02/04/2019

Carlos Alberto Cordero Robles

Regresión logística

Inicialmente se tiene una base de datos con dos tipos de hongos (p y e) que son los mencionados en la columna “class”.

Usando la librería de pandas lee el CSV.

Para la matriz X quita la columna “class” y hace uso de funciones de pandas para generar una matriz con solo 0 y 1, en general 1 significa que si tiene cierta característica y 0 que no y de ser una matriz de 8125x20 pasa a ser una de 8125x118 al final le agrega un bias de 1(una columna con puros 1s).

Para el vector y tomara la columna “class” y al tipo p=1 y e=0.

Usando la función train\_test\_split de sklearn separa de manera random 85% de las entradas para entrenamiento y 25% para pruebas.

Así como en la regresión clásica se tiene un vector con las constantes por las que se multiplicará cada columna de X y deberá dar el vector y, en este código este vector de constantes se iniciará con valores aleatorios entre -1 y 1 y se llaman theta y el objetivo será entrenar hasta optimizar este vector theta.

X\*theta = y

El objetivo será entrenar theta, para esto se tiene un “learning rate”, y una regularización.

Se iterará N veces aplicando la siguiente formula:

theta = theta -  Δtheta

Donde

Δtheta = ((learning rate/ Num.Entradas) \* (X\*error)) + (regularización \* theta )

y error es

error = sigmoid(X\*theta) – y

Tanto error como regularización tenderán a 0 cada iteración y llegara un momento en que theta se estabilizara y se tomara como los pesos ya entrenados.

También se calcula un valor de costo que es:

Costo = trans(-y)\*ln(sigmoid(X\*theta)) + ((-trans(1-y)) \* ln(1- sigmoid(X\*theta))) / Num.Entradas

Así como regularización y error, el costo se va reduciendo hasta llegar a un valor muy cercano a 0.

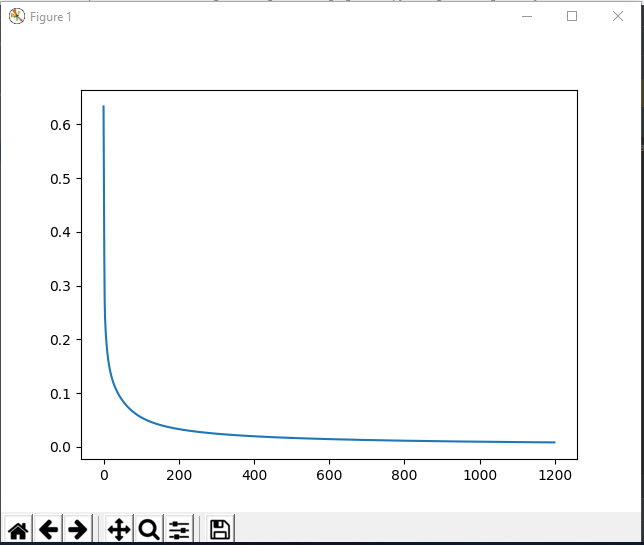


Imagen1. Costo

Con esto se podría ver que prácticamente a partir de la iteración 400 ya no se tiene mucho cambio.

Por ultimo se aplican las pruebas multiplicando X\_pruebas\*theta y el valor debe ser igual a y\_pruebas y al correr el código se podrán dar cuenta que el porcentaje de aciertos es 100% o un valor muy cercano.

Leyendo el documento nos dice que sigmoid(X\*theta) es la probabilidad de que una x este entre 0 y 1.

El costo será mayor mientras más alejado este sigmoid(X\*theta) de y, de esta manera mientras se va entrenando theta, el costo va bajando y se tienen predicciones más acertadas.