Regresión Lineal Multivariable

-Aprendizaje de Máquina-

Lia Natalia Chicue Garcia

Fundamento Matemático

El Modelo de Regresión Lineal Multivariado elimina algunas limitaciones de los métodos anteriores, pues considera combinaciones lineales de funciones fijas no lineales de las variables de entrada[1].

$$\hat{y}_i(x_i) = m_o x_i + b_o$$

Que de forma matricial:

$$\begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & ... & x_{1m} \\ 1 & x_{21} & x_{22} & x_{23} & ... & x_{2m} \\ 1 & x_{31} & x_{32} & x_{33} & ... & x_{3m} \\ . & . & . & . & ... & . \\ . & . & . & . & ... & . \\ . & . & . & . & ... & . \\ 1 & x_{n1} & x_{n2} & x_{n3} & ... & x_{nm} \end{bmatrix}_{nx(m+1)} \begin{bmatrix} b_o \\ b_1 \\ b_2 \\ . \\ . \\ . \\ b_m \end{bmatrix}_{m+1} = \begin{bmatrix} y_o \\ y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix}_{n+1}$$

$$[\mathbf{X}][\mathbf{b}] = [\mathbf{Y}]$$

Con el fin de despejar [**b**], tenemos:

$$[\mathbf{X}]^T[\mathbf{X}][\mathbf{b}] = [\mathbf{X}]^T[\mathbf{Y}]$$
$$([\mathbf{X}]^T[\mathbf{X}])[\mathbf{b}] = [\mathbf{X}]^T[\mathbf{Y}]$$
$$([\mathbf{X}]^T[\mathbf{X}])^-1([\mathbf{X}]^T[\mathbf{X}])[\mathbf{b}] = ([\mathbf{X}]^T[\mathbf{X}])^-1[\mathbf{X}]^T[\mathbf{Y}]$$

Siendo $([\mathbf{X}]^T[\mathbf{X}])^-1([\mathbf{X}]^T[\mathbf{X}]) = I$

$$\mathbf{I}[\mathbf{b}] = ([\mathbf{X}]^T[\mathbf{X}])^-1[\mathbf{X}]^T[\mathbf{Y}]$$
$$[\mathbf{b}] = \frac{([\mathbf{X}]^T[\mathbf{X}])^-1[\mathbf{X}]^T[\mathbf{Y}]}{\mathbf{I}}$$
$$[\mathbf{b}] = ([\mathbf{X}]^T[\mathbf{X}])^-1[\mathbf{X}]^T[\mathbf{Y}] \quad (1)$$

Caso Clasificación de Cuerpos Celestes

Presentación de la Base de datos

La base de datos aquí presentada se puede encontrar en kaggle como "Star Type Classification / NASA". El objetivo de esta base de datos se refiere a la predicción del tipo de estrella según medidas astrofísicas. Contiene 240 valores y 7 características. Para este caso, utilicemos la ultima característica como nuestro

target el cual, es un conjunto de datos que va de 0 a 5, de la forma 0: Enana Roja, 1: Enana Marrón, 2: Enana Blanca, 3: Secuencia Principal, 4: Super Gigantes y 5: Hiper Gigantes.

Por tanto, las características son:

- Temperature K
- L L/Lo (Avg Luminosity of Sun)
- R R/Ro (Avg Radius of Sun)
- AM – Mv
- Color: General Color of Spectrum
- Spectral Class: O,B,A,F,G,K,M / SMASS Ingresar aquí Para más información.

Codificación de Datos

Debido a que dos de las características de la base de datos son categorías, se debe realizar una codificación de sus datos con el fin de obtener valores cuantificables.

```
import numpy as np
import pandas as pd
from google.colab import files

    #Leemos la base de datos
    dfStars = pd.read_csv('Stars.csv')

    #Convertimos los datos categ ricos a datos num ricos
    dfStars["Color"] = dfStars["Color"].astype('category').cat.codes
    dfStars["Spectral_Class"] = dfStars["Spectral_Class"].astype('category').cat.codes

    #Convertimos la base de datos en tipo .xlsx y guardamos
    dfStars.to_excel("dfStars.xlsx",engine='xlsxwriter')
```

Listing 1: Codificación de Datos

	Temperature	L	R	A_M	Color	Spectral_Class	Type
0	3068	0.002400	0.1700	16.12	Red	M	0
1	3042	0.000500	0.1542	16.60	Red	M	0
2	2600	0.000300	0.1020	18.70	Red	M	0
3	2800	0.000200	0.1600	16.65	Red	M	0
4	1939	0.000138	0.1030	20.06	Red	M	0

Figure 1: Dataset con datos Categóricos.

	Temperature	L	R	A_M	Color	Spectral_Class	Type
0	3068	0.002400	0.1700	16.12	8	5	0
1	3042	0.000500	0.1542	16.60	8	5	0
2	2600	0.000300	0.1020	18.70	8	5	0
3	2800	0.000200	0.1600	16.65	8	5	0
4	1939	0.000138	0.1030	20.06	8	5	0

Figure 2: Dataset con sus datos Categóricos convertidos a númericos.

Implementación en MATLAB

Con el preprocesamiento de datos realizado, pasamos a la implementación del modelo en MATLAB.

La base de datos empleada posee características con pesos variables, lo cual puede generar problemas en la predicción. Se debe, por tanto, realizar la normalización de los datos; tanto del conjunto de características como el target.

```
%Leemos la base de datos
 [D, response, raw] = xlsread('dfStars.xlsx');
 [F C]=size(D);

%Realizamos Normalizacion
 Xnorm1 = (D(:,1) - (mean(D(:,1))))/std(D(:,1));%Temperatura

 Xnorm2 = (D(:,2) - (mean(D(:,2))))/std(D(:,2));%Valor promedio de Luminosidad con
    respecto al Sol (L)

 Xnorm3 = (D(:,3) - (mean(D(:,3))))/std(D(:,3));%Valor promedio Radial con respecto al
    Sol (R)

 Xnorm4 = (D(:,4) - (mean(D(:,4))))/std(D(:,4));%AM -Mv

 Xnorm5 = (D(:,5) - (mean(D(:,5))))/std(D(:,5));%Color - Color General del Espectro

 Xnorm6 = (D(:,6) - (mean(D(:,6))))/std(D(:,6));%Clase espectral

 Xnorm7 = (D(:,7) - (mean(D(:,7))))/std(D(:,7));%Tipo de Asteroide Espectral
```

Listing 2: Normalización de los datos

```
%Dividimos nuestro dataset
%Caracteristicas
 X = [ones(F,1) Xnorm1 Xnorm2 Xnorm3 Xnorm4 Xnorm5 Xnorm6];
%Target
 Y = Xnorm7;

%Proceso Matematico
 Xtrans = X';
 B = inv(Xtrans*X)*Xtrans*Y;

%Realizamos la Prediccion
 Ypredict = X*B;

%Mean Squared Error(MSE)
 MSE = (sum((Y - Ypredict).^2)/F);

%Root Mean Squared Error (RMSE)
 RMSE = (MSE)^0.5;

%Mean Absolute Error (MAE)
 MAE = (sum(abs(Y - Ypredict))/F);

%Graficamos
%Error de cada instancia
 figure(1)
 bar(Y - Ypredict)
 title('Residuals')

%Visualizacion de las Predicciones vs Los Valores reales
 figure(2)
 scatter(Y,Ypredict)
 title('Predicted vs. Actual')

%Matriz de Confusion
 plotconfusion(Y,Ypredict)
 C = confusionmat(Y,Ypredict);%Encuentra los datos
 figure(3)
 confusionchart(Y,Ypredict)
```

Listing 3: Implementación en MATLAB

Implementación en Regression Learner MATLAB

A continuación observemos cómo se implementa el modelo anterior en la aplicación de MATLAB *RegressionLearner*. Cargamos la aplicación, seleccionamos el dataset y entrenamos con el modelo de regresión linear.

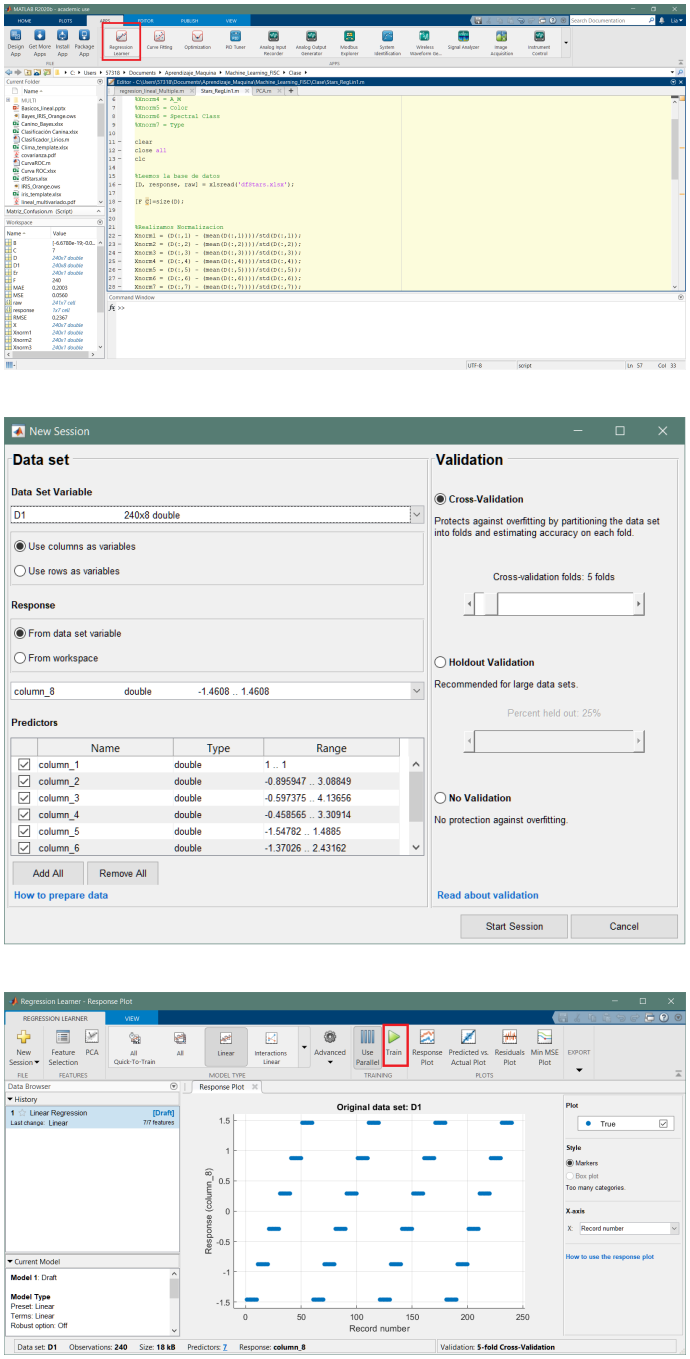

Figure 3: Manejo de Aplicación *RegressionLearner*

Resultados

Para la implementación en MATLAB, la figura 5 muestra un comportamiento linear. Esto nos indica que la predicción hallada tiene sentido. Corroboramos también los errores:

- Mean Squared Error(MSE)=0.056

- Root Mean Squared Error (RMSE)=0.236
- Mean Absolute Error (MAE)=0.200

Además, tenemos los resultados de la matriz de confusión 6 en la cual observamos que el modelo tuvo una respuesta.

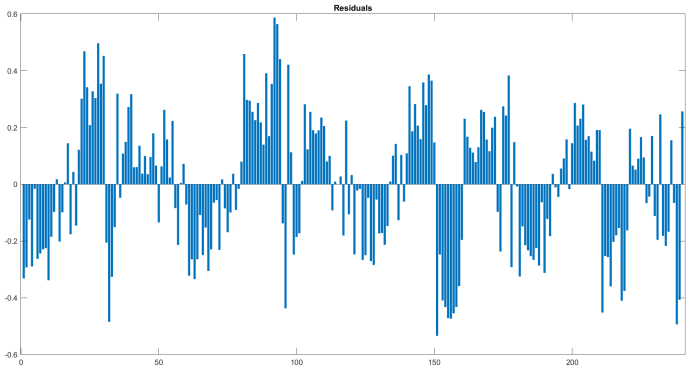

Figure 4: Gráfica de Valores Residuales $Y - Ypredictivos$.

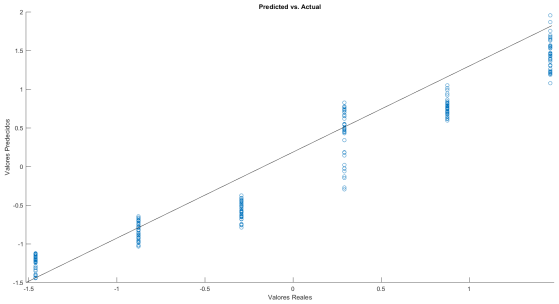

Figure 5: Gráfica de valores Predecidos vs. Reales.

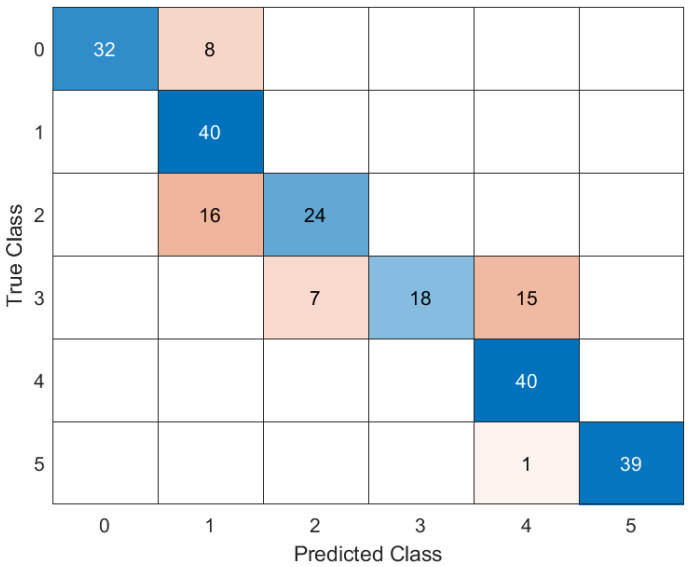

Figure 6: Gráfica de Matriz de Confusión.

Para la implementación con la Aplicación *RegressionLearner*, tenemos las figuras:

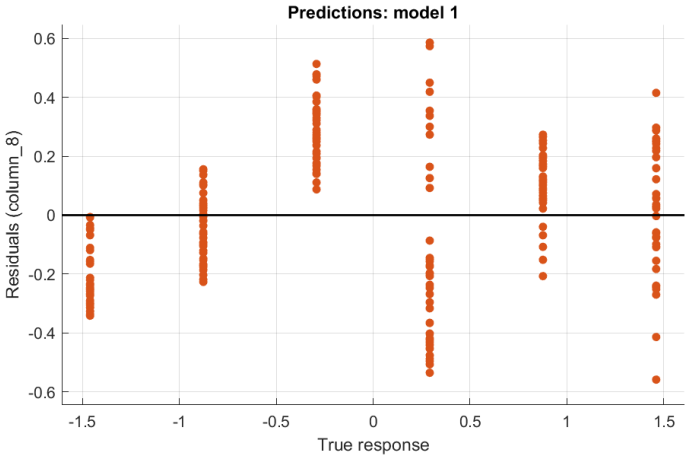

Figure 7: Valores Residuales

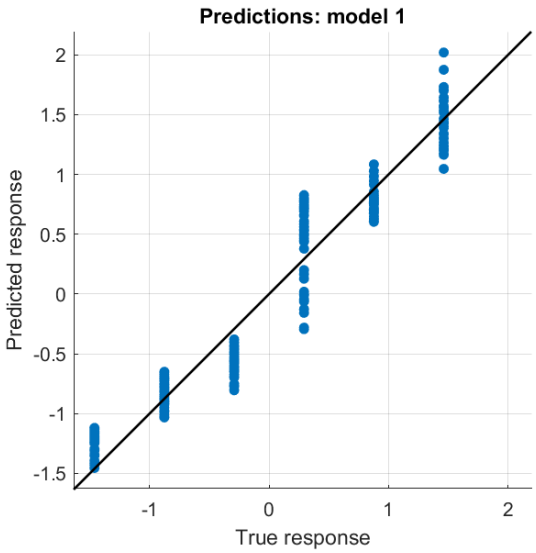

Figure 8: Valores Predecidos vs. Valores reales hallado por Regression Learner

Y sus errores respectivos:

- Mean Squared Error(MSE)=0.058
- Root Mean Squared Error (RMSE)=0.243
- Mean Absolute Error (MAE)=0.205

References

[1] Yuichiro Anzai. *Pattern recognition and machine learning.* Elsevier, 2012.