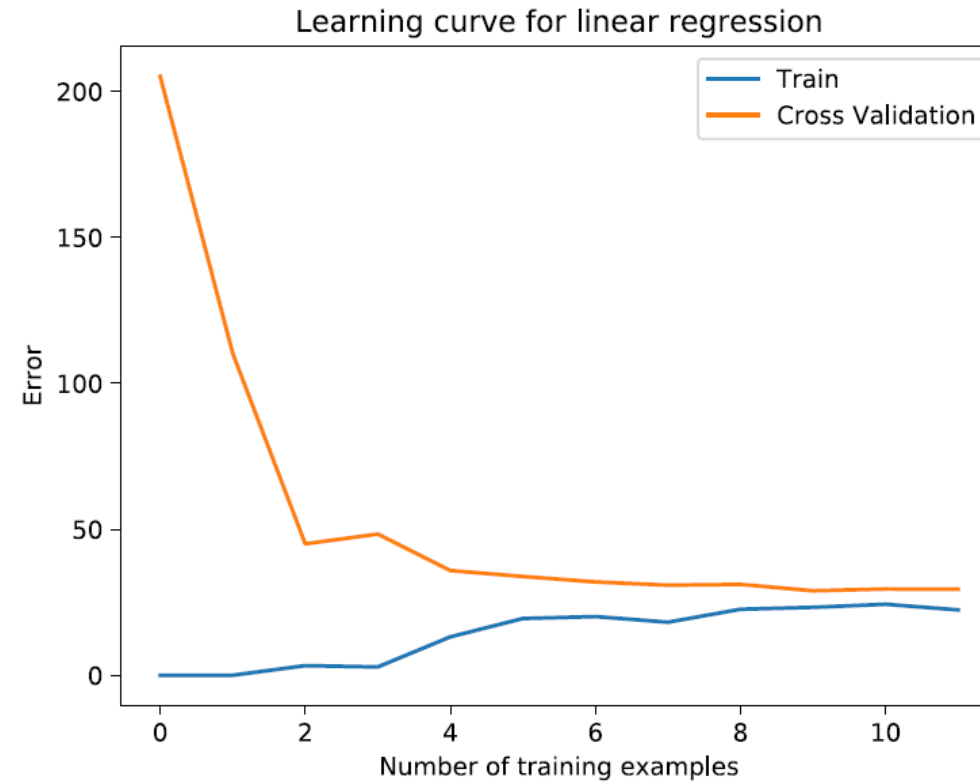
Para generar las curvas de aprendizaje, has de repetir el entrenamiento por regresión lineal del apartado anterior, utilizando diferentes subconjuntos de los datos de entrenamiento.

Primero, 1, después 2, y así sucesivamente hasta llegar a m (en python, puedes generar el subconjunto de tamaño i como $X[0:i]$, $y[0:i]$).

Una vez realizado el entrenamiento por regresión lineal para ajustarse al subconjunto de entrenamiento $X[0:i]$, $y[0:i]$, has de evaluar el error el resultado aplicado sobre ese mismo subconjunto, así como el error al clasificar a todos los ejemplos del conjunto de validación (X_{val} e y_{val}).

$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \right]$$

Parte 2: Curvas de aprendizaje



Parte 3: Regresión polinomial

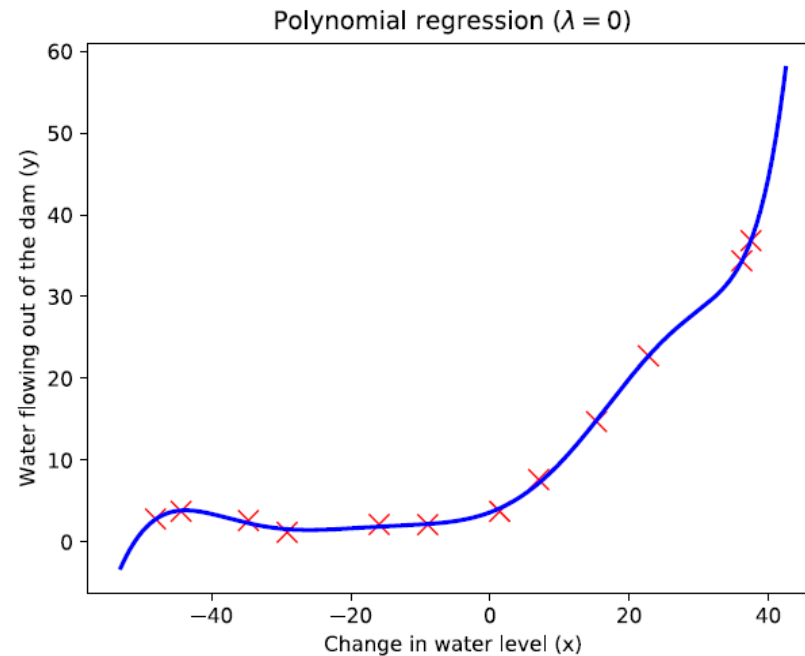
$$\begin{aligned}h_{\theta}(x) &= \theta_0 + \theta_1 * (\text{nivelAgua}) + \theta_2 * (\text{nivelAgua})^2 + \dots + \theta_p * (\text{nivelAgua})^p \\ &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_p x_p\end{aligned}$$

Para ello, en primer lugar, has de implementar una función que sea capaz de generar los nuevos datos de entrenamiento a partir de los datos originales X . Esta función recibirá una matriz X de dimensión $m \times 1$ y un número p , y devolverá otra matriz de dimensión $m \times p$ que en la primera columna contenga los valores de X , en la segunda el resultado de calcular $X.^2$, en la tercera $X.^3$, y así sucesivamente.

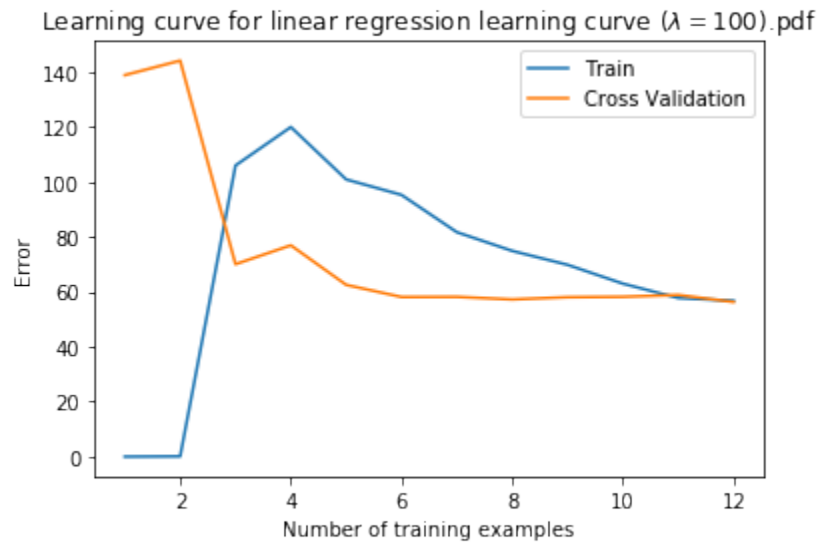
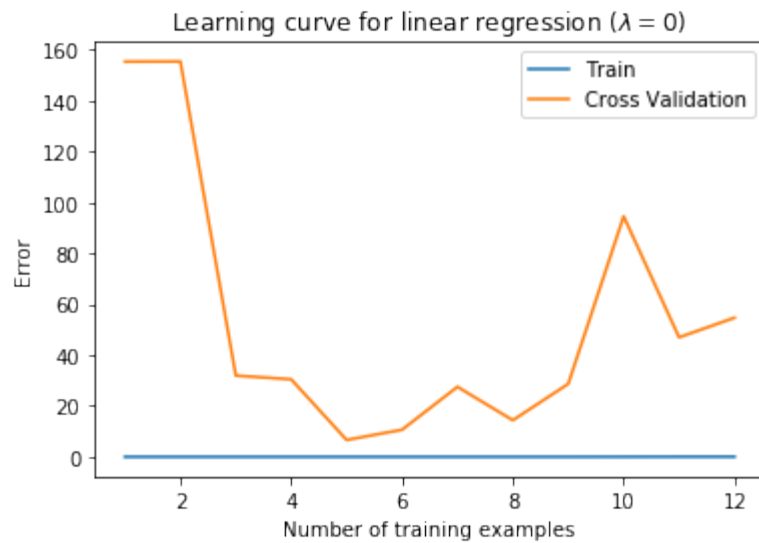
Después hay que normalizar los atributos antes de realizar el aprendizaje.

Parte 3: Regresión polinomial

Genera de esta forma los nuevos datos de entrada para aprender un polinomio de grado $p = 8$ (a los que has de añadir una columna de 1's) y vuelve a aplicar el método de regresión lineal para obtener el vector Theta que minimiza el error.



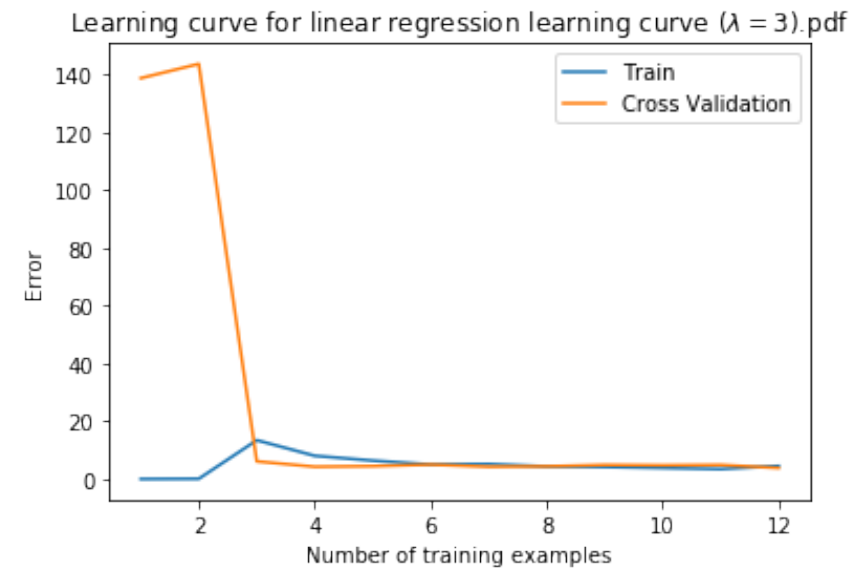
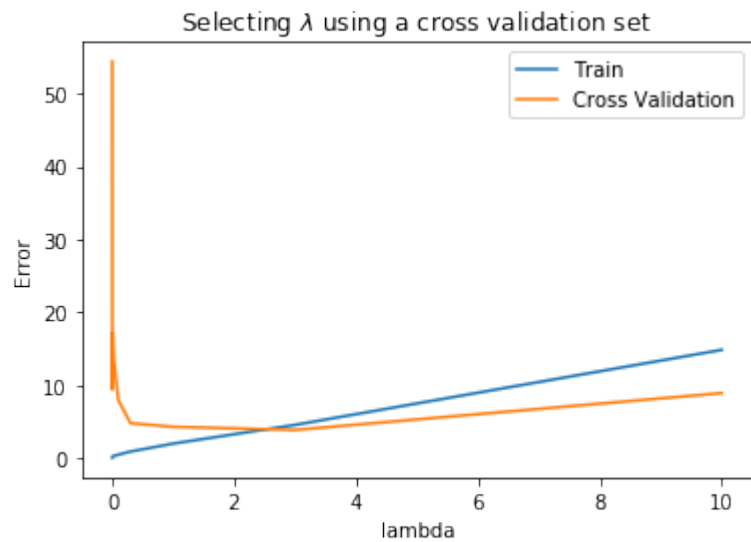
Parte 3: Curvas de aprendizaje



Parte 4: Selección del parámetro lambda

La técnica que se suele utilizar para elegir el valor de lambda es evaluar la hipótesis generada sobre los ejemplos de entrenamiento con un segundo conjunto de ejemplos de validación y seleccionar aquél valor de lambda que minimice el error.

Aplica esta técnica con los valores $\{ 0, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10 \}$



Parte 4: Evaluación final

Por último, deberías estimar el error de la hipótesis aplicándola a un tercer conjunto de ejemplos que no hayas utilizado para entrenar ni tampoco para seleccionar λ .

Calculando el error sobre los datos de prueba X_{test} (que también debes pasar a forma polinomial con potencia 8 y después normalizar con las medias y normales de los datos de entrenamiento) e y_{test} , para $\lambda = 3$ deberías obtener un error entorno a 3,572.