



# Tecnológico de Monterrey

ITESM CSF

Aprendizaje automático 7-10 lu

Dr. Víctor de la Cueva

Septiembre 4 2017

Adrián Biller A01018940

**Documentación Proyecto 3**

## Introducción

Con el conocimiento obtenido con respecto a la regresión logística se realizó el proyecto 3. Se utilizaron las fórmulas especificadas en clase para el costo:

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

Y la fórmula de gradiente de costo:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

## Manual de usuario

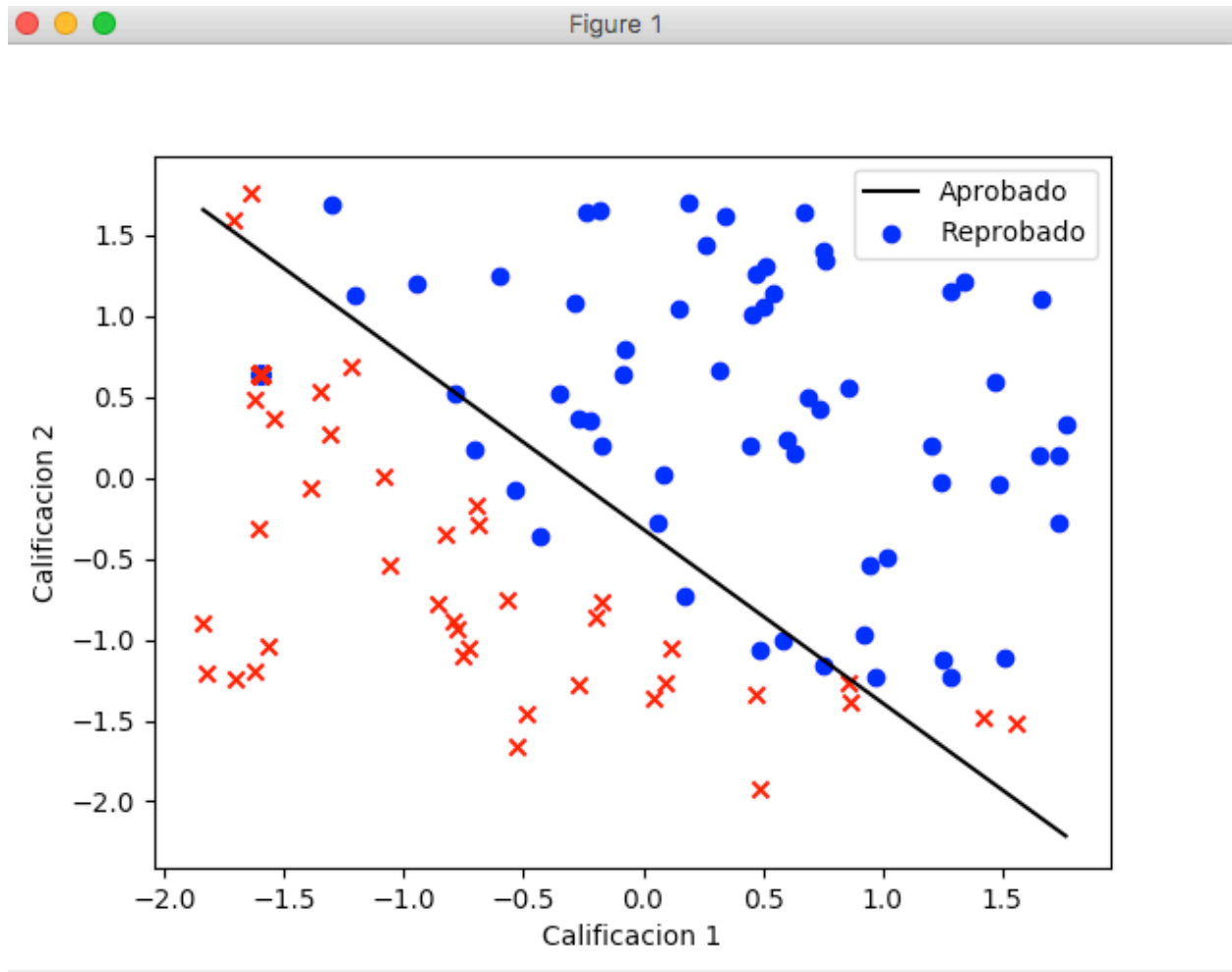
Para poder utilizar este programa se debe utilizar la terminal. Antes la computadora en donde se realice la prueba debe contener Python 3 (el programa fue realizado utilizando Python 3.6.1) además se utilizan las librerías *numpy* y *matplotlib.pyplot*, *math* y *pylab* en el programa, en caso de errores con las librerías se deberán instalar por separado dependiendo de la máquina que se esté utilizando puede variar el método de instalación.

Primero se deberá colocar en la carpeta en la que se localice el programa .py y el archivo .txt conteniendo los datos a analizar. Una vez dentro de la carpeta se deberá ingresar el siguiente comando en terminal:

```
Proyecto3 Biller$ python3 Proyecto3.py
```

Una vez ingresado el comando se realizará el programa y regresará el resultado y se mostrarán en pantalla, en caso de querer repetir el proceso se deberá realizar el mismo comando en terminal.

Al ingresar el comando se mostrará en pantalla una gráfica describiendo las calificaciones obtenidas por los alumnos y situándolos si son reprobados o aprobados:



## Funciones

- `graficaDatos(x, y, theta)`
  - Entrada:
    - $X$  = Valores de  $x$  sin columna de 1's
    - $Y$  = Valores de  $y$  obtenidos de los datos
    - $\Theta$  = vector  $\theta$
  - Salida:
    - Esta función no regresa ningún valor
- `aprende(theta, x, y, iteraciones)`
  - Entrada:
    - $\Theta$ : vector  $\theta$  inicializado en 0's

- $X$  = vector  $x$  con columnas de 1's concatenadas al inicio
  - $Y$  = vector  $y$  de los datos obtenidos
  - Iteraciones = número de iteraciones para realizar el algoritmo de gradiente descendente
- Salida:
  - $\Theta$  = valor de  $\theta$  después de pasar por el algoritmo
- `predice(theta, x)`
  - Entrada:
    - $\Theta$  = recibe  $\theta$  después de entrenamiento
    - $X$  = vector de  $x$  con columna de 1's concatenada
  - Salida:
    - Vector  $p$  con 1's y 0's describiendo la predicción de los alumnos
- `sigmoidal(z)`
  - Entrada:
    - $Z$  = valor  $z$  definido por  $x \Theta^T$
  - Salida:
    - Valor de función sigmoidal
- `funcionCosto(theta, x, y)`
  - Entrada:
    - $\Theta$  = vector  $\theta$
    - $X$  = vector  $x$  con columna de 1's concatenada
    - $Y$  = vector de  $y$
  - Salida:
    - $J$  = costo de la  $\theta$  ingresada
    - $\text{Grad}$  = gradiente de costo de  $\theta$
- `normalizar_datos(old_x)`
  - Entrada:
    - $\text{Old\_x}$  = vector de  $x$  antes de normalización, no contiene columna de 1's
  - Salida:
    - Nuevo vector  $x$  normalizado