
# EXAMEN PARCIAL 1: MODELOS DE REGRESIÓN

Alberto Javier Pelayo Brambila – A01636406

18 DE SEPTIEMBRE DE 2022

PROFESOR: EDGAR MACIAS GARCÍA

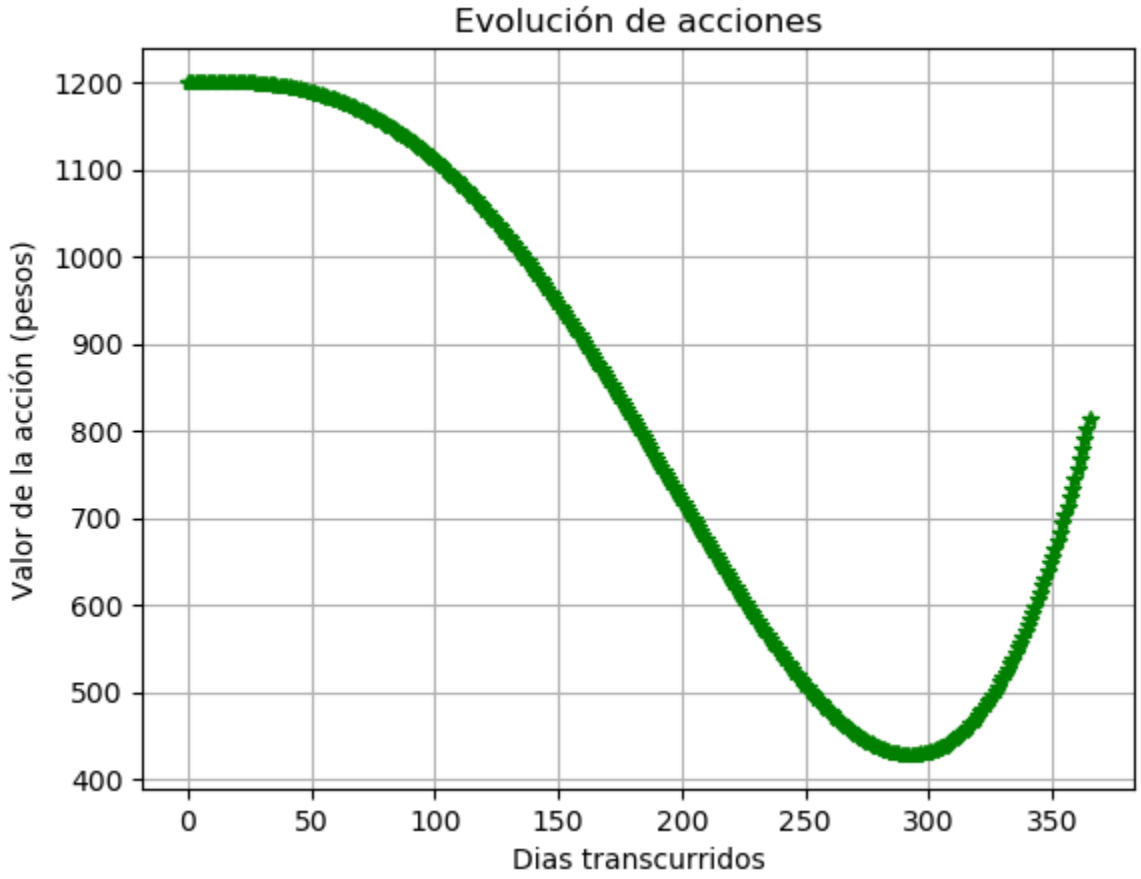
Proyecto de Control Avanzado 1

# Problema 1: Inversiones a largo plazo

## Análisis de datos

Carlos Delgado es un empresario que desea invertir parte de sus utilidades en acciones de una empresa de telecomunicaciones, a fin de venderlas al transcurso de un año. Para tomar la mejor decisión generó una base de datos que contiene el valor de las acciones a lo largo de un año.

Usando los datos obtenidos, el empresario pretende crear un modelo que le permita extrapolar el valor de las acciones a lo largo del año siguiente.




Viendo el modelo lo que primero hacemos es definir nuestras entradas y salidas de nuestro sistema

- **Entradas:** X (días transcurridos)
- **Salidas:** Y (valor de la acción en pesos)

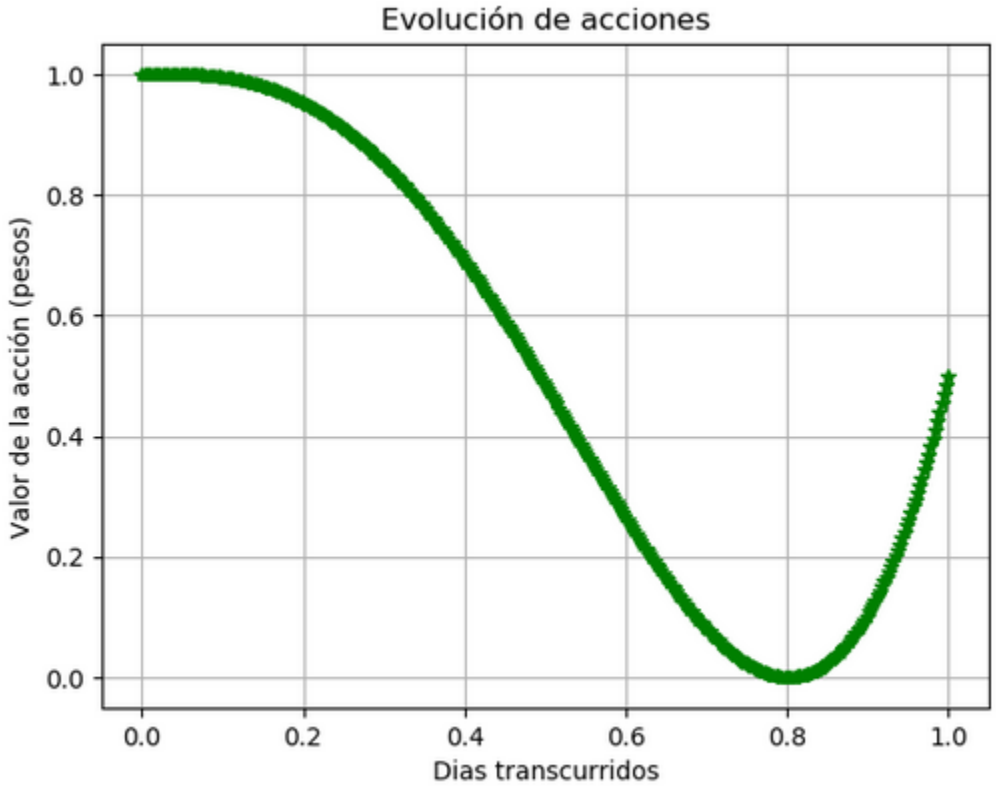
Al observar el modelo podemos ver que este modelo ambas matrices son de tamaño (365,). También podemos observar a su vez que los valores que son manejados son números reales lo que nos lleva a determinar un modelo de Regresión.

El siguiente paso es normalizar nuestro modelo con el fin de tener valores limitados entre 0 y 1 en ambos ejes, para esto se aplican las siguientes fórmulas de normalización:

$$x = \frac{x - x_{min}}{x_{max}}$$

$$y = \frac{y - y_{min}}{y_{max}}$$

Al aplicar estas fórmulas obtenemos:




## Planteamiento del modelo

Empezamos en esta sección proponiendo nuestro modelo tomando de referencia los datos de entrada (X) y salida (Y). Para lograr el comportamiento de la gráfica se optó por usar una función polinomial de cuarto orden.

$$h = ax^4 + bx^3 + cx^2 + dx + f$$

Posteriormente definimos nuestra función de error la cuál será el error medio cuadrático cuya fórmula es:

$$e = \frac{(y_m - y_d)^2}{2}$$

## Entrenamiento del modelo

Para poder optimizar nuestro problema a resolver se pretende minimizar a lo más posible nuestro error (e) mediante ajustar los parámetros de nuestro modelo parametrizado, es decir, de nuestra función polinomial de cuarto orden (a, b, c, d, f). Para lograr lo anterior usamos el algoritmo del gradiente descendiente.

$$W_{k+1} = W_k - L_r \frac{\partial e}{\partial W}$$

Por la naturaleza de nuestros parámetros se debe de aplicar la regla de la cadena para obtener los gradientes correspondientes

Para llegar a nuestra solución primero debemos de proponer los gradientes

$$\frac{\partial e}{\partial a} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial a}$$

$$\frac{\partial e}{\partial b} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial b}$$

$$\frac{\partial e}{\partial c} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial c}$$

$$\frac{\partial e}{\partial d} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial d}$$

$$\frac{\partial e}{\partial f} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial f}$$

Después encontramos los gradientes por separado

$$\frac{\partial e}{\partial y_m} = \frac{\partial}{\partial ym}\left[\frac{(y_m - y_d)^2}{2}\right] = y_m - y_d$$

$$\frac{\partial y_m}{\partial a} = \frac{\partial}{\partial a}[ax^4 + bx^3 + cx^2 + dx + f] = x^4$$

$$\frac{\partial y_m}{\partial b} = \frac{\partial}{\partial b}[ax^4 + bx^3 + cx^2 + dx + f] = x^3$$

$$\frac{\partial y_m}{\partial c} = \frac{\partial}{\partial c}[ax^4 + bx^3 + cx^2 + dx + f] = x^2$$

$$\frac{\partial y_m}{\partial d} = \frac{\partial}{\partial d}[ax^4 + bx^3 + cx^2 + dx + f] = x$$

$$\frac{\partial y_m}{\partial f} = \frac{\partial}{\partial f}[ax^4 + bx^3 + cx^2 + dx + f] = 1$$

Por último, sustituyendo y aplicando la regla de la cadena podemos obtener nuestros gradientes

$$\frac{\partial e}{\partial a} = (y_m - y_d) \cdot x^4$$

$$\frac{\partial e}{\partial b} = (y_m - y_d) \cdot x^3$$

$$\frac{\partial e}{\partial c} = (y_m - y_d) \cdot x^2$$

$$\frac{\partial e}{\partial d} = (y_m - y_d) \cdot x$$

$$\frac{\partial e}{\partial f} = (y_m - y_d)$$

## Resultados experimentales

Se creó una instancia para nuestro modelo con unos parámetros iniciales:

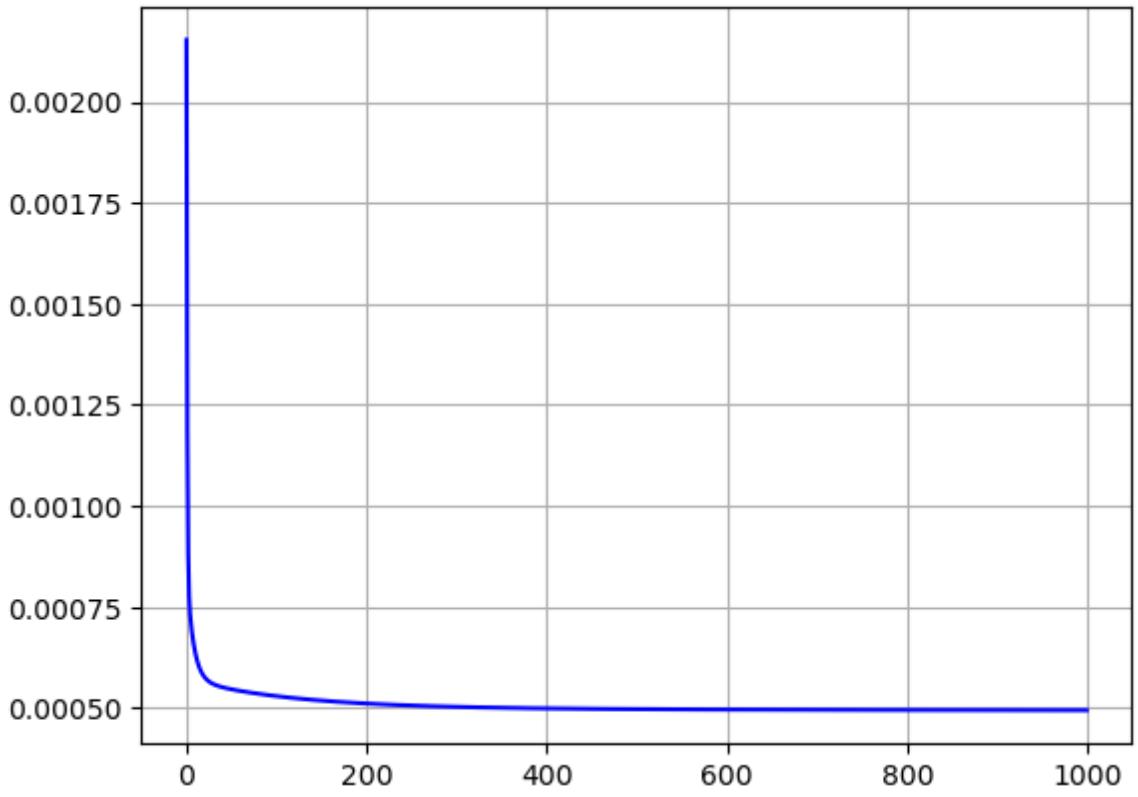
$$a = 0.9, \quad b = 5.5, \quad c = -9.0, \quad d = 2.0, \quad f = 0.9$$

```
#crear instancia del modelo
inversiones=cuarta(0.9,5.5,-9.0,2.0,0.9)
```

A este modelo se le sometió a un entrenamiento con una taza de aprendizaje $L_r = 0.001$ y durante 1000 épocas.

```
#entrenar modelo
inversiones.train(x,y,0.001,1000)
```

Con dicho entrenamiento obtenemos cómo se comporta el error (error medio cuadrático):



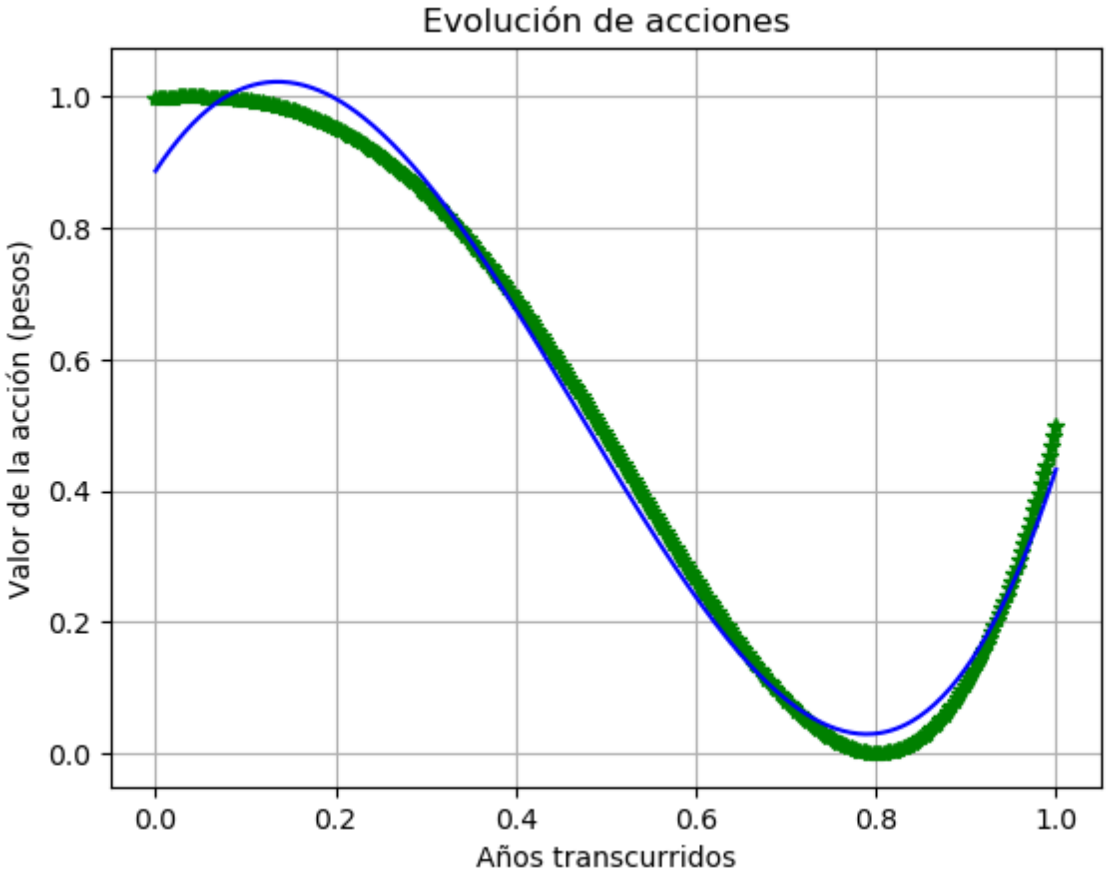

Con nuestro modelo parametrizado obtenemos un error igual a:

$$Error\ (e) = 0.000494566201577238$$

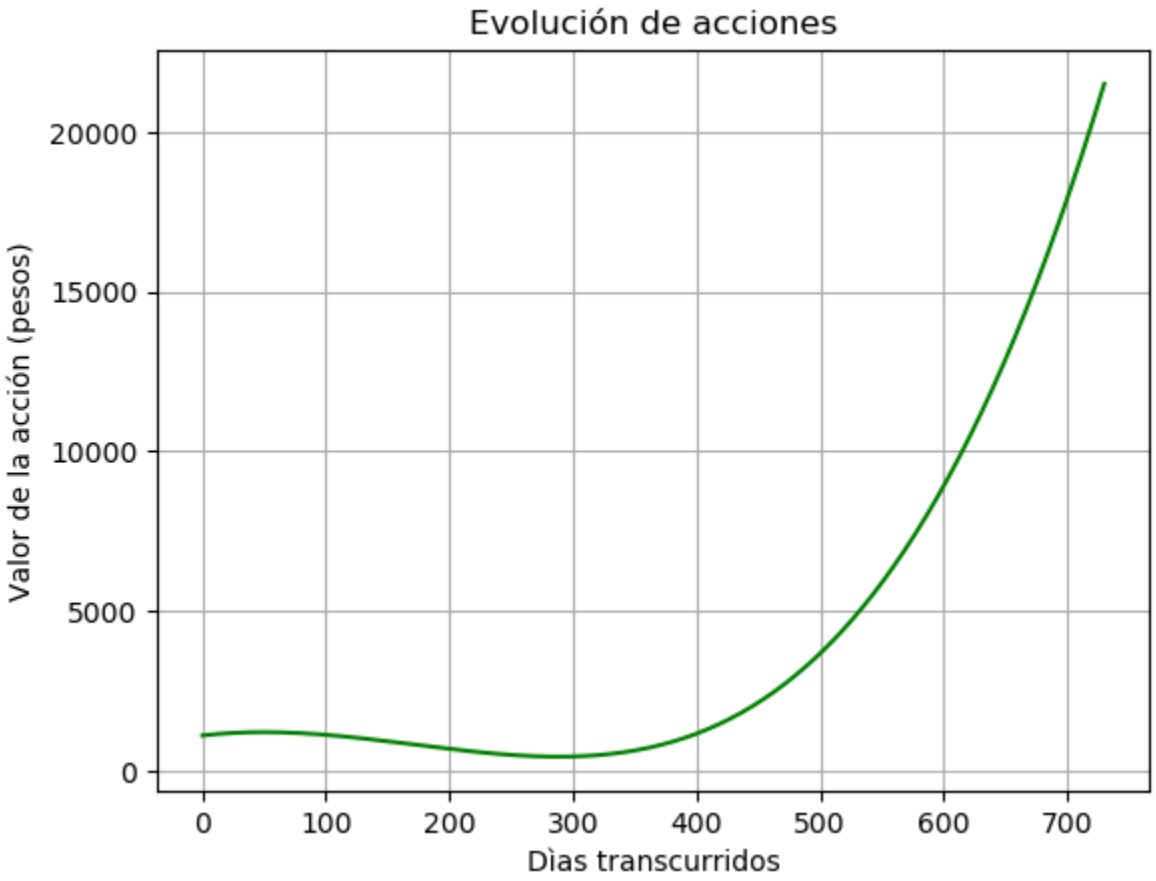
Con unos parámetros de modelos finales iguales a:

$$a = 0.8716, \qquad b = 5.4905, \qquad c = -8.9265, \qquad d = 2.1093, \qquad f = 0.88698$$

Posteriormente podemos visualizar el comportamiento de nuestro modelo parametrizado (color azul) comparado con nuestra gráfica inicial obtenida de los datos (color verde)




Para predecir los valores y el comportamiento de nuestro modelo procedemos a graficar su comportamiento con los valores de los parámetros (a, b, c, d, f) finales. Se debe también de desnormalizar la gráfica invirtiendo las fórmulas de normalización antes vistas. Obteniendo de esta manera nuestra gráfica con el comportamiento durante dos años (el primero es el ya visto y el segundo es el que se predice)




## Conclusiones

Analizando los resultados arrojados, nuestro modelo parametrizado en un principio no es tan exacto, aunque conforme va avanzando podemos ver que es prácticamente muy similar a la gráfica de datos y esto incluso se ve reflejado con nuestro error ya que es bastante pequeño.

Al visualizar la gráfica con la predicción de datos podemos observar que las acciones según nuestro modelo y su comportamiento durante un año tendrán a dispararse y crecer exponencialmente.

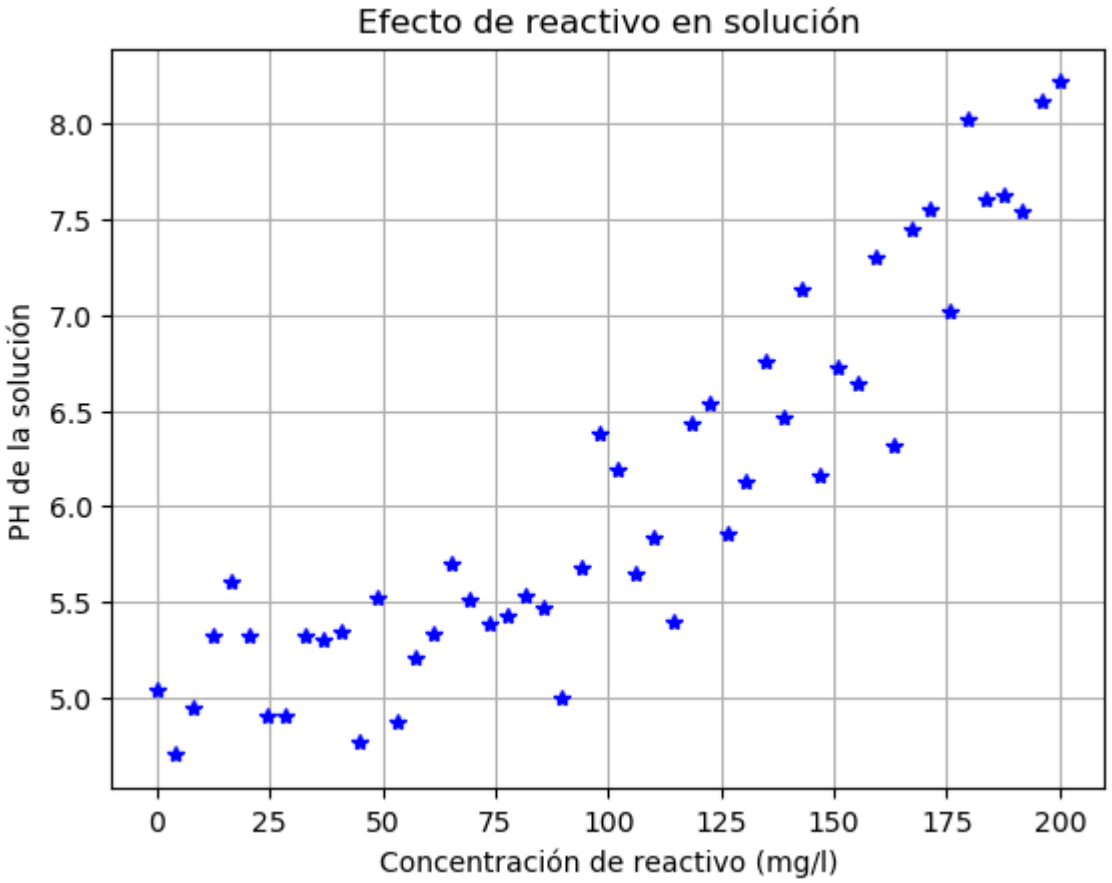
El grado que logró satisfacer nuestras necesidades en este problema fue una función de cuarto grado, se podría intentar con una función de quinto orden para ver si el error llegara a bajar aunque sería muy mínimo y solo aumentaría complejidad y más tiempo de ejecución de nuestro modelo volviéndolo poco eficiente.

# Problema 2: Concentración de reactivo

## Análisis de datos

Saúl Álvarez, un ingeniero egresado del Tecnológico de Monterrey desarrolló un reactivo orgánico que permite disminuir el grado de acidez de los alimentos, mismo que desea patentar para comenzar su producción a gran escala en México.

A fin de sustentar su invención con resultados cuantitativos, elaboró una serie de experimentos donde añadió diferentes grados de concentración del reactivo a soluciones ácidas y alcalinas, a fin de modelar el comportamiento de su reactivo.




Viendo el modelo lo que primero hacemos es definir nuestras entradas y salidas de nuestro sistema

- **Entradas:** X (Concentración de reactivo en mg/l)
- **Salidas:** Y (PH de la solución)

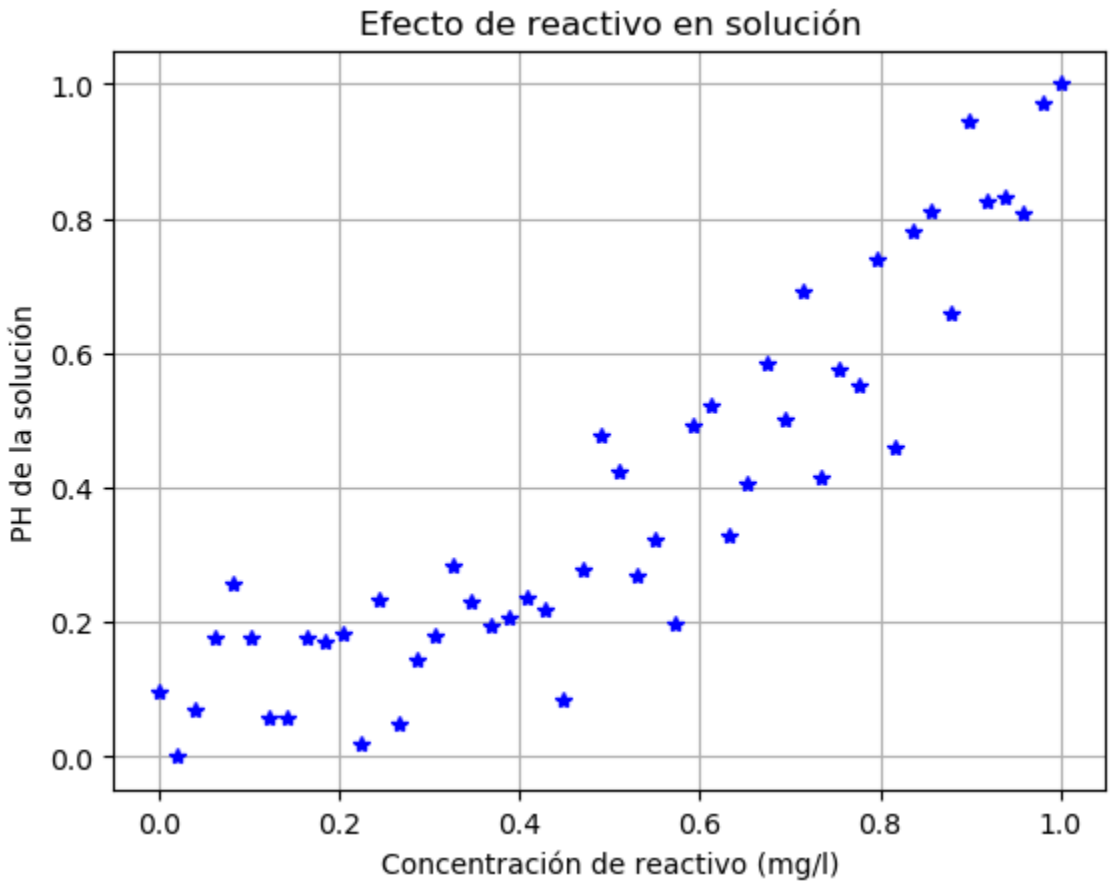
Al observar el modelo podemos ver que este modelo ambas matrices son de tamaño (50,). También podemos observar a su vez que los valores que son manejados son números reales lo que nos lleva a determinar un modelo de Regresión.

El siguiente paso es normalizar nuestro modelo con el fin de tener valores limitados entre 0 y 1 en ambos ejes, para esto se aplican las siguientes fórmulas de normalización:

$$x = \frac{x - x_{min}}{x_{max}}$$

$$y = \frac{y - y_{min}}{y_{max}}$$

Al aplicar estas fórmulas obtenemos:




## Planteamiento del modelo

Empezamos en esta sección proponiendo nuestro modelo tomando de referencia los datos de entrada (X) y salida (Y). Para lograr el comportamiento de la gráfica se optó por usar una función cuadrática.

$$h = ax^2 + bx + c$$

Posteriormente definimos nuestra función de error la cuál será el error medio cuadrático cuya fórmula es:

$$e = \frac{(y_m - y_d)^2}{2}$$

## Entrenamiento del modelo

Para poder optimizar nuestro problema a resolver se pretende minimizar a lo más posible nuestro error (e) mediante ajustar los parámetros de nuestro modelo parametrizado, es decir, de nuestra función cuadrática (a, b, c). Para lograr lo anterior usamos el algoritmo del gradiente descendiente.

$$W_{k+1} = W_k - L_r \frac{\partial e}{\partial W}$$

Por la naturaleza de nuestros parámetros se debe de aplicar la regla de la cadena para obtener los gradientes correspondientes

Para llegar a nuestra solución primero debemos de proponer los gradientes

$$\frac{\partial e}{\partial a} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial a}$$

$$\frac{\partial e}{\partial b} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial b}$$

$$\frac{\partial e}{\partial c} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial c}$$

Después encontramos los gradientes por separado

$$\frac{\partial e}{\partial y_m} = \frac{\partial}{\partial ym}\left[\frac{(y_m - y_d)^2}{2}\right] = y_m - y_d$$

$$\frac{\partial y_m}{\partial a} = \frac{\partial}{\partial a}[ax^2 + bx + c] = x^2$$

$$\frac{\partial y_m}{\partial b} = \frac{\partial}{\partial b}[ax^2 + bx + c] = x$$

$$\frac{\partial y_m}{\partial c} = \frac{\partial}{\partial c}[ax^2 + bx + c] = 1$$

Por último, sustituyendo y aplicando la regla de la cadena podemos obtener nuestros gradientes

$$\frac{\partial e}{\partial a} = (y_m - y_d) \cdot x^2$$

$$\frac{\partial e}{\partial b} = (y_m - y_d) \cdot x$$

$$\frac{\partial e}{\partial c} = (y_m - y_d)$$

## Resultados experimentales

Se creó una instancia para nuestro modelo con unos parámetros iniciales:

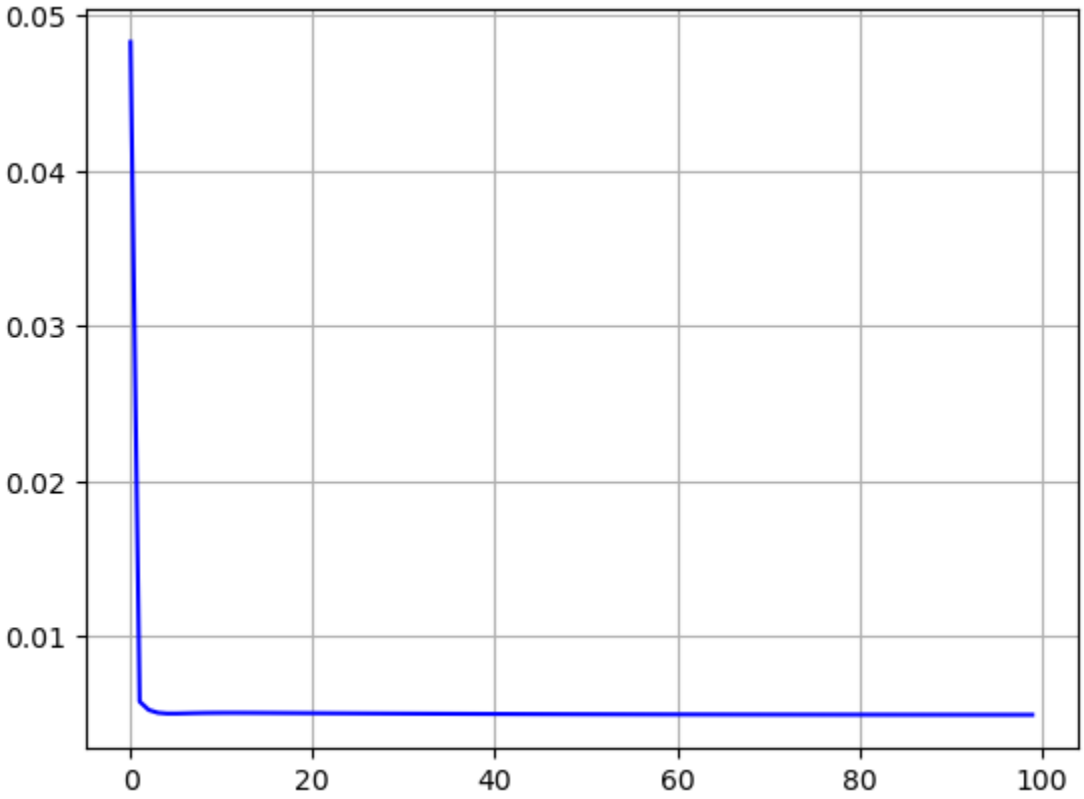
$$a = 0.6, \quad b = 0.2, \quad c = 1.0$$

```
#crear instancia del modelo
reactivo=cuadratica(0.6,0.2,1.0)
```

A este modelo se le sometió a un entrenamiento con una taza de aprendizaje $L_r = 0.1$ y durante 100 épocas.

```
#entrenar modelo
reactivo.train(x,y,0.1,100)
```

Con dicho entrenamiento obtenemos cómo se comporta el error (error medio cuadrático):



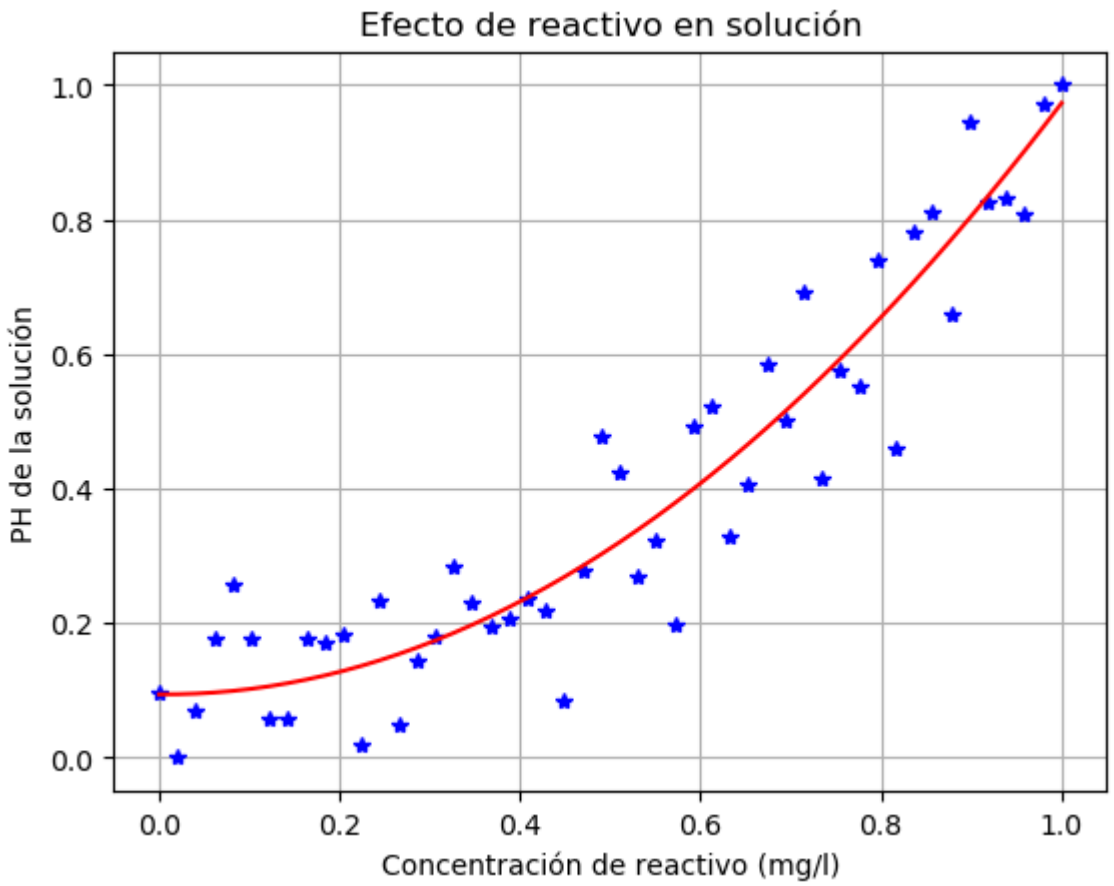

Con nuestro modelo parametrizado obtenemos un error igual a:

$$Error\ (e) = 0.0042676448632733436$$

Con unos parámetros de modelos finales iguales a:

$$a = 0.8900, \qquad b = -0.01098, \qquad c = 0.09418$$

Posteriormente podemos visualizar el comportamiento de nuestro modelo parametrizado (color rojo) comparado con nuestra gráfica inicial obtenida de los datos (color azul)




-

## Conclusiones

Analizando nuestros resultados después del entrenamiento podemos notar que los parámetros iniciales no estaban tan lejos de los parámetros finales, aunque siempre beneficia un ajuste en decimales.

Al comparar ambas gráficas podemos ver que nuestro modelo intenta asemejar el comportamiento de los puntos dispersos de los datos de la gráfica inicial. Nuestro modelo podemos decir que es funcional y práctico para este caso ya que su comportamiento si es muy similar, aunque el error obtenido no es tan pequeño como en el caso anterior, pero esto también influyen otros factores como la dispersión d ellos datos.

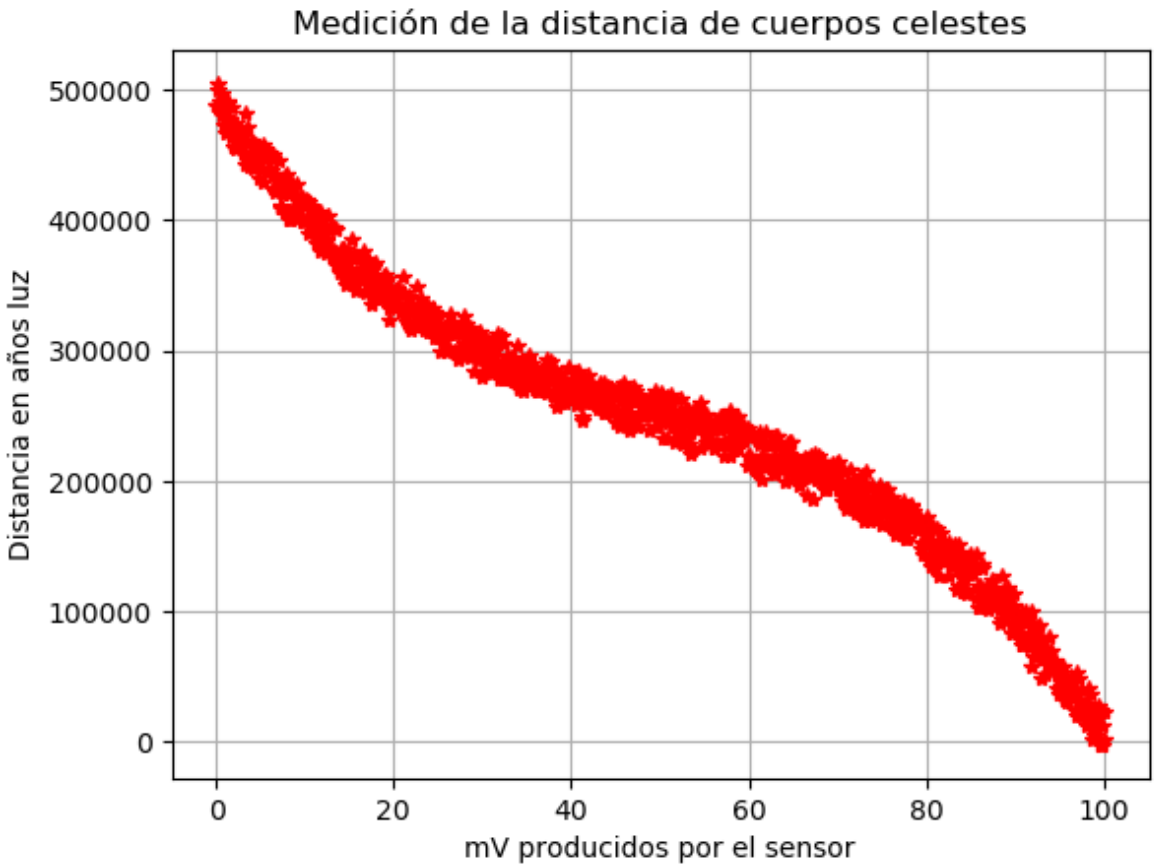
El grado utilizado para satisfacer las necesidades de este problema es una función cuadrática o de segundo grado, no tendría mucho caso aplicar grados más altos ya que el error no disminuiría considerablemente ya que la naturaleza de la dispersión de los datos originales no facilitaría esto, por lo que es suficiente una función de segundo grado.

# Problema 3: Calibración de sensores

## Análisis de datos

El Dr. Xavier López Rodríguez, un investigador de la UNAM desarrollo un sensor especializado de luz infrarroja, el cual permite estimar la distancia de los cuerpos celestes de manera directa a un bajo costo.

Se pretende añadir este sensor a una nueva línea de telescopios especializados que se lanzarán al público en los próximos meses. A fin de elaborar una interfaz que sea cómoda con el usuario se pretende crear un modelo que permita estimar la distancia de los cuerpos celestes en función de la carga eléctrica producida por el sensor, para lo cual se realizaron mediciones de cuerpos celestes ubicados entre 1 y 500,000 años luz de distancia.




Viendo el modelo lo que primero hacemos es definir nuestras entradas y salidas de nuestro sistema

- **Entradas:** X (mV producidos por el sensor)
- **Salidas:** Y (distancia en años luz)

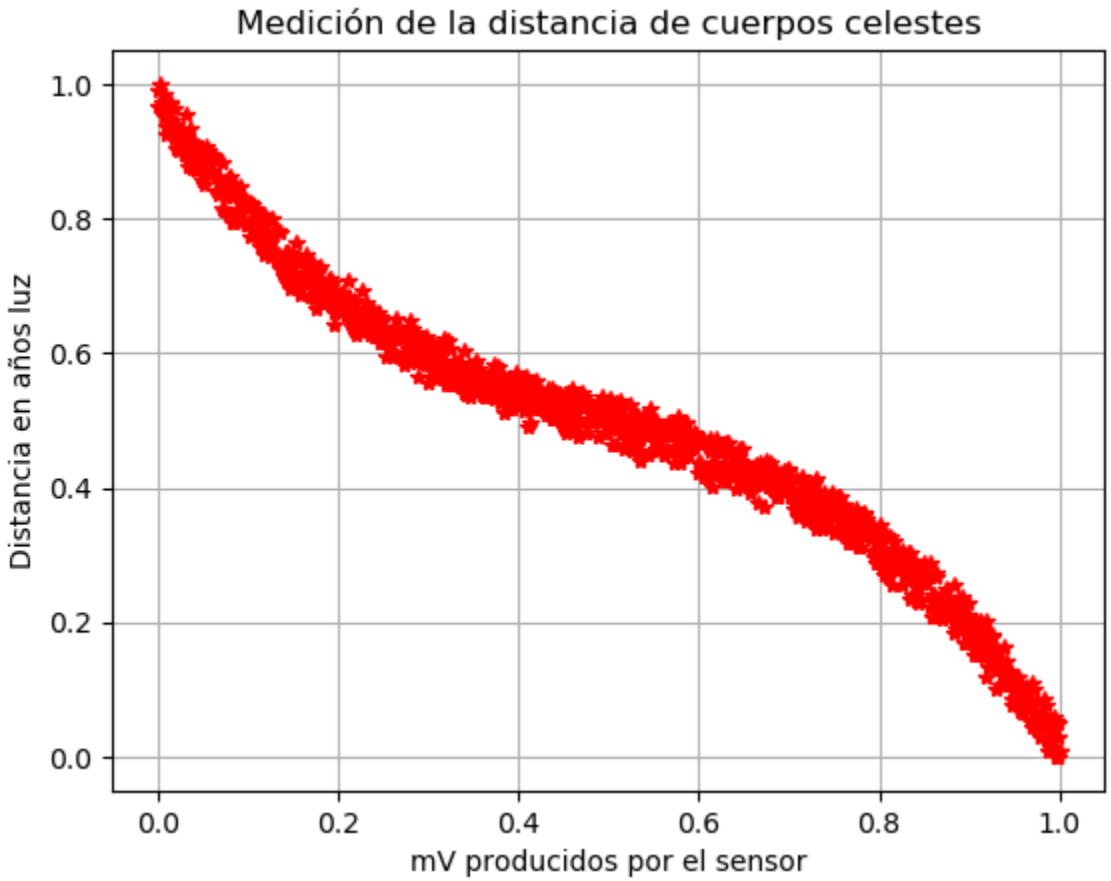
Al observar el modelo podemos ver que este modelo ambas matrices son de tamaño (1000,). También podemos observar a su vez que los valores que son manejados son números reales lo que nos lleva a determinar un modelo de Regresión.

El siguiente paso es normalizar nuestro modelo con el fin de tener valores limitados entre 0 y 1 en ambos ejes, para esto se aplican las siguientes fórmulas de normalización:

$$x = \frac{x - x_{min}}{x_{max}}$$

$$y = \frac{y - y_{min}}{y_{max}}$$

Al aplicar estas fórmulas obtenemos:




## Planteamiento del modelo

Empezamos en esta sección proponiendo nuestro modelo tomando de referencia los datos de entrada (X) y salida (Y). Para lograr el comportamiento de la gráfica se optó por usar una función cúbica.

$$h = ax^3 + bx^2 + cx + d$$

Posteriormente definimos nuestra función de error la cuál será el error medio cuadrático cuya fórmula es:

$$e = \frac{(y_m - y_d)^2}{2}$$

## Entrenamiento del modelo

Para poder optimizar nuestro problema a resolver se pretende minimizar a lo más posible nuestro error (e) mediante ajustar los parámetros de nuestro modelo parametrizado, es decir, de nuestra función

polinomial de cuarto orden (a, b, c, d). Para lograr lo anterior usamos el algoritmo del gradiente descendiente.

$$W_{k+1} = W_k - L_r \frac{\partial e}{\partial W}$$

Por la naturaleza de nuestros parámetros se debe de aplicar la regla de la cadena para obtener los gradientes correspondientes

Para llegar a nuestra solución primero debemos de proponer los gradientes

$$\frac{\partial e}{\partial a} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial a}$$
$$\frac{\partial e}{\partial b} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial b}$$
$$\frac{\partial e}{\partial c} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial c}$$
$$\frac{\partial e}{\partial d} = \frac{\partial e}{\partial y_m} \cdot \frac{\partial y_m}{\partial d}$$

Después encontramos los gradientes por separado

$$\frac{\partial e}{\partial y_m} = \frac{\partial}{\partial ym}\left[\frac{(y_m - y_d)^2}{2}\right] = y_m - y_d$$
$$\frac{\partial y_m}{\partial a} = \frac{\partial}{\partial a}[ax^3 + bx^2 + cx + d] = x^3$$
$$\frac{\partial y_m}{\partial b} = \frac{\partial}{\partial b}[ax^3 + bx^2 + cx + d] = x^2$$
$$\frac{\partial y_m}{\partial c} = \frac{\partial}{\partial c}[ax^3 + bx^2 + cx + d] = x$$
$$\frac{\partial y_m}{\partial d} = \frac{\partial}{\partial d}[ax^3 + bx^2 + cx + d] = 1$$

Por último, sustituyendo y aplicando la regla de la cadena podemos obtener nuestros gradientes

$$\frac{\partial e}{\partial a} = (y_m - y_d) \cdot x^3$$
$$\frac{\partial e}{\partial b} = (y_m - y_d) \cdot x^2$$
$$\frac{\partial e}{\partial c} = (y_m - y_d) \cdot x$$
$$\frac{\partial e}{\partial d} = (y_m - y_d)$$

## Resultados experimentales

Se creó una instancia para nuestro modelo con unos parámetros iniciales:

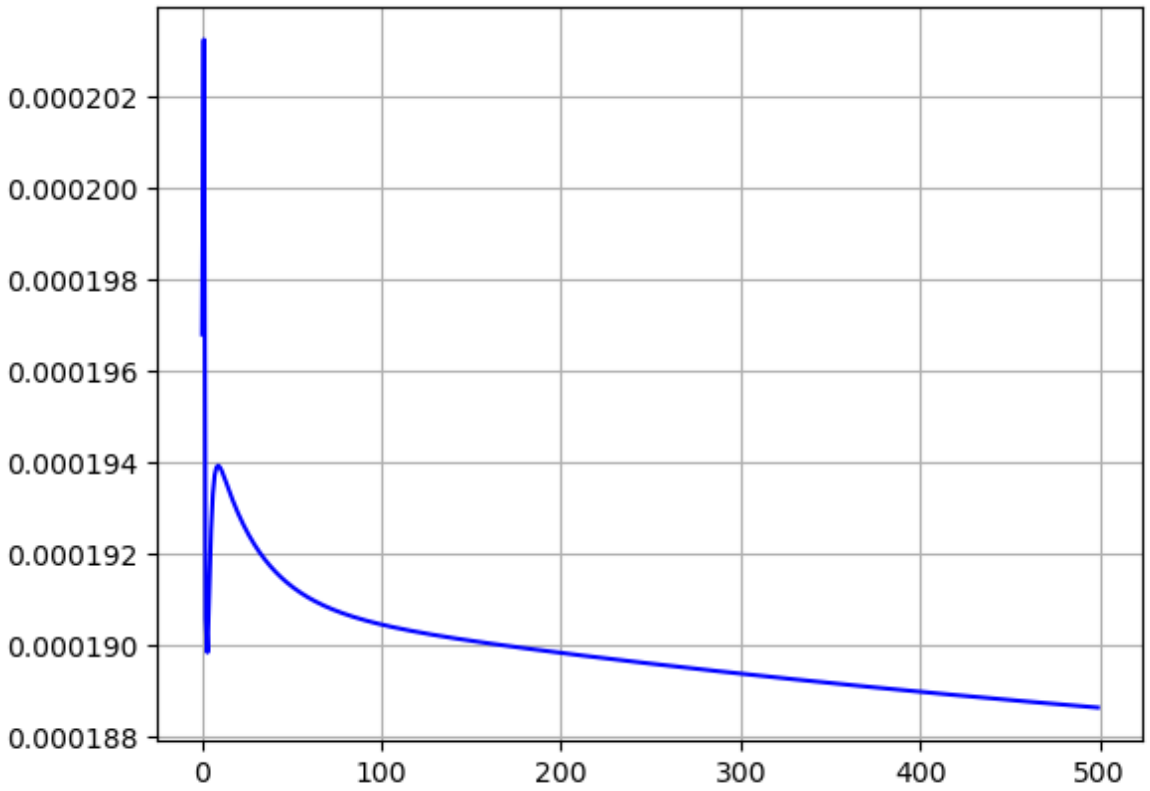
$$a = -2.0, \qquad b = 3.0, \qquad c = -2.0, \qquad d = 1.0$$

```
#crear instancia del modelo
sensores=cubica(-2.0,3.0,-2.0,1.0)
```

A este modelo se le sometió a un entrenamiento con una taza de aprendizaje $L_r = 0.01$ y durante 500 épocas.

```
#entrenar modelo
sensores.train(x,y,0.01,500)
```

Con dicho entrenamiento obtenemos cómo se comporta el error (error medio cuadrático):



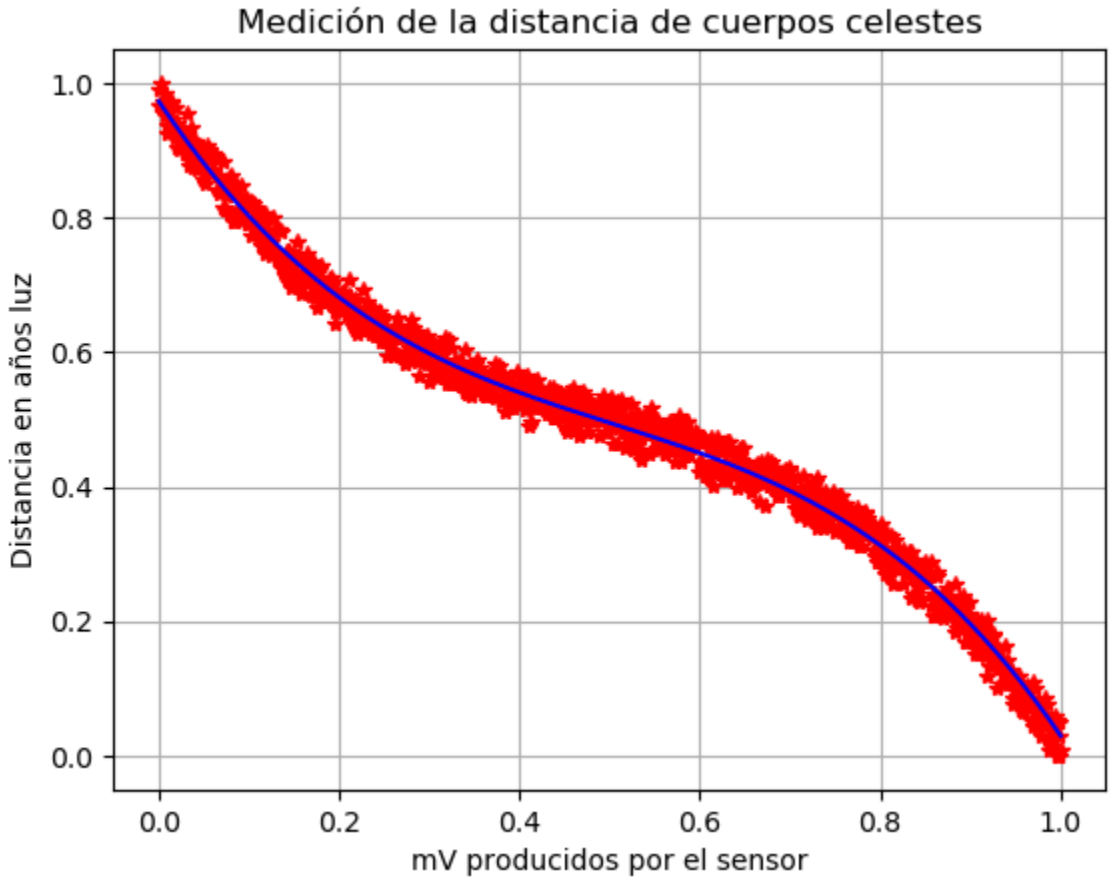

Con nuestro modelo parametrizado obtenemos un error igual a:

$$Error\ (e) = 0.00019446646615449154$$

Con unos parámetros de modelos finales iguales a:

$$a = -2.0554, \qquad b = 3.10875, \qquad c = -1.9963, \qquad d = 0.97255$$

Posteriormente podemos visualizar el comportamiento de nuestro modelo parametrizado (color azul) comparado con nuestra gráfica inicial obtenida de los datos (color rojo)




## Conclusiones

Analizando nuestros resultados después del entrenamiento podemos notar que los parámetros iniciales no estaban tan lejos de los parámetros finales.

Al comparar ambas gráficas podemos ver que nuestro modelo es prácticamente idéntico al de los datos de la gráfica inicial. Nuestro modelo podemos decir que es preciso y exacto en cuanto a apegarse al comportamiento de la gráfica de los sensores y esto también se ve reflejado con el error ya que el error es demasiado pequeño

El grado utilizado para satisfacer las necesidades de este problema es una función cúbica o de tercer grado, para este caso no tiene sentido seguir incrementando o aplicando funciones de polinomios más grandes, ya que la función cubica describe perfectamente el comportamiento.