

Regresión Logística IA ARLG

Abraham López

Marzo 2025

1 Introducción

1.1 ¿Qué es la regresión logística?

La regresión logística es un tipo de modelo estadístico (también conocido como modelo logit) que se utiliza a menudo para la clasificación y el análisis predictivo. A diferencia de la regresión lineal, que se emplea para predecir valores continuos, la regresión logística se utiliza para problemas de clasificación, donde la variable dependiente es categórica (por ejemplo, 0 o 1). Este modelo estima la probabilidad de que ocurra un evento en función de un conjunto de variables independientes.

Dado que el resultado es una probabilidad, la variable dependiente está limitada entre 0 y 1. En la regresión logística, se aplica una transformación logit a las probabilidades, es decir, la probabilidad de éxito dividida por la probabilidad de fracaso. Esto también se conoce comúnmente como probabilidades logarítmicas, o el logaritmo natural de probabilidades, y esta función logística se representa mediante las siguientes fórmulas:

$$\text{Logit}(p_i) = \ln \left(\frac{p_i}{1 - p_i} \right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$

Donde:

- p_i : Es la probabilidad de que la variable dependiente sea 1 (éxito).
- $\ln \left(\frac{p_i}{1 - p_i} \right)$: Es el logaritmo natural de las probabilidades (logit).
- β_0 : Es el término independiente (intercept).
- $\beta_1, \beta_2, \dots, \beta_k$: Son los coeficientes de las variables independientes X_1, X_2, \dots, X_k .

El parámetro beta, o coeficiente, en este modelo se estima comúnmente mediante la estimación de máxima verosimilitud (MLE). Este método prueba diferentes valores de beta a través de múltiples iteraciones para optimizar el mejor ajuste de las probabilidades de registro. Todas estas iteraciones producen la función de verosimilitud logarítmica, y la regresión logística busca maximizar esta función para encontrar la mejor estimación de parámetros.

Una vez que se encuentra el coeficiente óptimo (o los coeficientes si hay más de una variable independiente), las probabilidades condicionales para cada observación se pueden calcular, registrar y sumar para obtener una probabilidad predicha. Para la clasificación binaria, una probabilidad inferior a 0.5 predecirá 0, mientras que una probabilidad superior a 0.5 predecirá 1.

1.2 Regresión logística y aprendizaje automático

Dentro del aprendizaje automático, la regresión logística pertenece a la familia de modelos de aprendizaje automático supervisado. También se considera un modelo discriminativo, lo que significa que intenta distinguir entre clases (o categorías). A diferencia de un algoritmo generativo, como bayesiano ingenuo, no puede, ya que el nombre implica, generar información, como una imagen, de la clase que intenta predecir (por ejemplo, una imagen de un gato).

Anteriormente, mencionamos cómo la regresión logística maximiza la función de verosimilitud logarítmica para determinar los coeficientes beta del modelo. Esto cambia ligeramente en el contexto del aprendizaje automático. Dentro del aprendizaje automático, se utilizó el logaritmo de probabilidad negativo como función de pérdida, utilizando el proceso de descenso de gradiente para encontrar el máximo global. Esta es solo otra forma de llegar a las mismas estimaciones discutidas anteriormente.

La regresión logística también puede ser propensa al sobreajuste, especialmente cuando hay una gran cantidad de variables predictoras dentro del modelo. La regularización se utiliza normalmente para penalizar parámetros con coeficientes grandes cuando el modelo adolece de una alta dimensionalidad.

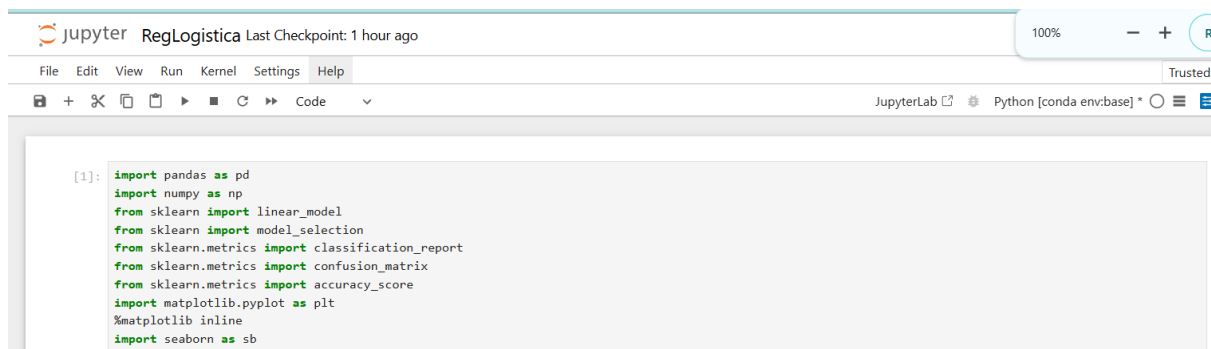
1.3 Regresión lineal frente a regresión logística

Al igual que la regresión lineal, la regresión logística también se utiliza para estimar la relación entre una variable dependiente y una o más variables independientes, pero se utiliza para hacer una predicción sobre una variable categórica frente a una continua. Una variable categórica puede ser verdadera o falsa, sí o no, 1 o 0, etc. La unidad de medida también difiere de la regresión lineal en que produce una probabilidad, pero la función logit transforma la curva S en línea recta.

Si bien ambos modelos se utilizan en el análisis de regresión para hacer predicciones sobre resultados futuros, la regresión lineal suele ser más fácil de entender. La regresión lineal tampoco requiere un tamaño de muestra tan grande como la regresión logística necesita una muestra adecuada para representar valores en todas las categorías de respuesta. Sin una muestra más grande y representativa, es posible que el modelo no tenga suficiente poder estadístico para detectar un efecto significativo.

2 Metodología

Se importan las librerías necesarias para el análisis, como 'pandas', 'numpy', 'sklearn', 'matplotlib' y 'seaborn'.



```
[1]: import pandas as pd
import numpy as np
from sklearn import linear_model
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sb
```

Se carga el archivo CSV en un DataFrame y se exploran los datos utilizando:

- 'head()': Para ver las primeras filas.
- 'describe()': Para obtener estadísticas descriptivas.
- 'groupby('clase').size()': Para contar el número de observaciones por clase.

```
[3]: dataframe = pd.read_csv(r"usuarios_win_mac_lin.csv")
dataframe.head()
```

```
[3]:
```

	duracion	paginas	acciones	valor	clase
0	7.0	2	4	8	2
1	21.0	2	6	6	2
2	57.0	2	4	4	2
3	101.0	3	6	12	2
4	109.0	2	6	12	2

```
[5]: dataframe.describe()
```

```
[5]:
```

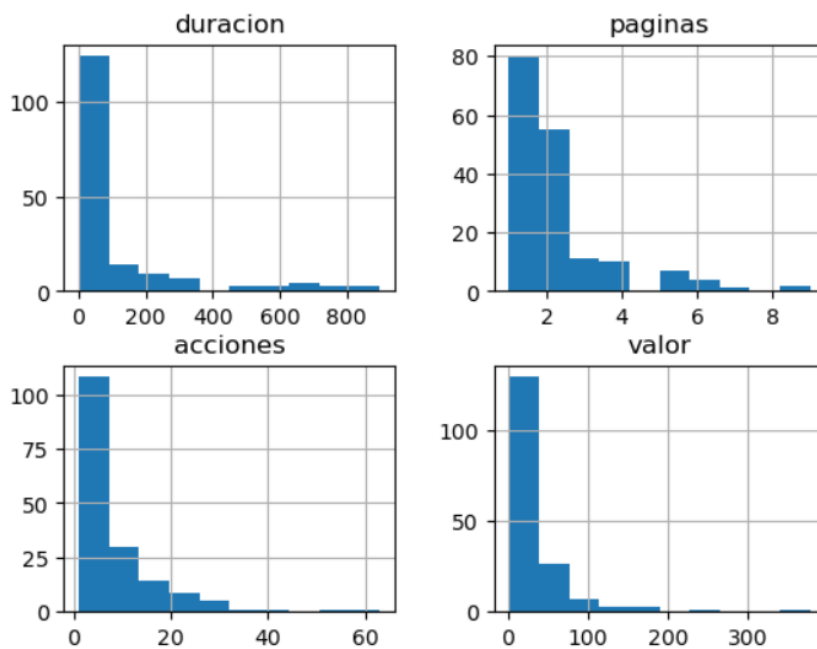
	duracion	paginas	acciones	valor	clase
count	170.000000	170.000000	170.000000	170.000000	170.000000
mean	111.075729	2.041176	8.723529	32.676471	0.752941
std	202.453200	1.500911	9.136054	44.751993	0.841327
min	1.000000	1.000000	1.000000	1.000000	0.000000
25%	11.000000	1.000000	3.000000	8.000000	0.000000
50%	13.000000	2.000000	6.000000	20.000000	0.000000
75%	108.000000	2.000000	10.000000	36.000000	2.000000
max	898.000000	9.000000	63.000000	378.000000	2.000000

```
[7]: print(dataframe.groupby('clase').size())
```

```
clase
0      86
1      40
2      44
dtype: int64
```

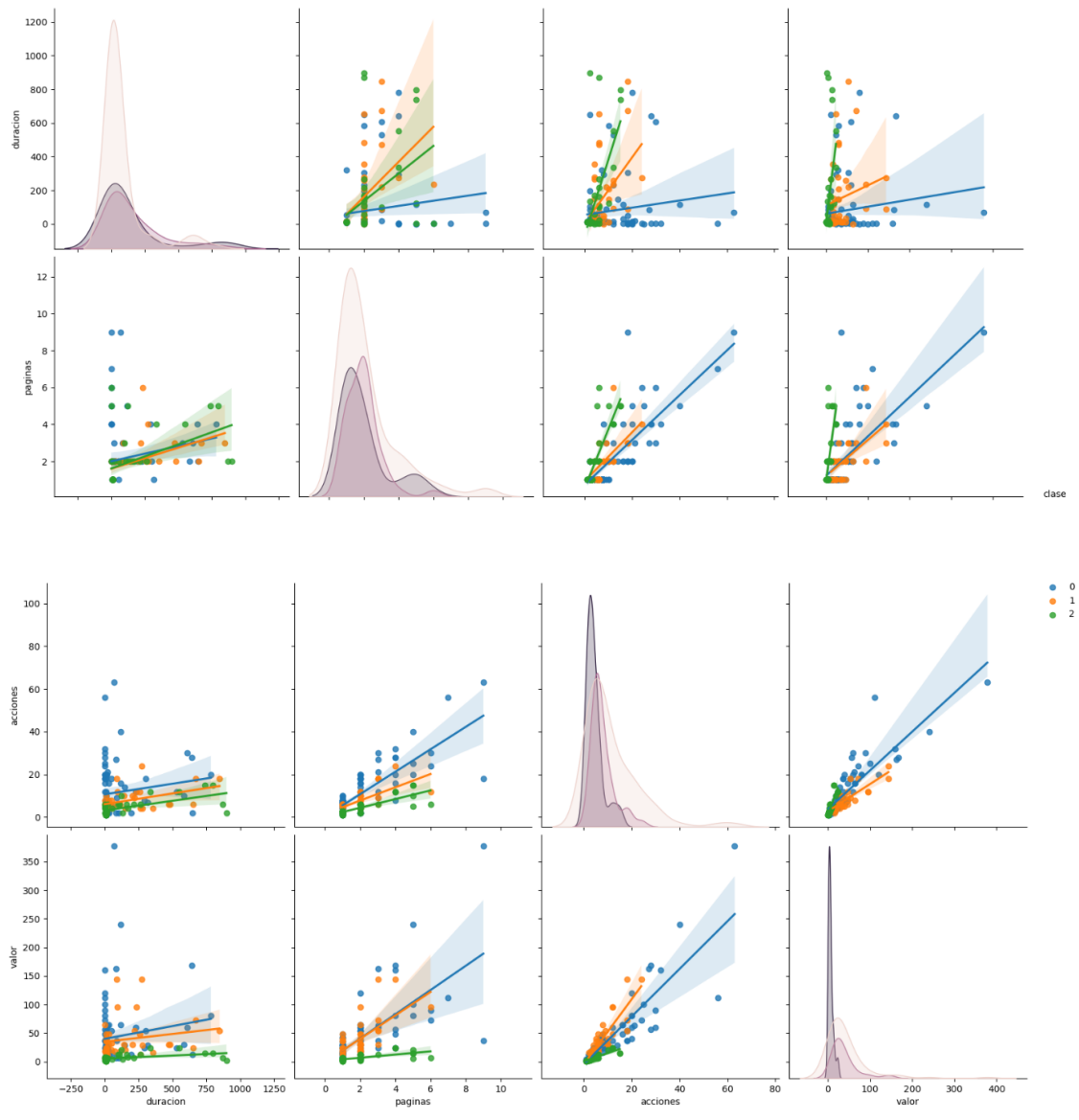
Se generan histogramas de las variables independientes ('duracion', 'paginas', 'acciones', 'valor') excluyendo la variable dependiente ('clase').

```
[9]: dataframe.drop(['clase'], axis=1).hist()
plt.show()
```



Se crea un gráfico de pares ('pairplot') para visualizar las relaciones entre las variables independientes, coloreando los puntos según la clase ('hue='clase').

```
[11]: sb.pairplot(dataframe.dropna(), hue='clase', height=4, vars=["duracion", "paginas", "acciones", "valor"], kind='reg')
plt.show()
```



Se separan las variables independientes ('X') y la variable dependiente ('y'):

- 'X' contiene las columnas 'duracion', 'paginas', 'acciones' y 'valor'.
- 'y' contiene la columna 'clase'.

Se verifica la forma de 'X' para asegurarse de que los datos estén correctamente preparados.

Se crea un modelo de regresión logística ('LogisticRegression') con un máximo de 1,000 iteraciones para asegurar la convergencia. Luego, se entrena el modelo utilizando las variables independientes ('X') y la variable dependiente ('y'). Se realizan predicciones ('predictions') sobre el conjunto de datos original y se imprime una muestra de las predicciones. Finalmente, se calcula la precisión del modelo ('model.score(X, y)').

```
[13]: X = np.array(dataframe.drop(['clase'], axis=1))
      y = np.array(dataframe['clase'])
      X.shape

[13]: (170, 4)

[19]: model = linear_model.LogisticRegression(max_iter=1000)
      model.fit(X, y)

[19]: LogisticRegression
      LogisticRegression(max_iter=1000)

[21]: predictions = model.predict(X)
      print(predictions[:5])

[21]: [2 2 2 2 2]

[23]: model.score(X, y)

[23]: 0.7764705882352941
```

Se divide el conjunto de datos en entrenamiento (80%) y validación (20%). Se utiliza validación cruzada con 10 particiones ('KFold') para evaluar el modelo y se imprime el resultado de la validación. Además, se realizan predicciones sobre el conjunto de validación, se calcula e imprime la precisión del modelo, se genera e imprime la matriz de confusión y, por último, se imprime un informe de clasificación.

```
[25]: validation_size = 0.20
      seed=7
      X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, y, test_size=validation_size, random_state=seed)
      name='Logistic Regression'
      kfold = model_selection.KFold(n_splits=10, shuffle=True, random_state=seed)
      cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy')
      msg = "%s: %f(%f)" % (name, cv_results.mean(), cv_results.std())
      print(msg)

      Logistic Regression: 0.720330(0.151123)

[27]: predictions = model.predict(X_validation)
      print(accuracy_score(Y_validation, predictions))

[27]: 0.8529411764705882

[29]: print(confusion_matrix(Y_validation, predictions))

[[16  0  2]
 [ 3  3  0]
 [ 0  0 10]]

[31]: print(classification_report(Y_validation, predictions))

              precision    recall  f1-score   support

     0       0.84         0.89         0.86         18
     1       1.00         0.50         0.67          6
     2       0.83         1.00         0.91         10

   accuracy                   0.85         34
  macro avg              0.89         0.80         0.81         34
 weighted avg              0.87         0.85         0.84         34
```

3 Resultados

Tras realizar el análisis de regresión logística, se obtuvieron los siguientes resultados.

- La **precisión del modelo en el conjunto de entrenamiento** es:

$$\text{model.score}(X, y) = 0.7764$$

Esto indica que el modelo clasifica correctamente el 77.64% de las observaciones en el conjunto de entrenamiento. Este valor es un indicador inicial de que el modelo tiene un desempeño razonable, pero debe ser validado con datos no vistos (conjunto de validación).

- El resultado de la **validación cruzada** con 10 particiones ('KFold') es:

$$\text{LogisticRegression} = 0.720330 \quad (\text{Desviación estándar} = 0.151123)$$

La precisión media del modelo durante la validación cruzada es del **72.03%**, con una desviación estándar de 0.151123. Esto sugiere que el modelo tiene un desempeño relativamente estable en diferentes particiones de los datos, aunque la precisión es menor que en el conjunto de entrenamiento, lo cual es normal debido a la naturaleza de la validación cruzada.

- La **precisión del modelo en el conjunto de validación** es:

$$\text{accuracy_score}(Y_validation, \text{predictions}) = 0.852941$$

Este valor indica que el modelo clasifica correctamente el 85.29% de las observaciones en el conjunto de validación. Este es un resultado alentador, ya que muestra que el modelo generaliza bien a datos no vistos y tiene un buen desempeño predictivo.

4 Conclusión

El análisis de regresión logística que se llevó a cabo nos permitió predecir la clase de usuarios (clase) basándonos en variables como la duración de la sesión (duracion), el número de páginas visitadas (paginas), las acciones realizadas (acciones) y el valor generado (valor). Los resultados fueron bastante alentadores, mostrando que el modelo tiene un buen desempeño con una precisión del 85.29% en el conjunto de validación. Esto sugiere que el modelo puede generalizar bien a datos que no ha visto antes, además, la precisión media en la validación cruzada fue del 72.03%, lo que refuerza la idea de que el modelo es robusto y consistente en diferentes particiones de los datos.

La regresión logística es una herramienta muy útil para abordar problemas de clasificación pues no solo predice la clase a la que pertenece un usuario, sino que también estima la probabilidad de que pertenezca a cada clase, esto resulta especialmente valioso en situaciones donde queremos entender cómo ciertas variables afectan la clasificación. En este caso, pudimos observar cómo la duración, el número de páginas, las acciones y el valor influyen en la probabilidad de que un usuario pertenezca a una clase específica, también de su facilidad de implementación y su capacidad de interpretación la hacen una opción muy atractiva para el análisis de datos.