



Tecnológico de Monterrey

Instituto Tecnológico de Estudios Superiores de Monterrey

Analítica de datos y herramientas de inteligencia artificial II

Actividad 7: Regresión Logística

Profesor:
Alfredo García Suárez

Equipo 4:

María Fernanda Ledesma Martínez	A01734203
Karla Yamila Pagés Mejía	A01733409
César Ricardo Gastelú Parra	A01735328
Agustín Ibarra Sota	A01552618

Fecha de entrega: 18/10/2023

Regresión logística

Esta actividad tiene como objetivo analizar las variables de la base de datos 'Training Data Complete' a través de la correlación logística de cada una, y crear distintos modelos de regresión logística que ayuden a describir de mejor manera los datos.

Para ello, se trabajó con la base de datos 'BD Socio formador (Training Data Complete).csv', la cual contiene 13 columnas y 252,000 registros. Y cada modelo fue evaluado con los coeficientes de precisión, exactitud, sensibilidad y el puntaje de F1.

Preprocesamiento de los datos

Como primer paso para trabajar con la actividad se realizó la limpieza de la base de datos, quitando aquellos valores nulos y los outliers encontrados.

Una vez teniendo la base limpia, se procedió a realizar el análisis de 10 distintos modelos tomando como variables dependientes columnas que tuvieran únicamente dos clases, es decir, que fueran dicotómicas.

La base de datos original contiene únicamente tres columnas que son dicotómicas, sin embargo, con la finalidad de realizar mayores experimentos, se crearon distintos filtros convirtiendo la columna únicamente contenida con dos clases. Ya que el dataframe tenía desde datos categóricos hasta numéricos con diferentes tipos de escalas, con variables que contienen años o cantidad de dinero. Los filtros para cada análisis se describen a continuación:

1. Caso 1 Filtro:

Para este primer caso, se tomó la base de datos y se hizo uso de la columna de 'STATE'; de la cual se realizó previamente un análisis del comportamiento, en donde, se obtuvo que los estados con mayor frecuencia dentro de los datos son 'Uttar_Pradesh' y 'Maharashtra'. De esta manera se obtuvo un nuevo data frame con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
df1 = df.copy()  
#Convierto una variable a dicotómica  
filtro = df1[(df1['STATE'] == 'Uttar_Pradesh') | (df1['STATE'] == 'Maharashtra')]
```

Imagen 1. Filtro caso 1

En este caso, la columna de 'STATE' pasó a ser una variable dicotómica con dos clases, la cuales son:

- Clase 1 - 'Uttar_Paradesh'
- Clase 2: 'Maharashtra'

2. Caso 2 Filtro:

Se hizo uso de la columna de 'CURRENT_HOUSE_YRS', de la cual se realizó previamente un análisis del comportamiento, en donde, se obtuvo que los datos con una mayor frecuencia dentro de la base resultaron ser cuando los usuarios vivieron en ese domicilio 13 y 10 años. De esta manera se obtuvo un nuevo data frame con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
1 #Convierto una variable a dicotómica
2 filtro = df[(df['CURRENT_HOUSE_YRS'] == 13) | (df['CURRENT_HOUSE_YRS'] == 10)]
```

Imagen 2. Filtro caso 2

En este caso, la columna de 'CURRENT_HOUSE_YRS' resultó en ser una variable dicotómica con dos clases, la cuales son:

- Clase 1 - 13
- Clase 2: 10

3. Caso 3 Filtro:

Para este filtro se hizo uso de la columna de 'CURRENT_JOB_YRS', de la cual se realizó previamente un análisis del comportamiento; en donde, se obtuvo que los datos con una mayor frecuencia dentro de la base resultaron ser cuando los usuarios trabajaron en su puesto actual 5 y 6 años. De esta manera se obtuvo un nuevo data frame con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
#Convierto una variable a dicotómica
filtro = df[(df['CURRENT_JOB_YRS'] == 5) | (df['CURRENT_JOB_YRS'] == 6)]
```

Imagen 3. Filtro caso 3

En este caso, la columna de 'CURRENT_JOB_YRS' resultó en ser una variable dicotómica con dos clases, la cuales son:

- Clase 1: 5
- Clase 2: 6

4. Caso 4 Filtro:

Se tomó la columna 'Profession', de la cual se realizó previamente un análisis del comportamiento de esta variable; en donde, se obtuvo que los datos con una mayor frecuencia dentro de la base fueron aquellos usuarios que su profesión era 'Physician' y 'Statistician'. De esta manera se obtuvo un nuevo data frame con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
#Convierto una variable a dicotómica
filtro = df[(df['Profession'] == 'Physician') | (df['Profession'] == 'Statistician')]
```

Imagen 4. Filtro caso 4

En este caso, la columna de 'Profession' resultó en ser una variable dicotómica con dos clases, la cuales son:

- Clase 1 - 'Physician'
- Clase 2: 'Statistician'

5. Caso 5 Filtro:

Se ocupó la columna 'CITY', de la cual se realizó previamente un análisis del comportamiento de esta variable; en donde, se obtuvo que los estados con mayor frecuencia dentro de los datos son 'Vijayanagaram' y 'Bhopal'. De esta manera se obtuvo un nuevo data frame con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
#Convierto una variable a dicotómica
filtro = df[(df['CITY'] == 'Vijayanagaram') | (df['CITY'] == 'Bhopal')]
```

Imagen 5. Filtro caso 5

En este caso, la columna de 'Profession' resultó en ser una variable dicotómica con dos clases, la cuales son:

- Clase 1 - 'Vijayanagaram'
- Clase 2: 'Bhopal'

6. Caso 6 Filtro:

Ahora bien, para este caso, tomamos la base de datos e hicimos uso de la columna de 'Profession', de la cual se realizó previamente un análisis del comportamiento de esta variable, en donde, se obtuvo que los datos con una mayor frecuencia dentro de la base fueron aquellos usuarios que su profesión era 'Web_designer' y 'Psychologist'. De esta manera se obtuvo un nuevo data frame con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
#Convierto una variable a dicotómica  
filtro = df[(df['Profession'] == 'Web_designer') | (df['Profession'] == 'Psychologist')]
```

Imagen 6. Filtro caso 6

En este caso, la columna de 'Profession' resultó en ser una variable dicotómica con dos clases, la cuales son:

Clase 1 - 'Web_designer'

Clase 2: 'Psychologist'

7. Caso 7 Filtro:

Ahora bien, para este caso, tomamos la base de datos e hicimos uso de la columna de 'CURRENT_HOUSE_YRS', de la cual se realizó previamente un análisis del comportamiento de esta variable, en donde, se obtuvo que los datos con una mayor frecuencia dentro de la base resultaron ser cuando los usuarios vivieron en ese domicilio 11 y 12 años, estos datos representan los terceros y cuartos datos con mayor frecuencia. De esta manera se obtuvo un nuevo data frame con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
#Convierto una variable a dicotómica  
filtro = df[(df['CURRENT_HOUSE_YRS'] == 11) | (df['CURRENT_HOUSE_YRS'] == 12)]
```

Imagen 7. Filtro caso 7

En este caso, la columna de 'CURRENT_HOUSE_YRS' resultó en ser una variable dicotómica con dos clases, la cuales son:

Clase 1 - 11

Clase 2 - 12

8. Caso 8 Filtro:

Ahora bien, para este caso, tomamos la base de datos e hicimos uso de la columna de ‘CURRENT_JOB_YRS’, de la cual se realizó previamente un análisis del comportamiento de esta variable, en donde, se obtuvo que los datos con una mayor frecuencia dentro de la base resultaron ser cuando los usuarios trabajaron en su puesto actual 3 y 4 años, estos datos representan los terceros y cuartos datos con mayor frecuencia. De esta manera se obtuvo un nuevo data frame con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
#Convierto una variable a dicotómica
filtro = df[(df['CURRENT_JOB_YRS'] == 3) | (df['CURRENT_JOB_YRS'] == 4)]
```

Imagen 8. Filtro caso 8

En este caso, la columna de ‘CURRENT_JOB_YRS’ resultó en ser una variable dicotómica con dos clases, la cuales son:

Clase 1 - 3

Clase 2 - 4

9. Caso 9 Filtro:

Ahora bien, para este caso, tomamos la base de datos e hicimos uso de la columna de ‘Experience_Range’, de la cual se realizó previamente un análisis del comportamiento de esta variable, en donde, se obtuvo que los datos con una mayor frecuencia dentro de la base fueron aquellos usuarios que su llevaban una experiencia de alrededor 20 y 16 años. De esta manera se obtuvo un nuevo data frame con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
1 #se definen los rangos de edad en los cuáles se va a comparar cada registro
2 age_bins = [-1, 1, 4, 8, 12, 16, 20]
3
4 #se definen las etiquetas para cada rango de edad
5 age_label = [0, 4, 8, 12, 16, 20]
6
7 #el código asigna la edad del registro con el rango que corresponde de los definidos anteriormente, y dependiendo del correspondiente,
8 #asigna la etiqueta que corresponde al rango
9 df1['Experience_Range'] = pd.cut(df1['Experience'], age_bins, labels = age_label)
10 df1
11
12 #Convierto una variable a dicotómica
13 filtro = df1[(df1['Experience_Range'] == 20) | (df1['Experience_Range'] == 16)]
```

Imagen 9. Filtro caso 9

En este caso, la columna de ‘Experience_Range’ resultó en ser una variable dicotómica con dos clases, la cuales son:

Clase 1 - 20

Clase 2 - 16

10. Caso 10 Filtro:

Ahora bien, para este caso, tomamos la base de datos e hicimos uso de la columna de 'CITY', de la cual se realizó previamente un análisis del comportamiento de esta variable, en donde, se obtuvo que los estados con mayor frecuencia dentro de los datos son 'Saharsa[29]' y 'Indore'. De esta manera se obtuvo un nuevo data frame con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
1 #Convierto una variable a dicotómica
2 filtro = df1[(df1['CITY'] == 'Saharsa[29]') | (df1['CITY'] == 'Indore')]
```

Imagen 10. Filtro caso 10

En este caso, la columna de 'CITY' resultó en ser una variable dicotómica con dos clases, la cuales son:

Clase 1 - 'Saharsa[29]'

Clase 2 - 'Indore'

11. Caso 11 Filtro (Prueba):

Ahora bien, adicionalmente se hizo un caso en donde tomamos la base de datos e hicimos uso de la columna de 'Age', de la cual se obtuvo límite inferior y superior del valor de la variable y tomando en cuenta el promedio de la columna se dividieron las instancias, del promedio al límite superior es una instancia llamada 'Mayores' y del promedio al límite inferior está la clase 'Menores'. De esta manera se creó una nueva columna con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
import math
intervalos=[-math.inf, 21, math.inf]
categorias=['Menores', 'Mayores']
df['Age_intervalos']=pd.cut(x=df['Age'], bins=intervalos, labels=categorias)
df.head()
```

Imagen 11. Filtro caso prueba 11

12. Caso 12 Filtro (Prueba):

adicionalmente se hizo un segundo caso en donde tomamos la base de datos e hicimos uso de la columna de 'CURRENT_HOUSE_YRS', de la cual se obtuvo límite inferior y superior del valor de la variable y tomando en cuenta el promedio de la columna se dividieron las instancias, del promedio al límite superior es una instancia llamada 'Altos' y del promedio al límite inferior está la clase 'Bajos'. De esta manera se creó una nueva columna con este filtro, el cual se utilizó para realizar el modelo de regresión logística.

```
import math
intervalos = [-math.inf, 12, math.inf]
categorias = ['Altos', 'Bajos']
df['CURRENT_HOUSE_Inter'] = pd.cut(x = df['CURRENT_HOUSE_YRS'], bins = intervalos, labels = categorias)
df.head()
```

Imagen 12. Filtro caso prueba 12

A continuación, se presenta la tabla con los cálculos de precisión, exactitud, sensibilidad y puntaje F1. También, se presenta cada una de las métricas y cómo se calculan:

Precisión (Precision):

La precisión mide la proporción de instancias clasificadas como positivas que son verdaderamente positivas.

Se calcula como el número de verdaderos positivos (TP) dividido por la suma de verdaderos positivos y falsos positivos (FP).

$$precision = \frac{TP}{TP + FP}$$

Imagen 13. Fórmula para precisión

Exactitud (Accuracy):

La exactitud mide la proporción de todas las predicciones correctas en relación con el número total de instancias.

Se calcula como la suma de verdaderos positivos (TP) y verdaderos negativos (TN) dividida por el número total de instancias.

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

Imagen 14. Fórmula para exactitud

Recall (Recall o Sensibilidad):

El recall mide la proporción de instancias positivas que el modelo ha identificado correctamente en comparación con todas las instancias positivas reales.

Se calcula como el número de verdaderos positivos (TP) dividido por la suma de verdaderos positivos y falsos negativos (FN).

$$recall = \frac{TP}{TP + FN}$$

Imagen 15. Fórmula para sensibilidad

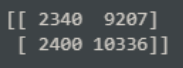
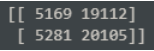
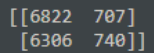
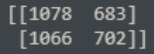
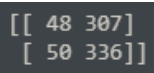
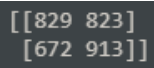
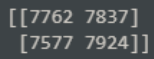
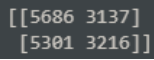
Valor F1 (F1-Score):

El valor F1 es una métrica que combina precisión y recall en una sola medida. Es útil cuando se desea encontrar un equilibrio entre ambas métricas.

Se calcula utilizando la siguiente fórmula:

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Imagen 16. Fórmula para F1

Clase	Var dep.	Vars Indep.	Matriz confusión	Precisión	Exactitud	Sensibilidad	Puntaje F1
Uttar_Pradesh	STATE	'Income', 'Age', 'CURRENT_HOUSE_YRS'		0.53	0.52	0.81	0.64
13	CURRENT_HOUSE_YRS	'Income', 'Age'		0.51	0.51	0.79	0.62
5	CURRENT_JOB_YRS	'Income', 'Age'		0.52	0.52	0.91	0.66
Physician	Profession	'Income', 'Age', 'Experience',		0.50	0.50	0.61	0.55
Vijayanagaram	CITY	'Income', 'Age', 'Risk_Flag'		0.52	0.52	0.87	0.65
Web_designer	Profession	'Income', 'Age', 'Experience',		0.53	0.54	0.58	0.55
12	CURRENT_HOUSE_YRS	'Income', 'Age', 'Risk_Flag'		0.50	0.50	0.51	0.51
3	CURRENT_JOB_YRS	'Income', 'Age', 'Experience',		0.52	0.51	0.64	0.57

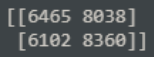
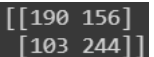
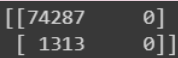
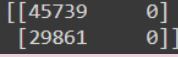
20	Experi nce_Ra nge	'Income', 'Age', 'CURRENT _JOB_YRS'		0.51	0.51	0.58	0.54
Saharsa [29]	CITY	'Income', 'Age', 'Risk_Flag'		0.61	0.61	0.67	0.64
Menore	Age	'Income', 'Experience'		0	0.98	0	0
Bajos	CURR ENT_H OUSE_ YRS	'Income', 'Experience', 'Risk_Flag'		0	0.65	0	0

Tabla 1. Resumen de los modelos

Conclusiones

Los resultados presentados anteriormente en las tablas nos muestran diferentes coeficientes en las métricas que son interesantes para ser analizados y evaluar el por qué de esos valores.

- Para las métricas de precisión y exactitud, la ecuación logística tenía como variable dependiente “CITY”, utilizando como variables independientes “Income”, “Age” y “Risk_Flag”. Los filtros para las clases que se utilizaron fueron los siguientes:

```
1 #Convierto una variable a dicotómica
2 filtro = df1[(df1['CITY'] == 'Saharsa[29]') | (df1['CITY'] == 'Indore')]
```

- Para las métricas de sensibilidad y el puntaje F1, la ecuación logística tenía como variable dependiente “CURRENT_JOB_YRS”, utilizando como variables independientes “Income” y “Age”. Los filtros para las clases que se utilizaron fueron los siguientes:

```
#Convierto una variable a dicotómica
filtro = df[(df['CURRENT_JOB_YRS'] == 5) | (df['CURRENT_JOB_YRS'] == 6)]
```

Por lo que vale la pena analizar el porqué estos dos modelos almacenaron los puntajes más altos en las métricas, en lugar de ser distribuidos entre todos los demás.