



Universidad autónoma de baja California

Ingeniería en computación

Inteligencia artificial

Meta 5.1: Aprendizaje supervisado

Erik garcia Chávez 01275973

Juan Ramon castro rodriguez

28 de mayo del 2025

índice

Teoría fórmulas para regresión polinomial multivariable:	3
Resultados del aprendizaje:	5
Regresión polinomial multivariable utilizando el optimizador GDX:.....	5
Regresión polinomial multivariable utilizando AdamD:.....	7
Teoría clasificación: regresión logística – regresión softmax:	10
Resultados del aprendizaje:	12
Clasificador polinomial multivariable utilizando GDX – regresión logística:.....	12
clasificador polinomial multivariable utilizando GDX – regresión	
SoftMax:	13
<i>Clasificación polinomial multivariable – AdamD – regresión logística:</i>	<i>15</i>
Clasificación polinomial multivariable – AdamD – SoftMax :	16
<i>Clasificación polinomial multivariable – multiclase:</i>	<i>18</i>
<i>Clasificador polinomual multivariable – multiclass- GDX- regresión</i>	
<i>logística:.....</i>	<i>18</i>
Clasificador polinomial multivariable – multiclass - GDX – regresor	
softmax:	20
<i>Clasificador polinomial multivariable – multiclass – AdamD – regresión</i>	
<i>logística:.....</i>	<i>22</i>
<i>Clasificador polinomial multivariable – multiclass – AdamD – regresión</i>	
<i>SoftMax:</i>	<i>24</i>

Teoría fórmulas para regresión polinomial multivariable:

El modelo de regresión polinomial multivariable de grado (τ) permite capturar relaciones no lineales entre múltiples variables productoras. Y una variable de respuesta

$$P_{\tau}(\mathbf{x}) = \theta_0 + \sum_{l_1=1}^n \theta_{l_1} x_{l_1} + \sum_{l_1=1}^n \sum_{l_2=l_1}^n \theta_{l_1, l_2} x_{l_1} x_{l_2} + \sum_{l_1=1}^n \sum_{l_2=l_1}^n \cdots \sum_{l_{\tau}=l_{\tau-1}}^n \theta_{l_1, l_2, \dots, l_{\tau}} x_{l_1} x_{l_2} \cdots x_{l_{\tau}}$$

Donde:

$\mathbf{X} = [x_1, x_2, \dots, x_n]$: vector de variables de entrada

τ = grado máximo del polinomio

$\theta_{l_1, l_2, \dots, l_{\tau}}$: coeficiente del termino de combinación.

Modificación de la función para calcular el **número de parámetros**,

def polyParamsNumber(n, tau),

razon:

los factores crecen exponencialmente y causan desbordamiento para valores moderados de n o τ , tiene un mayor costo computacional.

Beneficios de la nueva versión:

Utiliza una identidad combinatoria equivalente:

$$\sum_{l=0}^{\tau} \binom{n+l-1}{l} = \binom{n+\tau}{\tau}$$

Identidad de Hockey-stick

```
1 def polyParamsNumber(n, tau):
2     return int(math.comb(n + tau, tau))
3
4
```

- La función **mat.comb** calcula combinaciones directamente sin factoriales intermedios.

- Evita números gigantescos
- Es mucho más simple.

Explicación de la función y ecuaciones perdida

Tenemos a:

$$SSE = \|\Xi\|_F^2 = \text{tr}(\Xi^T \Xi)$$

Donde $E = Y_{\text{verdadera}} - Y_{\text{pred}}$ es la matriz de errores

$\|\cdot\|_F$ es la norma de Frobenius

SSE con norma de Frobenius: es el estándar para modelos multivariados.

Término de regularización Ridge (L2)

$$\hat{\theta} = \min_{\theta} \|y - A\theta\|_2^2 + \lambda \|\theta\|_2^2 = \min_{\theta} SSE + \lambda \|\theta\|_2^2$$

Donde $\lambda \|\theta\|_2^2$ excluye el término intercepto (θ_0) para evitar su penalización.

Dentro de la función **THETA[1:,:]** excluye la primera fila, parámetros θ_1 a θ_n dejando fuera θ_0 .

Función *desingMatrix(Tau, X)*:

Esta función construye la matriz de diseño para regresión polinómica, transforma la matriz de entrada X en una matriz con todas las combinaciones polinómicas de sus características hasta grado **tau**.

Función *poweVector(Tau, X)*:

Esta función genera todas las combinaciones polinómicas de un vector de características X hasta un grado máximo Tau , usando recursividad.

Resultados del aprendizaje:

Regresión polinomial multivariable utilizando el optimizador GDX:

Datos de entrenamiento: engine_dataset.mat

Mini lotes

```
% R2 for raw train data  
R2_train = r2_score(targets_train_descaled, outputTrain_descaled.reshape(-1, 1))  
print(R2_train)
```

0.9503984993728382

```
# MSE for raw train data  
MSE_train = mean_squared_error(targets_train_descaled, outputTrain_descaled.reshape(-1, 1))  
print(MSE_train)
```

12935.872327524816

```
# R2 for raw test data  
R2_test = r2_score(targets_test_descaled, outputTest_descaled.reshape(-1, 1))  
print(R2_test)
```

0.9574186002109912

```
# MSE for raw test data  
MSE_test = mean_squared_error(targets_test_descaled, outputTest_descaled.reshape(-1, 1))  
print(MSE_test)
```

12035.92286675407

THETA

```
array([[ 0.01958919,  0.35407876],  
       [ 0.97937582,  1.0734356 ],  
       [-0.05531105, -1.09346678],  
       [ 0.03358722, -0.23560476],  
       [ 0.09185862, -0.10700749],  
       [-0.04545827, -0.17645382]])
```

Online

```
print(MSE_train)

0.9501071392602626

: # MSE for raw train data
MSE_train = mean_squared_error(targets_train_descaled, outputTrain_descaled.reshape(-1, 1))
print(MSE_train)

13011.857875743231

: # R2 for raw test data
R2_test = r2_score(targets_test_descaled, outputTest_descaled.reshape(-1, 1))
print(R2_test)

0.9549455024165753

: # MSE for raw test data
MSE_test = mean_squared_error(targets_test_descaled, outputTest_descaled.reshape(-1, 1))
print(MSE_test)

12734.960813909878

: THETA

array([[ 0.01388956,  0.27247694],
       [ 0.94568848,  1.00001092],
       [-0.03633054, -0.98021644],
       [ 0.027629   , -0.21647424],
       [ 0.08598198, -0.11202569],
       [-0.04392353, -0.15354643]])

:
```

Lote completo:

```
: # R2 for raw train data
R2_train = r2_score(targets_train_descaled, outputTrain_descaled.reshape(-1, 1))
print(R2_train)

0.9520533098628233

: # MSE for raw train data
MSE_train = mean_squared_error(targets_train_descaled, outputTrain_descaled.reshape(-1, 1))
print(MSE_train)

12504.30439199801

: # R2 for raw test data
R2_test = r2_score(targets_test_descaled, outputTest_descaled.reshape(-1, 1))
print(R2_test)

0.9581312107999834

: # MSE for raw test data
MSE_test = mean_squared_error(targets_test_descaled, outputTest_descaled.reshape(-1, 1))
print(MSE_test)

11834.498626930097

: THETA

: array([[ 0.01984981,  0.30785984],
       [ 0.9752518 ,  1.048895  ],
       [-0.05860286, -1.08462375],
       [ 0.0249616 , -0.2224697 ],
       [ 0.0914151 , -0.10370667],
       [-0.046274 , -0.1773417 ]])
```

Regresión polinomial multivariable utilizando AdamD:

datos de entrenamiento: engiene_dataset.mat

Mini lote:

```
print(R2_train)

0.9392825174822423

6]: MSE_train = mean_squared_error(tTrain_original, outputTrain)
print(MSE_train)

12601.903725260476

7]: # Test data
R2_test = r2_score(tTest_original, outputTest)
print(R2_test)

0.9489383729778305

8]: MSE_test = mean_squared_error(tTest_original, outputTest)
print(MSE_test)

12093.46994797191

9]: THETA

9]: array([[ 0.0131116 ,  0.3159656 ],
          [ 0.96851457,  1.06278731],
          [-0.04771009, -1.05697628],
          [ 0.03083468, -0.22194714],
          [ 0.09303978, -0.11221019],
          [-0.04727476, -0.16912499]])
```

Online:

```
print(R2_train)
```

```
0.8608386135212868
```

```
[16]: MSE_train = mean_squared_error(tTrain_original, outputTrain)
      print(MSE_train)
```

```
30564.5184847236
```

```
[17]: # Test data
      R2_test = r2_score(tTest_original, outputTest)
      print(R2_test)
```

```
0.862740391839625
```

```
[18]: MSE_test = mean_squared_error(tTest_original, outputTest)
      print(MSE_test)
```

```
34073.7039908613
```

```
[19]: THETA
```

```
[19]: array([[ -0.00855305,  0.08414193],
          [ 0.76468938,  0.74624405],
          [ 0.04609344, -0.57032626],
          [ 0.05426993, -0.16298801],
          [ 0.06850528, -0.09729734],
          [-0.03958482, -0.09762124]])
```


Lote completo:

```
print(R2_train)

0.9393667313012334

[16]: MSE_train = mean_squared_error(tTrain_original, outputTrain)
print(MSE_train)

12590.29625323632

[17]: # Test data
R2_test = r2_score(tTest_original, outputTest)
print(R2_test)

0.9490973201716718

[18]: MSE_test = mean_squared_error(tTest_original, outputTest)
print(MSE_test)

12064.092936739908

[19]: THETA

[19]: array([[ 0.01384425,  0.32816749],
 [ 0.9629813 ,  1.05766762],
 [-0.04753597, -1.07709777],
 [ 0.03195839, -0.22140541],
 [ 0.09642981, -0.10781042],
 [-0.0500841 , -0.18561584]])
```

Teoría clasificación: regresión logística – regresión softmax:

El objetivo es predecir la probabilidad de que una instancia pertenezca a una clase específica. Una probabilidad puede ser una salida entre 0 y 1. (función sigmoide)

El modelo se entra mediante una serie de elementos, como la función de costo, la cual mide el error entre las probabilidades predichas y las etiquetas reales.

El uso de un optimizador que ajusta los coeficientes para minimizar la pérdida logarítmica, como en nuestro caso es GDX y AdamD.

Se usa **One-vs-Rest** o **SoftMax Regression** para una clasificación de múltiples clases.

Sigmoide logística:

$$S \equiv P(Y = k|X = x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}}$$

Esta función transforma cualquier valor numérico en el rango [0,1], lo que permite interpretar su salida como una **probabilidad**

Donde **P(Y=K | X=x)** es la probabilidad de que la variable objetivo **Y** sea 1, dado el predictor **X=x**

Por lo que la sigmoide es el puente entre la regresión lineal (genera valores continuos) y la clasificación binaria

Función de costo:

$$\mathcal{L}(\theta) = - \sum_{p=1}^q \sum_{j=1}^m [y_{p,j} \ln(S_{p,j}) + (1 - y_{p,j}) \ln(1 - S_{p,j})]$$

Donde:

q: número de muestras

y_p: etiqueta real (0 o 1) para la muestra p

S_p: probabilidad predicha por el modelo

Theta: parámetros del modelo

Dentro del código se implementa epsilon como una protección extra ya que si h = 0 o h=1, los logaritmos devuelven -inf causando errores, por lo que la función **np.clip()**, fuerza las predicciones al rango [epsilon, 1-epsilon]

softmax

la función softmax es un componente fundamental en la clasificación de multiclase, su objetivo es convertir un vector de valores numéricos arbitrarios en probabilidades multinomiales normalizadas, donde cada valor representa la probabilidad de pertenecer a una clase específica.

Formula:

$$S_{p,j} = \text{softmax}(h_{\theta}(x)) = \frac{\exp(h_{\theta,j}(x_{p,:}))}{\sum_{k=1}^m \exp(h_{\theta,k}(x_{p,:}))}$$

Donde

M : numero de clases

H: vector de logits

Exp(h_j) exponencial del logit de la clase j

Esta formula nos sirve para transformar salidas numéricas en probabilidades comprensibles.

Resultados del aprendizaje:

Clasificador polinomial multivariable utilizando GDX – regresión logística:

Utilizando datos de: **cancer_dataset.dat**

Mini lotes:

datos de entrenamiento:

Accuracy: 0.9857

matriz de confusion:

```
[[321  7]
 [  0 161]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.98	0.99	328
1	0.96	1.00	0.98	161
accuracy			0.99	489
macro avg	0.98	0.99	0.98	489
weighted avg	0.99	0.99	0.99	489

Datos de prueba:

Accuracy: 0.9857

matriz de confusion:

```
[[128  2]
 [  1  79]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	0.99	0.98	0.99	130
1	0.98	0.99	0.98	80
accuracy			0.99	210
macro avg	0.98	0.99	0.98	210
weighted avg	0.99	0.99	0.99	210

Online:

Datos de entrenamiento

Accuracy: 0.9816

matriz de confusion:

```
[[325  3]
 [  6 155]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	328
1	0.98	0.96	0.97	161
accuracy			0.98	489
macro avg	0.98	0.98	0.98	489
weighted avg	0.98	0.98	0.98	489

Datos de prueba

Accuracy: 0.9810

matriz de confusion:

```
[[129  1]
 [  3  77]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	0.98	0.99	0.98	130
1	0.99	0.96	0.97	80
accuracy			0.98	210
macro avg	0.98	0.98	0.98	210
weighted avg	0.98	0.98	0.98	210

Lote:

Datos de entrenamiento: Datos de prueba:

Accuracy: 0.9796

matriz de confusion:

```
[[318 10]
 [ 0 161]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.97	0.98	328
1	0.94	1.00	0.97	161
accuracy			0.98	489
macro avg	0.97	0.98	0.98	489
weighted avg	0.98	0.98	0.98	489

Accuracy: 0.9810

matriz de confusion:

```
[[127 3]
 [ 1 79]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	0.99	0.98	0.98	130
1	0.96	0.99	0.98	80
accuracy			0.98	210
macro avg	0.98	0.98	0.98	210
weighted avg	0.98	0.98	0.98	210

**clasificador polinomial multivariable utilizando GDX – regresión SoftMax:
utilizando datos de: cancer_dataset.dat**

mini lote:

datos de entrenamiento

Resultados en conjunto de entrenamiento:

Accuracy: 0.9959

Matriz de confusión:

```
[[326 2]
 [ 0 161]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.99	1.00	328
1	0.99	1.00	0.99	161
accuracy			1.00	489
macro avg	0.99	1.00	1.00	489
weighted avg	1.00	1.00	1.00	489

Datos de prueba:

Resultados en conjunto de prueba:

Accuracy: 0.9952

Matriz de confusión:

```
[[129 1]
 [ 0 80]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.99	1.00	130
1	0.99	1.00	0.99	80
accuracy			1.00	210
macro avg	0.99	1.00	0.99	210
weighted avg	1.00	1.00	1.00	210

Online:

Datos de entrenamiento:

resultados en conjunto de entrenamiento:

Accuracy: 0.9959

matriz de confusion:

```
[[326  2]
 [  0 161]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.99	1.00	328
1	0.99	1.00	0.99	161
accuracy			1.00	489
macro avg	0.99	1.00	1.00	489
weighted avg	1.00	1.00	1.00	489

datos de prueba

resultados en conjunto de prueba:

Accuracy: 0.9905

matriz de confusion:

```
[[129  1]
 [  1 79]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	130
1	0.99	0.99	0.99	80
accuracy			0.99	210
macro avg	0.99	0.99	0.99	210
weighted avg	0.99	0.99	0.99	210

Lote:

Datos de entrenamiento:

resultados en conjunto de entrenamiento:

Accuracy: 0.9796

matriz de confusion:

```
[[318 10]
 [  0 161]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.97	0.98	328
1	0.94	1.00	0.97	161
accuracy			0.98	489
macro avg	0.97	0.98	0.98	489
weighted avg	0.98	0.98	0.98	489

datos de prueba:

resultados en conjunto de prueba:

Accuracy: 0.9857

matriz de confusion:

```
[[128  2]
 [  1 79]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	0.99	0.98	0.99	130
1	0.98	0.99	0.98	80
accuracy			0.99	210
macro avg	0.98	0.99	0.98	210
weighted avg	0.99	0.99	0.99	210

Clasificación polinomial multivariable – AdamD – regresión logística:

Utilizando datos de: cancer_dataset.dat

Mini lote:

Datos de entrenamiento

resultados en conjunto de entrenamiento:

Accuracy: 0.9775

Matriz de confusión:

```
[[318 10]
```

```
[ 1 160]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.97	0.98	328
1	0.94	0.99	0.97	161
accuracy			0.98	489
macro avg	0.97	0.98	0.97	489
weighted avg	0.98	0.98	0.98	489

datos de prueba

resultados en conjunto de prueba:

Accuracy: 0.9810

Matriz de confusión:

```
[[128 2]
```

```
[ 2 78]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	130
1	0.97	0.97	0.97	80
accuracy			0.98	210
macro avg	0.98	0.98	0.98	210
weighted avg	0.98	0.98	0.98	210

Online:

Datos de entrenamiento:

resultados en conjunto de entrenamiento:

Accuracy: 0.9959

Matriz de confusión:

```
[[326 2]
```

```
[ 0 161]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.99	1.00	328
1	0.99	1.00	0.99	161
accuracy			1.00	489
macro avg	0.99	1.00	1.00	489
weighted avg	1.00	1.00	1.00	489

datos de prueba:

resultados en conjunto de prueba:

Accuracy: 0.9952

Matriz de confusión:

```
[[129 1]
```

```
[ 0 80]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.99	1.00	130
1	0.99	1.00	0.99	80
accuracy			1.00	210
macro avg	0.99	1.00	0.99	210
weighted avg	1.00	1.00	1.00	210

Lote:

Datos de entrenamiento:

resultados en conjunto de entrenamiento:

Accuracy: 0.9775

Matriz de confusión:

```
[[318 10]
 [ 1 160]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.97	0.98	328
1	0.94	0.99	0.97	161
accuracy			0.98	489
macro avg	0.97	0.98	0.97	489
weighted avg	0.98	0.98	0.98	489

datos de prueba:

resultados en conjunto de entrenamiento:

Accuracy: 0.9775

Matriz de confusión:

```
[[318 10]
 [ 1 160]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.97	0.98	328
1	0.94	0.99	0.97	161
accuracy			0.98	489
macro avg	0.97	0.98	0.97	489
weighted avg	0.98	0.98	0.98	489

Clasificación polinomial multivariable – AdamD – SoftMax :

Utilizando datos de: cancer_dataset.dat

Mini lotes:

Datos de entrenamiento:

Resultados en conjunto de entrenamiento:

Accuracy: 0.9836

Matriz de confusión:

```
[[321 7]
 [ 1 160]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.98	0.99	328
1	0.96	0.99	0.98	161
accuracy			0.98	489
macro avg	0.98	0.99	0.98	489
weighted avg	0.98	0.98	0.98	489

datos de prueba.

Resultados en conjunto de prueba:

Accuracy: 0.9857

Matriz de confusión:

```
[[128 2]
 [ 1 79]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	0.99	0.98	0.99	130
1	0.98	0.99	0.98	80
accuracy			0.99	210
macro avg	0.98	0.99	0.98	210
weighted avg	0.99	0.99	0.99	210

Online:

Datos de entrenamiento:

Resultados en conjunto de entrenamiento:

Accuracy: 0.9939

Matriz de confusión:

```
[[326  2]
 [  1 160]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.99	1.00	328
1	0.99	0.99	0.99	161
accuracy			0.99	489
macro avg	0.99	0.99	0.99	489
weighted avg	0.99	0.99	0.99	489

datos de prueba:

Resultados en conjunto de prueba:

Accuracy: 0.9762

Matriz de confusión:

```
[[128  2]
 [  3  77]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	130
1	0.97	0.96	0.97	80
accuracy			0.98	210
macro avg	0.98	0.97	0.97	210
weighted avg	0.98	0.98	0.98	210

Lote:

Datos de entrenamiento:

Resultados en conjunto de entrenamiento:

Accuracy: 0.9796

Matriz de confusión:

```
[[319  9]
 [  1 160]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.97	0.98	328
1	0.95	0.99	0.97	161
accuracy			0.98	489
macro avg	0.97	0.98	0.98	489
weighted avg	0.98	0.98	0.98	489

datos de prueba:

Resultados en conjunto de prueba:

Accuracy: 0.9857

Matriz de confusión:

```
[[128  2]
 [  1  79]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	0.99	0.98	0.99	130
1	0.98	0.99	0.98	80
accuracy			0.99	210
macro avg	0.98	0.99	0.98	210
weighted avg	0.99	0.99	0.99	210

Clasificación polinomial multivariable – multiclase:

Clasificador polinomial multivariable – multiclass- GDX- regresión logística:

Utilizando datos de: dermatology.dat

Mini lote:

Datos entrenamiento:

datos de prueba:

Rendimiento en entrenamiento (multiclase):
Accuracy: 0.8594

Rendimiento en prueba (multiclase):
Accuracy: 0.8273

Matriz de confusión:
[[72 0 0 0 0 0]
[0 45 0 0 0 0]
[0 0 53 0 0 0]
[0 32 0 0 1 1]
[0 0 0 0 34 2]
[0 0 0 0 0 16]]

Matriz de confusión:
[[38 0 0 0 0 2]
[0 16 0 0 0 0]
[0 0 19 0 0 0]
[0 15 0 0 0 0]
[0 1 0 0 14 1]
[0 0 0 0 0 4]]

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	72
1	0.58	1.00	0.74	45
2	1.00	1.00	1.00	53
3	0.00	0.00	0.00	34
4	0.97	0.94	0.96	36
5	0.84	1.00	0.91	16
accuracy			0.86	256
macro avg	0.73	0.82	0.77	256
weighted avg	0.78	0.86	0.81	256

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.95	0.97	40
1	0.50	1.00	0.67	16
2	1.00	1.00	1.00	19
3	0.00	0.00	0.00	15
4	1.00	0.88	0.93	16
5	0.57	1.00	0.73	4
accuracy			0.83	110
macro avg	0.68	0.80	0.72	110
weighted avg	0.78	0.83	0.79	110

Online:

Datos de entrenamiento

rendimiento en entrenamiento (multiclase):
Accuracy: 0.9883

matriz de confusion:
[[72 0 0 0 0 0]
[0 43 0 2 0 0]
[0 0 53 0 0 0]
[0 1 0 33 0 0]
[0 0 0 0 36 0]
[0 0 0 0 0 16]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	72
1	0.98	0.96	0.97	45
2	1.00	1.00	1.00	53
3	0.94	0.97	0.96	34
4	1.00	1.00	1.00	36
5	1.00	1.00	1.00	16
accuracy			0.99	256
macro avg	0.99	0.99	0.99	256
weighted avg	0.99	0.99	0.99	256

datos de prueba:

rendimiento en prueba (multiclase):
Accuracy: 0.9727

matriz de confusion:
[[39 1 0 0 0 0]
[0 15 0 1 0 0]
[0 0 19 0 0 0]
[0 1 0 14 0 0]
[0 0 0 0 16 0]
[0 0 0 0 0 4]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.97	0.99	40
1	0.88	0.94	0.91	16
2	1.00	1.00	1.00	19
3	0.93	0.93	0.93	15
4	1.00	1.00	1.00	16
5	1.00	1.00	1.00	4
accuracy			0.97	110
macro avg	0.97	0.97	0.97	110
weighted avg	0.97	0.97	0.97	110

Lote:

Datos de entrenamiento:

rendimiento en entrenamiento (multiclase):
Accuracy: 0.7812

matriz de confusion:
[[72 0 0 0 0 0]
[0 45 0 0 0 0]
[0 0 53 0 0 0]
[0 32 0 2 0 0]
[0 24 0 0 12 0]
[0 0 0 0 0 16]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	72
1	0.45	1.00	0.62	45
2	1.00	1.00	1.00	53
3	1.00	0.06	0.11	34
4	1.00	0.33	0.50	36
5	1.00	1.00	1.00	16
accuracy			0.78	256
macro avg	0.91	0.73	0.70	256
weighted avg	0.90	0.78	0.74	256

datos de prueba:

rendimiento en prueba (multiclase):
Accuracy: 0.8818

matriz de confusion:
[[40 0 0 0 0 0]
[0 15 0 1 0 0]
[0 0 19 0 0 0]
[0 1 0 14 0 0]
[0 5 0 6 5 0]
[0 0 0 0 0 4]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	40
1	0.71	0.94	0.81	16
2	1.00	1.00	1.00	19
3	0.67	0.93	0.78	15
4	1.00	0.31	0.48	16
5	1.00	1.00	1.00	4
accuracy			0.88	110
macro avg	0.90	0.86	0.84	110
weighted avg	0.91	0.88	0.87	110

Clasificador polinomial multivariable – multiclass - GDX – regresor softmax:

Utilizando datos de: dermatology.dat

Mini lote:

Datos de entrenamiento:

rendimiento en entrenamiento (multiclase):
Accuracy: 0.8594

matriz de confusion:
[[71 0 0 1 0 0]
[0 44 0 0 0 1]
[0 0 53 0 0 0]
[0 32 0 1 0 1]
[0 1 0 0 35 0]
[0 0 0 0 0 16]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.99	0.99	72
1	0.57	0.98	0.72	45
2	1.00	1.00	1.00	53
3	0.50	0.03	0.06	34
4	1.00	0.97	0.99	36
5	0.89	1.00	0.94	16
accuracy			0.86	256
macro avg	0.83	0.83	0.78	256
weighted avg	0.85	0.86	0.82	256

datos de prueba:

rendimiento en prueba (multiclase):
Accuracy: 0.7273

matriz de confusion:
[[27 0 0 0 0 13]
[0 16 0 0 0 0]
[0 0 19 0 0 0]
[0 15 0 0 0 0]
[0 1 0 0 14 1]
[0 0 0 0 0 4]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.68	0.81	40
1	0.50	1.00	0.67	16
2	1.00	1.00	1.00	19
3	0.00	0.00	0.00	15
4	1.00	0.88	0.93	16
5	0.22	1.00	0.36	4
accuracy			0.73	110
macro avg	0.62	0.76	0.63	110
weighted avg	0.76	0.73	0.71	110

Online:

Datos de entrenamiento:

rendimiento en entrenamiento (multiclase):
Accuracy: 0.9844

matriz de confusion:
[[72 0 0 0 0 0]
[0 42 0 3 0 0]
[0 0 53 0 0 0]
[0 1 0 33 0 0]
[0 0 0 0 36 0]
[0 0 0 0 0 16]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	72
1	0.98	0.93	0.95	45
2	1.00	1.00	1.00	53
3	0.92	0.97	0.94	34
4	1.00	1.00	1.00	36
5	1.00	1.00	1.00	16
accuracy			0.98	256
macro avg	0.98	0.98	0.98	256
weighted avg	0.98	0.98	0.98	256

datos de prueba:

rendimiento en prueba (multiclase):
Accuracy: 0.9636

matriz de confusion:
[[40 0 0 0 0 0]
[0 15 0 1 0 0]
[0 0 19 0 0 0]
[0 3 0 12 0 0]
[0 0 0 0 16 0]
[0 0 0 0 0 4]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	40
1	0.83	0.94	0.88	16
2	1.00	1.00	1.00	19
3	0.92	0.80	0.86	15
4	1.00	1.00	1.00	16
5	1.00	1.00	1.00	4
accuracy			0.96	110
macro avg	0.96	0.96	0.96	110
weighted avg	0.97	0.96	0.96	110

Lote:

Datos de entrenamiento:

rendimiento en entrenamiento (multiclase):
Accuracy: 0.8242

matriz de confusion:
[[65 7 0 0 0 0]
[0 44 0 0 0 1]
[0 0 53 0 0 0]
[0 31 0 2 0 1]
[0 5 0 0 31 0]
[0 0 0 0 0 16]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.90	0.95	72
1	0.51	0.98	0.67	45
2	1.00	1.00	1.00	53
3	1.00	0.06	0.11	34
4	1.00	0.86	0.93	36
5	0.89	1.00	0.94	16
accuracy			0.82	256
macro avg	0.90	0.80	0.77	256
weighted avg	0.91	0.82	0.79	256

datos de prueba:

rendimiento en prueba (multiclase):
Accuracy: 0.9636

matriz de confusion:
[[39 0 0 0 0 1]
[0 15 0 1 0 0]
[0 0 19 0 0 0]
[0 1 0 13 1 0]
[0 0 0 0 16 0]
[0 0 0 0 0 4]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.97	0.99	40
1	0.94	0.94	0.94	16
2	1.00	1.00	1.00	19
3	0.93	0.87	0.90	15
4	0.94	1.00	0.97	16
5	0.80	1.00	0.89	4
accuracy			0.96	110
macro avg	0.93	0.96	0.95	110
weighted avg	0.97	0.96	0.96	110

Clasificador polinomial multivariable – multiclass – AdamD – regresión logística:

Utilizando datos de: dermatology.dat

Mini lote:

Datos de entrenamiento:

datos de prueba:

Rendimiento en entrenamiento (multiclase - AdamD):
Accuracy: 0.9570

Matriz de confusión:

```
[[70 0 0 2 0 0]
 [ 0 39 0 5 0 1]
 [ 0 0 53 0 0 0]
 [ 0 2 0 32 0 0]
 [ 0 0 0 0 36 0]
 [ 0 0 0 0 1 15]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.97	0.99	72
1	0.95	0.87	0.91	45
2	1.00	1.00	1.00	53
3	0.82	0.94	0.88	34
4	0.97	1.00	0.99	36
5	0.94	0.94	0.94	16
accuracy			0.96	256
macro avg	0.95	0.95	0.95	256
weighted avg	0.96	0.96	0.96	256

Rendimiento en prueba (multiclase - AdamD):
Accuracy: 0.9636

Matriz de confusión:

```
[[40 0 0 0 0 0]
 [ 0 14 0 0 0 2]
 [ 0 0 19 0 0 0]
 [ 0 1 0 13 0 1]
 [ 0 0 0 0 16 0]
 [ 0 0 0 0 0 4]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	40
1	0.93	0.88	0.90	16
2	1.00	1.00	1.00	19
3	1.00	0.87	0.93	15
4	1.00	1.00	1.00	16
5	0.57	1.00	0.73	4
accuracy			0.96	110
macro avg	0.92	0.96	0.93	110
weighted avg	0.97	0.96	0.97	110

Online:

Datos de entrenamiento:

rendimiento en entrenamiento (multiclase - AdamD):
Accuracy: 0.9844

matriz de confusion:

```
[[71 0 0 1 0 0]
 [ 0 44 0 1 0 0]
 [ 0 0 52 1 0 0]
 [ 0 1 0 33 0 0]
 [ 0 0 0 0 36 0]
 [ 0 0 0 0 0 16]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.99	0.99	72
1	0.98	0.98	0.98	45
2	1.00	0.98	0.99	53
3	0.92	0.97	0.94	34
4	1.00	1.00	1.00	36
5	1.00	1.00	1.00	16
accuracy			0.98	256
macro avg	0.98	0.99	0.98	256
weighted avg	0.99	0.98	0.98	256

datos de prueba

Rendimiento en prueba (multiclase - AdamD):
Accuracy: 0.9636

matriz de confusion:

```
[[40 0 0 0 0 0]
 [ 0 15 0 1 0 0]
 [ 0 0 19 0 0 0]
 [ 0 2 0 13 0 0]
 [ 0 0 0 1 15 0]
 [ 0 0 0 0 0 4]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	40
1	0.88	0.94	0.91	16
2	1.00	1.00	1.00	19
3	0.87	0.87	0.87	15
4	1.00	0.94	0.97	16
5	1.00	1.00	1.00	4
accuracy			0.96	110
macro avg	0.96	0.96	0.96	110
weighted avg	0.96	0.96	0.96	110

Lote:

Datos de entrenamiento:

rendimiento en entrenamiento (multiclase - AdamD):
Accuracy: 0.9375

matriz de confusion:

```
[[69 1 0 0 0 2]
 [ 0 33 0 11 0 1]
 [ 0 0 53 0 0 0]
 [ 0 0 0 34 0 0]
 [ 0 0 0 1 35 0]
 [ 0 0 0 0 0 16]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.96	0.98	72
1	0.97	0.73	0.84	45
2	1.00	1.00	1.00	53
3	0.74	1.00	0.85	34
4	1.00	0.97	0.99	36
5	0.84	1.00	0.91	16
accuracy			0.94	256
macro avg	0.93	0.94	0.93	256
weighted avg	0.95	0.94	0.94	256

datos de prueba:

rendimiento en entrenamiento (multiclase - AdamD):
Accuracy: 0.9375

matriz de confusion:

```
[[69 1 0 0 0 2]
 [ 0 33 0 11 0 1]
 [ 0 0 53 0 0 0]
 [ 0 0 0 34 0 0]
 [ 0 0 0 1 35 0]
 [ 0 0 0 0 0 16]]
```

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.96	0.98	72
1	0.97	0.73	0.84	45
2	1.00	1.00	1.00	53
3	0.74	1.00	0.85	34
4	1.00	0.97	0.99	36
5	0.84	1.00	0.91	16
accuracy			0.94	256
macro avg	0.93	0.94	0.93	256
weighted avg	0.95	0.94	0.94	256

Clasificador polinomial multivariable – multiclass – AdamD – regresión SoftMax:

Utilizando datos de: dermatology.dat

Mini lotes:

Datos de entrenamiento:

rendimiento en entrenamiento (multiclase - AdamD):
Accuracy: 0.9609

matriz de confusion:
[[70 1 0 1 0 0]
[1 39 0 5 0 0]
[0 0 52 1 0 0]
[0 1 0 33 0 0]
[0 0 0 0 36 0]
[0 0 0 0 0 16]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	0.99	0.97	0.98	72
1	0.95	0.87	0.91	45
2	1.00	0.98	0.99	53
3	0.82	0.97	0.89	34
4	1.00	1.00	1.00	36
5	1.00	1.00	1.00	16
accuracy			0.96	256
macro avg	0.96	0.97	0.96	256
weighted avg	0.96	0.96	0.96	256

datos de prueba:

rendimiento en prueba (multiclase - AdamD):
Accuracy: 0.9727

matriz de confusion:
[[40 0 0 0 0 0]
[0 16 0 0 0 0]
[0 0 18 0 0 1]
[0 2 0 13 0 0]
[0 0 0 0 16 0]
[0 0 0 0 0 4]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	40
1	0.89	1.00	0.94	16
2	1.00	0.95	0.97	19
3	1.00	0.87	0.93	15
4	1.00	1.00	1.00	16
5	0.80	1.00	0.89	4
accuracy			0.97	110
macro avg	0.95	0.97	0.96	110
weighted avg	0.98	0.97	0.97	110

Online:

Datos de entrenamiento:

rendimiento en entrenamiento (multiclase - AdamD):
Accuracy: 0.9805

matriz de confusion:
[[71 1 0 0 0 0]
[0 44 0 1 0 0]
[0 0 53 0 0 0]
[0 3 0 31 0 0]
[0 0 0 0 36 0]
[0 0 0 0 0 16]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.99	0.99	72
1	0.92	0.98	0.95	45
2	1.00	1.00	1.00	53
3	0.97	0.91	0.94	34
4	1.00	1.00	1.00	36
5	1.00	1.00	1.00	16
accuracy			0.98	256
macro avg	0.98	0.98	0.98	256
weighted avg	0.98	0.98	0.98	256

datos de prueba:

rendimiento en prueba (multiclase - AdamD):
Accuracy: 0.9636

matriz de confusion:
[[40 0 0 0 0 0]
[0 15 0 1 0 0]
[0 0 19 0 0 0]
[0 3 0 12 0 0]
[0 0 0 0 16 0]
[0 0 0 0 0 4]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	40
1	0.83	0.94	0.88	16
2	1.00	1.00	1.00	19
3	0.92	0.80	0.86	15
4	1.00	1.00	1.00	16
5	1.00	1.00	1.00	4
accuracy			0.96	110
macro avg	0.96	0.96	0.96	110
weighted avg	0.97	0.96	0.96	110

Lote:

Datos de entrenamientos:

rendimiento en entrenamiento (multiclase - AdamD):
Accuracy: 0.9023

matriz de confusion:
[[68 3 0 0 0 1]
[0 41 0 3 0 1]
[0 0 53 0 0 0]
[0 2 0 32 0 0]
[0 0 0 0 35 1]
[0 0 0 5 9 2]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.94	0.97	72
1	0.89	0.91	0.90	45
2	1.00	1.00	1.00	53
3	0.80	0.94	0.86	34
4	0.80	0.97	0.88	36
5	0.40	0.12	0.19	16
accuracy			0.90	256
macro avg	0.81	0.82	0.80	256
weighted avg	0.89	0.90	0.89	256

datos de entrenamientos

rendimiento en prueba (multiclase - AdamD):
Accuracy: 0.9273

matriz de confusion:
[[39 0 0 0 0 1]
[0 15 0 1 0 0]
[0 0 19 0 0 0]
[0 1 0 13 0 1]
[0 0 0 0 16 0]
[0 0 0 1 3 0]]

reporte de clasificacion:

	precision	recall	f1-score	support
0	1.00	0.97	0.99	40
1	0.94	0.94	0.94	16
2	1.00	1.00	1.00	19
3	0.87	0.87	0.87	15
4	0.84	1.00	0.91	16
5	0.00	0.00	0.00	4
accuracy			0.93	110
macro avg	0.77	0.80	0.78	110
weighted avg	0.91	0.93	0.92	110

