

Regresión Logística



A 27 de abril de 2025

Regresión logística para datos de “Wuupi”

Equipo 3:

Braulio González Esquivel || A01425274

Lucero Jannete López García || A01736938

Pilar Méndez Briones || A01736843

Rubén Uriel Flores Bello || A01737098

Analítica de datos y herramientas de inteligencia artificial II (Gpo 501)

Profesores: Juan Manuel Ahuactzin Larios,
Candy Yuridiana Alemán Muñoz,
Rigoberto Cerino Jiménez y
Alfredo García Suárez

CC: Sexto semestre profesional – Escuela de Negocios
– Licenciatura en Administración y Transformación de
Negocios.

Regresión logística para datos de “Wuupi”



Índice

Contenido

Índice	2
Introducción	3
Botón correcto	4
Juego.....	4
Autopush.....	5
Colores.....	6
Dificultad	7
Mini Juego	8
Número de interacciones	11
Usuarios	12
Usuarios	14
Bibliografía.....	15

Introducción

Para el data frame que utilizamos solo le quitamos los valores nulos porque en los outliers no desaparecen datos al tener más valores '0', lo que ocasiona que los valores de 1 y 2 al ser menos datos, al momento de volver a rellenar nulos de los outliers devolvió ceros, ocasionando que nos tomara en cuenta toda la información del socio formador casando inconsistencias dentro de nuestros análisis.

```
Limite superior permitido botón correcto      2.500000
tiempo de interacción      21.792686
número de interacción      19.500000
auto push      0.000000
tiempo de lección      0.000000
tiempo de sesión      0.000000
dtype: float64
Limite inferior permitido botón correcto      -0.500000
tiempo de interacción      -3.058288
número de interacción      -1.500000
auto push      0.000000
tiempo de lección      0.000000
tiempo de sesión      0.000000
dtype: float64
```

De igual forma, para el análisis de nuestros algoritmos había casos dónde aparecen ceros en nuestra matrices de confusión generando que nuestras métricas fuera difíciles de comprender y analizar, por ello, dentro de nuestro algoritmo implementamos una nueva línea código que generará un balance entre los datos y no solo toma una sola clase debido a que sean más datos.

```
Matriz de confusión:
[[1739   0]
 [  21   0]]
```

```
precision del modelo:
0.0
```

```
exactitud del modelo
0.9880681818181818
```

```
sensibilidad del modelo:
0.0
```

```
#Definimos el algoritmo a utilizar
from sklearn.linear_model import LogisticRegression
algoritmo = LogisticRegression(class_weight='balanced')
```

De igual manera para las variables independientes, la tomamos referentes a la matriz de correlaciones que hicimos en actividades anteriores y generar mejores modelos.

Botón correcto

Variable dependiente	Objetivo	Variables independientes	Precisión (0)	Precisión (1)	Exactitud	Sensibilidad (0)	Sensibilidad (1)
Botón correcto	Las variables objetivo son 1 y 0	'mini juego', 'Juego', 'número de interacción'	0.62	0.68	0.63	0.81	0.45
Observaciones	<p>Mi objetivo era identificar si los usuarios presionaban correctamente el botón (valor 1). Como los datos están relativamente balanceados entre ceros y unos (2668 vs 3197), el modelo logró una sensibilidad alta (0.81), lo cual me parece positivo porque justamente me interesa detectar a quienes sí presionaron el botón correcto.</p> <p>Sin embargo, la precisión (0.62) es más baja, lo que indica que también está clasificando incorrectamente como "correcto" a varios que no lo hicieron bien. La exactitud general (0.63) confirma que hay margen de mejora. A futuro, podría intentar agregar otras variables para mejorar la precisión del modelo.</p>						

Juego

Variable dependiente	Objetivo	Variables independientes	Precisión (Cadetes)	Precisión (asteroides)	Exactitud	Sensibilidad (cadetes)	Sensibilidad (asteroides)
juego	cadetes y asteroides	'Usuario', 'mini juego', 'color presionado'	0.98	0.87	0.94	0.92	0.99

Observaciones	Para esta variable me enfoqué en predecir la categoría "cadetes", ya que era la clase con más datos. El modelo funcionó muy bien: obtuve una precisión de 0.98, una sensibilidad de 0.92 y una exactitud de 0.94. Esto me indica que las variables independientes elegidas (como 'Usuario', 'mini juego' y 'color presionado') fueron bastante efectivas. En general, el modelo casi no se equivoca, lo cual probablemente también se debe a que la clase "cadetes" tenía un buen volumen de datos. Estoy satisfecha con el desempeño en este caso.
---------------	---

Autopush

Variable dependiente	Objetivo	Variables independientes	Precisión (1)	Precisión (0)	Exactitud	Sensibilidad (1)	Sensibilidad (0)
autopush	Las variables objetivas son 0 y 1	Las variables que ocupamos fueron: 'Usuario', 'mini juego', 'color presionado'	0.05	0.99	0.77	0.67	0.75
Observaciones	Esta fue la variable más complicada, ya que tiene un desbalance muy fuerte: la mayoría de los registros son 0 y muy pocos son 1. El modelo logró una exactitud de 0.77, pero esto es engañoso porque está acertando principalmente por predecir ceros, no por identificar los unos. La sensibilidad (0.67) no está tan mal, pero la precisión fue extremadamente baja (0.05). Esto significa que de los casos que el modelo predice como autopush (1), casi todos son falsos positivos. Creo que esto se debe al fuerte desbalance de clases, podríamos probar con otros algoritmos que manejan mejor este tipo de datos desbalanceados.						

Colores

Variable dependiente	Objetivo	Variables independientes	Precisión Violeta	Precisión Otros	Exactitud	Sensibilidad violeta	Sensibilidad otros
Color presionado	violet	'botón correcto', 'número de interacción', 'tiempo de interacción'	0.3545	0.7610	0.6522	0.3515	0.7634
Color presionado 2	green	Las variables que ocupamos fueron: 'botón correcto', 'número de interacción', 'tiempo de interacción'	0.2660	0.7777	0.4420	0.6945	0.3573
Color presionado 3	yellow	Las variables que ocupamos fueron: 'botón correcto', 'número de interacción', 'tiempo de interacción'	0.2723	0.8011	0.425	0.7714	0.3088

Color presionado 4	blue	Las variables que ocupamos fueron: 'botón correcto', 'número de interacción', 'tiempo de interacción'	0.2100	0.7590	0.3744	0.6709	0.2911
Color presionado 5	red	Las variables que ocupamos fueron: 'botón correcto', 'número de interacción', 'tiempo de interacción'	0.0237	1.0	0.7664	1	0.7651
Observaciones	En general, el desempeño de los modelos varía bastante según el color que se intenta predecir. El mejor resultado se obtuvo con el color red, mostrando una excelente sensibilidad de 1.0, alta exactitud (0.7755) y una precisión aceptable (0.2948), siendo el único modelo que detecta todos los casos positivos sin errores. En contraste, los modelos para blue, yellow y green presentan buena sensibilidad (por encima de 0.67), pero su precisión es muy baja (entre 0.22 y 0.25), lo que indica que aunque detectan muchos positivos, también cometen muchos errores al predecir. El modelo de violet es un poco más balanceado, pero igualmente limitado, con precisión baja (0.33) y una exactitud de 0.633. En todos los casos, las matrices de confusión muestran una alta cantidad de falsos positivos o negativos, lo cual sugiere posibles problemas de desbalance en los datos o necesidad de ajustar los modelos.						

Dificultad

Variable dependiente	Objetivo	Variables independientes	Precisión (Niv básico)	Precisión (Niv avanzado)	Exactitud	Sensibilidad (Niv básico)	Sensibilidad (Niv avanzado)
Dificultad	“Niveles básicos”, que representan el nivel 0 y 1, porque tenía un nivel más alto en cuanto a las métricas. La otra variable es ‘Niveles avanzados’	Se utilizaron porque están directamente relacionadas con la complejidad o duración de la actividad del usuario 'tiempo de interacción', 'mini juego', 'Juego'	0.7	0.401	0.547	0.52	0.583
Obs	<p>De todas las veces que el modelo predijo “Niveles básicos”, acertó un 70.6%. Cuando el modelo ve patrones claros de comportamiento “básico” (como tiempos cortos o más aciertos), generalmente tiene razón.</p> <p>La exactitud representa el porcentaje total de aciertos, el cual 54% representa que esa fue la cantidad de porcentaje total, pero no es buena en modelos desbalanceados.</p> <p>De todos los que sí eran “Niveles básicos”, el modelo solo detectó correctamente un 52.9%.</p>						

Mini Juego

Variable dependiente	Objetivo	Variables independientes	Precisión (Asteroides)	Precisión (Otros)	Exactitud	Sensibilidad (Asteroides)	Sensibilidad (Otros)
----------------------	----------	--------------------------	------------------------	-------------------	-----------	---------------------------	----------------------

Minijuego Asteroides	Predecir si una sesión corresponde al minijuego Asteroides o a otro, utilizando variables de interacción del usuario. La otra variable es 'Mini Juego tipo B'	Permiten analizar tanto el comportamiento del usuario como el contexto en el que ocurre la actividad. 'botón correcto', 'tiempo de interacción', 'Juego'	0.26	0.9	0.615	0.86	0.58
Obs	<p>El modelo identifica bien cuando una sesión es realmente "Asteroides" (alta sensibilidad), aunque aún se confunde con casos que no lo son (precisión baja).</p> <p>El modelo aprende a reconocer "Asteroides" porque las personas que lo juegan se comportan de forma más o menos parecida: responden rápido, presionan ciertos botones, están en cierto entorno. Pero como hay otros juegos donde los usuarios también actúan parecido, el modelo se confunde y etiqueta mal. Aun así, el contexto del juego (Juego) le da al modelo una ventaja extra, como si le soplaran la respuesta.</p>						

Variable dependiente	Objetivo	Variables independientes	Precisión (Despegue)	Precisión (Otros)	Exactitud	Sensibilidad (Despegue)	Sensibilidad (Otros)
----------------------	----------	--------------------------	----------------------	-------------------	-----------	-------------------------	----------------------

Minijuego Despegue	Predecir si una sesión pertenece al minijuego o Despegue. La otra variable es 'Mini Juego tipo A', que son los restantes	Elegí las mismas variables porque pertenecen a la misma categoría de minijuego, y fueron las que además, tienen mayor correlación 'botón correcto', 'tiempo de interacción', 'Juego'	0.34	0.977	0.653	0.942	0.587
Obs	Despegue es hasta el mejor minijuego para modelar: mantiene una alta sensibilidad, pero con una mejor precisión que Asteroide. Solo 12 falsos positivos indica que el modelo es confiable cuando predice esta clase. Este parece ser el más fácil de reconocer para el modelo, no solo porque los números son buenos, sino porque los usuarios que lo juegan lo hacen de forma muy parecida entre sí. Eso significa que, cuando alguien entra a Despegue, tiende a jugarlo de una manera consistente, ya sea en el tiempo que tarda, en la forma en que responde (botón correcto) o en el entorno (Juego) en el que aparece.						

Número de interacciones

Variable dependiente	Objetivo	Variables independientes	Precisión 1	Precisión 2	Exactitud	Sensibilidad 1	Sensibilidad 2
Número de interacciones	Predecir si una sesión se encuentra en el rango de interacciones del 1 al 128 (rango que nos dio el código). La segunda variable es 'Interacciones del 129 al 256'	Representan el tipo de actividad visual y el comportamiento temporal del usuario, que están correlacionados con el número de intentos. 'color presionado', 'tiempo de interacción'	0.996	0.206	0.921	0.919	0.972
Obs	<p>El modelo logró predecir con mucha precisión cuándo un usuario realizó pocas o muchas interacciones dentro del minijuego. Esto demuestra que el comportamiento del usuario sí cambia en función del número total de intentos, y que esas diferencias son detectables por el modelo usando solo variables simples como color y tiempo.</p> <p>Eso sugiere que los usuarios que hacen pocas interacciones se comportan de forma muy distinta a los que hacen muchas. Tal vez quienes interactúan poco resuelven rápido, no se equivocan tanto, o usan colores más definidos. Mientras que quienes interactúan más pasan más tiempo, cometen más errores o exploran más.</p>						

Usuarios

Variable dependiente	Objetivo	Variables independientes	Precisión (Usuario)	Precisión (otros)	Exactitud	Sensibilidad (Usuario)	Sensibilidad (otros)
Usuario 1	Valentín	Tiempo de interacción', 'botón correcto', 'número de interacción'	0.03	0.98	0.46	0.69	0.45
Obs	Elegí a Valentín porque fue el paciente que me tocó analizar. Aunque la sensibilidad (0.69) es relativamente buena —lo cual significa que el modelo detecta bien a los casos que sí pertenecen a Valentín—, la precisión (0.03) es muy baja. Esto quiere decir que, de todos los casos que el modelo predice como "Valentín", casi todos están mal clasificados. La exactitud (0.46) también es baja. Creo que esto se debe a la desproporción entre las clases, ya que Valentín solo tiene 158 registros frente a una gran cantidad de "otros usuarios".						
Usuario 2	Yael david (11-163 registros)	tiempo de interacción', 'botón correcto', 'número de interacción'	0.03	0.98	0.61	0.53	0.53
Obs	Para este caso los resultados fueron similares. La sensibilidad (0.53) es algo más baja que en el caso de Valentín, y la precisión (0.03) sigue siendo muy baja. Aunque la exactitud (0.61) subió un poco, sigue sin ser suficiente para confiar plenamente en el modelo. Esto refuerza la idea de que los modelos no están manejando bien la clase minoritaria (en este caso, Yael David).						
Usuario 3	Yeremy Jazmin 24-64 registros	tiempo de interacción', 'botón correcto', 'número de interacción'	0.019	0.99	0.48	0.85	0.54

obs	A pesar de tener una sensibilidad bastante alta (0.85), la precisión (0.019) es todavía más baja que en los casos anteriores. La exactitud (0.48) tampoco ayuda mucho. Este es un buen ejemplo de un modelo que detecta bien los positivos reales (sensibilidad), pero no distingue correctamente a los negativos, clasificando mal a muchos como positivos (baja precisión).						
Usuario 4	Ramiro 6(288)	tiempo de interacción', 'botón correcto', 'número de interacción	0.05	0.97	0.43	0.8	0.52
obs	Ramiro tiene más registros que los anteriores, y eso parece ayudar un poco: la sensibilidad (0.80) es buena, y la precisión mejora a 0.05, aunque sigue siendo baja. La exactitud baja (0.43) indica que el modelo todavía está cometiendo muchos errores. Aquí empiezo a ver que, si bien más datos ayudan a la sensibilidad, no son suficientes por sí solos para mejorar la precisión si hay mucho desbalance.						
Usuario 5	Irving 32 solo hay dos registros	tiempo de interacción', 'botón correcto', 'número de interacción	0.0019	1	0.7	1	0.70
Obs	Este fue el caso más extremo. El modelo logró una sensibilidad de 1, lo cual en principio suena muy bien, pero la precisión es casi nula (0.0019). Lo más probable es que esto se deba a la cantidad mínima de registros que tiene Irving. El modelo predice muchísimos falsos positivos para lograr detectar los pocos casos correctos. Aunque la exactitud global (0.70) parece buena, no significa que el modelo funcione bien para este usuario en específico.						

Usuarios

Podemos concluir que para los algoritmos tengan mayor efectividad, se necesita mayor número de datos o incluso mayor número de entrenamientos. La efectividad de los mismos no se encuentra sujeta al número de variables independientes que se le añade.

Bibliografía

Wuupi. (s.f.). Wuupi. Recuperado el 9 de marzo de 2025, de
<https://wuupi.com.mx/>

El Economista. (2023, 10 de octubre). Conoce los 4 proyectos ganadores de los UP Innovation Awards 2023. Recuperado el 9 de marzo de 2025, de
<https://www.eleconomista.com.mx/el-empresario/Conoce-los-4-proyectos-ganadores-de-los-UP-Innovation-Awards-2023-20231010-0113.html>

Valor Compartido. (2023, 4 de diciembre). Finalizan mentorías a emprendimientos sociales tecnológicos de T-Systems. Recuperado el 9 de marzo de 2025, de
<https://valor-compartido.com/finalizan-mentorias-a-emprendimientos-sociales-tecnologicos-de-t-systems/>