

## Momento de Retroalimentación: Módulo 2

### Implementación de un modelo de aprendizaje máquina: Regresión logística

A continuación, se muestran los resultados del modelo de regresión logística implementado en el archivo *logistic\_regresion\_PabloSpinola.py*. Cabe mencionar que las explicaciones detalladas sobre el método, modelo, algoritmo y procedimiento se encuentran en dicho archivo. En el archivo presente se muestran los resultados de este modelo de aprendizaje máquina y una breve interpretación de los resultados arrojados por este.

```

----- Regresion logistica -----
-----
alpha: 0.1
train %: 60%
validation 20%:
test %: 20%
-----
X entrenamiento: [90, 89, 16, 87, 86, 42, 81, 83, 32, 40, 94, 72, 88, 92, 100, 24, 30, 73, 66, 58, 35, 22, 15, 69, 70, 74, 86, 62, 94, 77]
Y entrenamiento: [1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1]
-----
X validación: [10, 88, 82, 79, 61, 15, 96, 71, 13, 85]
Y validación: [0, 1, 1, 1, 0, 0, 1, 1, 0, 1]
-----
X test: [24, 49, 76, 84, 99, 2, 80, 67, 97, 99]
Y test: [0, 0, 1, 1, 1, 0, 1, 0, 1, 1]
-----
----- Datasets X escalados -----
-----
X entrenamiento: [0.25899999999999995, 0.24899999999999994, -0.48000000000000005, 0.22099999999999995, 0.21899999999999994, -0.22000000000000006, 0.16099999999999995, 0.18099999999999994, -0.32000000000000007, -0.24000000000000005, 0.29099999999999999, 0.07099999999999999, 0.23099999999999996, 0.27099999999999996, 0.35099999999999999, -0.40000000000000003, -0.34000000000000003, 0.08099999999999995, 0.010999999999999994, -0.06000000000000006, -0.29000000000000004, -0.42000000000000005, -0.49999999999999999, 0.08099999999999999, 0.05099999999999999, 0.02099999999999999, 0.00099999999999999, 0.21099999999999994, -0.020000000000000057, 0.29099999999999999, 0.12099999999999999]
X validación: [-0.52083333333333334, 0.29166666666666667, 0.22916666666666666, 0.19791666666666666, 0.010416666666666666, -0.46875, 0.375, 0.11458333333333333, -0.48958333333333333, 0.26041666666666667]
X test: [-0.4414141414141414, -0.18888888888888889, 0.08383838383838381, 0.16464646464646462, 0.31616161616161614, -0.6636363636363637, 0.12424242424242421, -0.0070707070707071, 0.29599999999999999, 0.31616161616161614]
-----
Pesos finales ajustados del modelo:
w0 = -3.061250604803801
w1 = 71.63624922381815
-----
Pérdida con data de validación: 0.0100
Pérdida con data de prueba: 0.0802
Precisión: 0.7142857142857143
Recall: 1.0
Puntuación F1: 0.8333333333333333

```

Figura 1

```

----- Regresion logistica -----
-----
alpha: 0.1
train %: 60%
validation 20%:
test %: 20%
-----
X entrenamiento: [90, 89, 16, 87, 86, 42, 81, 83, 32, 40, 94, 72, 88, 92, 100, 24, 30, 73, 66, 58, 35, 22, 15, 69, 70, 74, 86, 62, 94, 77]
Y entrenamiento: [1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1]
-----
X validación: [10, 88, 82, 79, 61, 15, 96, 71, 13, 85]
Y validación: [0, 1, 1, 1, 0, 0, 1, 1, 0, 1]
-----
X test: [24, 49, 76, 84, 99, 2, 80, 67, 97, 99]
Y test: [0, 0, 1, 1, 1, 0, 1, 0, 1, 1]
-----

```

Figura 1.1

En esta figura se pueden ver claramente los datos iniciales, incluyendo de los valores para el aprendizaje (alfa, porcentajes para train, validation y test sets), al igual que el dataset seccionado para el efectivo entrenamiento del modelo.

```

----- Datasets X escalados -----
-----
X entrenamiento: [0.25899999999999995, 0.24899999999999994, -0.48000000000000005, 0.22099999999999995, 0.21899999999999994, -0.22000000000000006, 0.16099999999999995, 0.18099999999999994, -0.32000000000000007, -0.24000000000000005, 0.29099999999999999, 0.07099999999999999, 0.23099999999999996, 0.27099999999999996, 0.35099999999999999, -0.40000000000000003, -0.34000000000000003, 0.08099999999999995, 0.010999999999999994, -0.06000000000000006, -0.29000000000000004, -0.42000000000000005, -0.49999999999999999, 0.08099999999999999, 0.05099999999999999, 0.02099999999999999, 0.00099999999999999, 0.21099999999999994, -0.020000000000000057, 0.29099999999999999, 0.12099999999999999]
X validación: [-0.52083333333333334, 0.29166666666666667, 0.22916666666666666, 0.19791666666666666, 0.010416666666666666, -0.46875, 0.375, 0.11458333333333333, -0.48958333333333333, 0.26041666666666667]
X test: [-0.4414141414141414, -0.18888888888888889, 0.08383838383838381, 0.16464646464646462, 0.31616161616161614, -0.6636363636363637, 0.12424242424242421, -0.0070707070707071, 0.29599999999999999, 0.31616161616161614]
-----

```

Figura 1.2

Aquí (Figura 1.2) se encuentran escaladas las variables independientes, con las cuales se espera que el modelo aprenda.

```
-----  
Pesos finales ajustados del modelo:  
w0 = -3.061258469489301  
w1 = 71.63624922301815  
  
Pérdida con data de validación: 0.0100  
Pérdida con data de prueba: 0.0082  
Precisión: 0.7142857142857143  
Recall: 1.0  
Puntuación F1: 0.8333333333333333
```

Figura 1.3

Finalmente, podemos ver la última impresión en consola del programa, donde podemos ver datos estadísticos concretos que nos ayudarán a evaluar el modelo y su precisión para predecir cualquier valor futuro. Este análisis se hará más adelante en el documento.

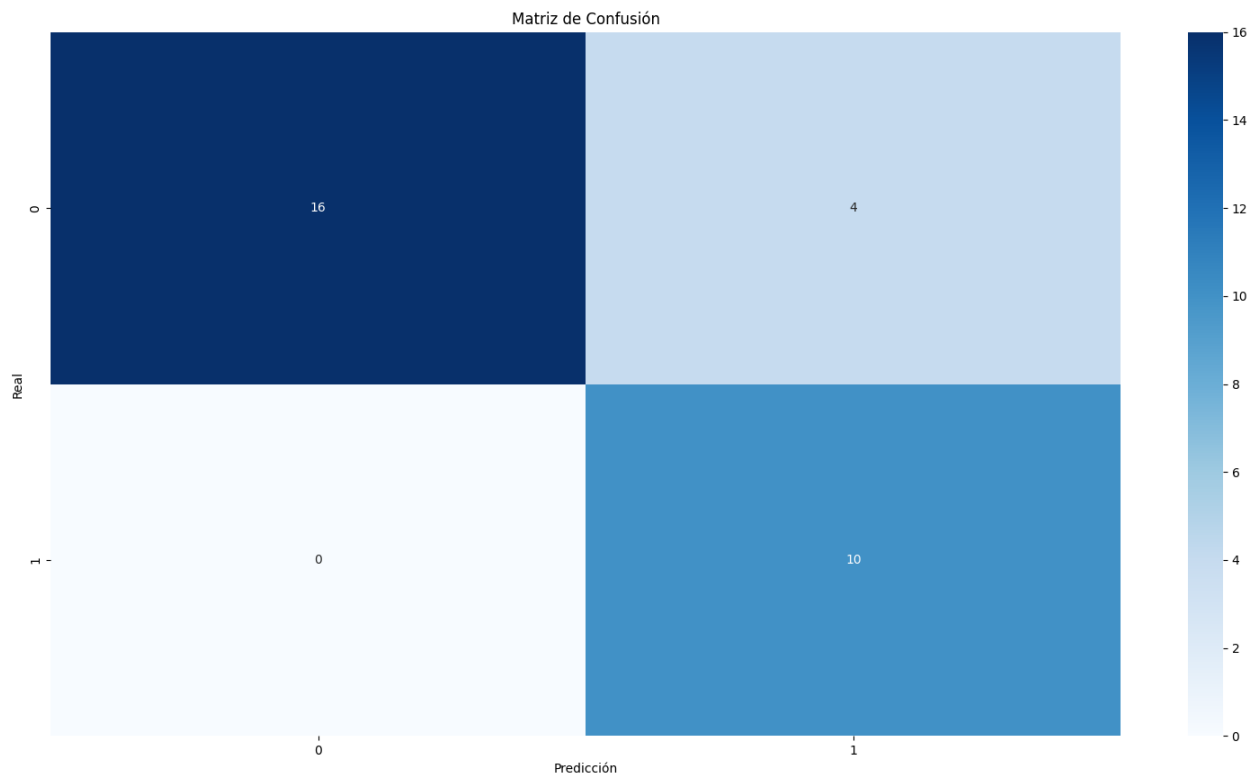


Figura 2

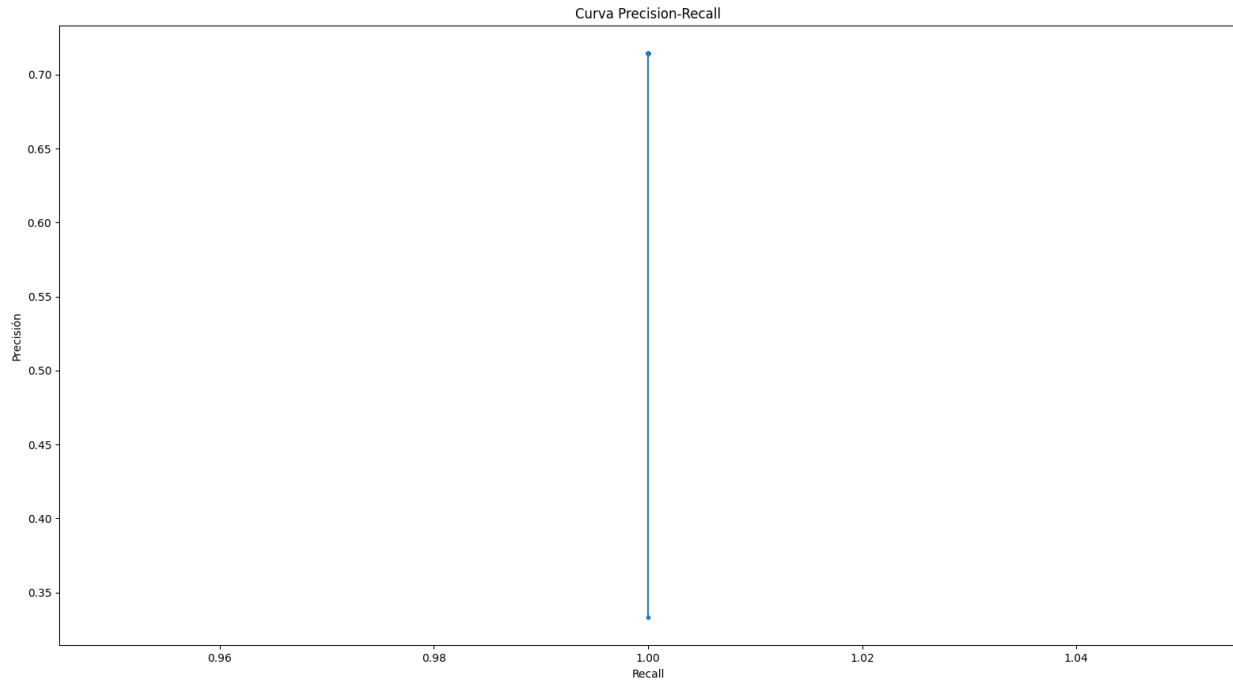


Figura 3

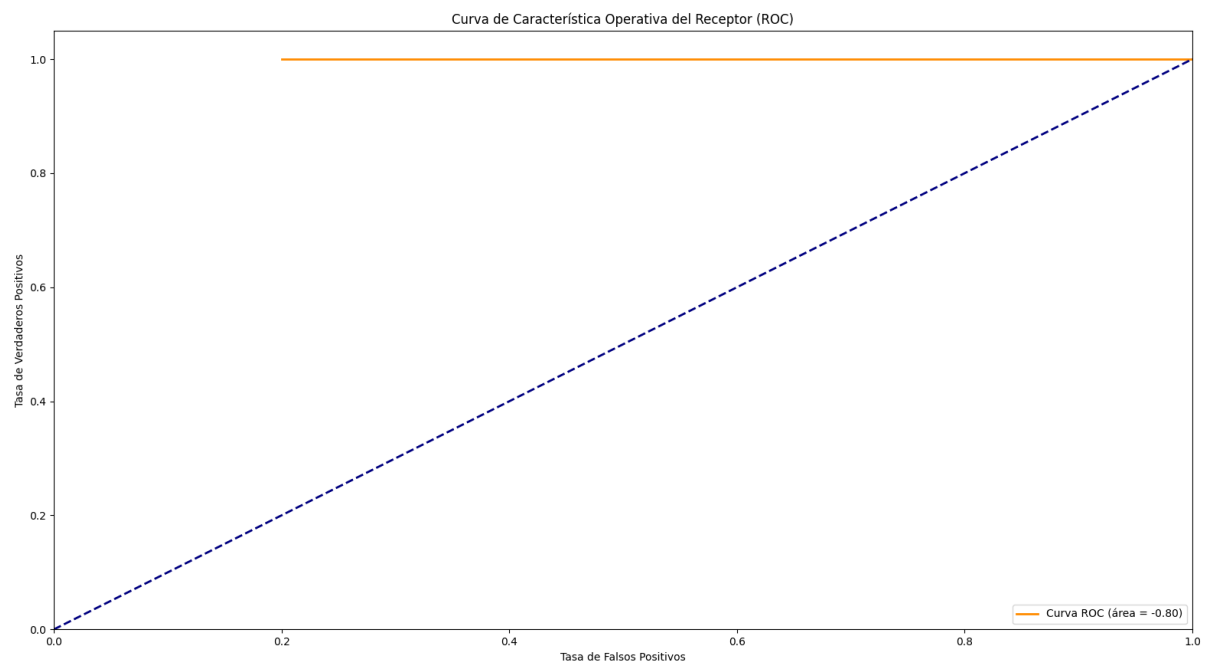


Figura 4

Una vez que hemos visualizado las métricas y los valores de evaluación, nos es posible medir y calcular la precisión de nuestro modelo, describiendo así si es que resultó eficiente al momento de aprender este patrón de clasificación de los datasets proporcionados.

Tras ver los resultados del modelo de regresión logística en la impresión de la Figura 1.3, podemos decir satisfactoriamente que se observa un ajuste robusto, con una pérdida baja tanto en los datos de validación (0.0100) como en los datos de prueba (0.0082) con una pérdida todavía menor, lo que nos indica claramente que el modelo tiene una buena capacidad de generalización. Además, tiene una puntuación F1 de 83.33% (0.8333), mostrando un buen equilibrio entre precisión y recall, es decir, el modelo puede identificar correctamente los positivos y al mismo tiempo mantiene un bajo número de errores en sus predicciones positivas, mismo dato que podemos ver plasmado en la matriz de confusión, con una alta densidad tanto de verdaderos positivos como de verdaderos negativos, y en la curva PR, con una aproximación de 1, 7.

Sin embargo, a pesar de que el recall del 100% nos dice que el modelo captura todos los casos positivos reales, la precisión del 71.43% nos indica que existe, aunque poca, cierta área de oportunidad para mejorar el modelo en dirección a la reducción de los falsos positivos. Aunque, al ver la curva ROC y observar que se acerca a la esquina superior izquierda de la gráfica, se presume un buen rendimiento, indicando que el modelo logra una alta tasa de verdaderos positivos y, con esto, una baja tasa de falsos positivos. El área bajo la curva termina siendo de alrededor de 0.8, dejando claro que el modelo es altamente capaz de distinguir entre clases, es decir, es un modelo muy bueno.

En resumen, el modelo muestra un excelente rendimiento en términos de detección de casos positivos, con una posible mejora en la precisión para reducir los falsos positivos. Estos resultados destacan la efectividad del modelo en términos generales; con una pérdida baja, buena capacidad de generalización, identificación de verdaderos positivos y negativos. También se sugiere la necesidad de investigar y ajustar el equilibrio entre precisión y recall para optimizar el rendimiento general y así avanzar en dirección de una precisión más exacta y clara mejora del modelo.