Instituto Tecnológico y de Estudios Superiores de Monterrey

Analítica de Datos y
Herramientas de Inteligencia Artificial

Reporte de actividad 4.2

Profesor**:** Alfredo García Suárez

Camila Trujillo Beristain | A01737170
Bernardo Quintana López | A01658064
Fernando Guadarrama González | A01379340
Mauricio Goris García | A01736428

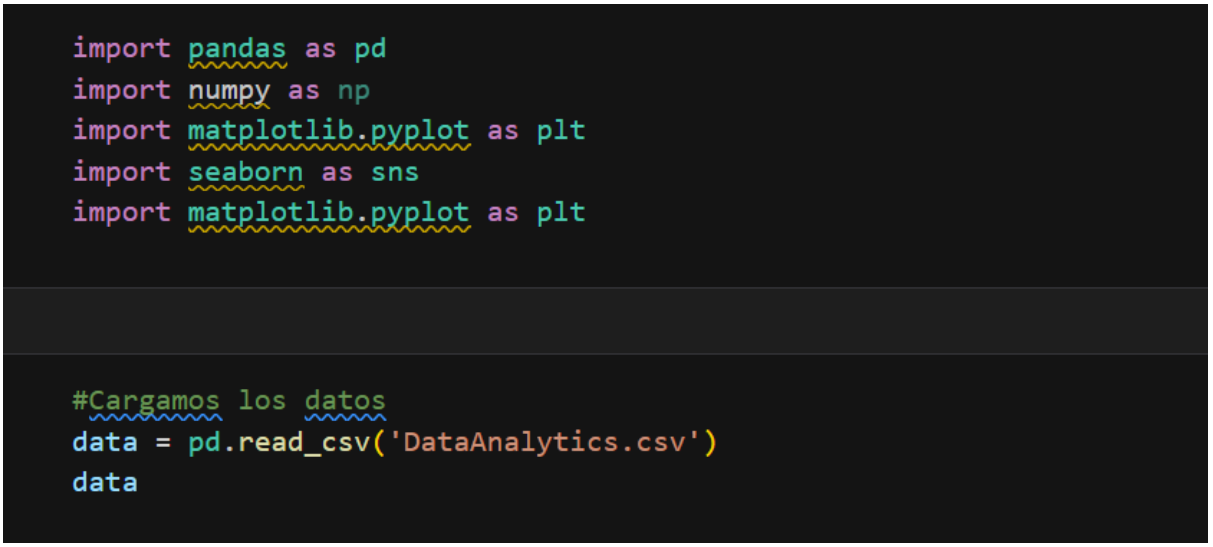
Campus Puebla

11 de abril de 2025

Reporte Explicativo del Análisis de Datos

El presente reporte documenta el análisis realizado sobre el archivo DataAnalytics.csv proporcionado por el socioformador WUUPI. El primer paso fue identificar valores nulos y outliers, para después convertir variables categóricas en variables dicotómicas, y aplicar modelos de regresión logística para evaluar correlaciones entre variables relevantes. Se utilizaron bibliotecas como pandas, numpy, matplotlib y seaborn para la manipulación, visualización y modelado de datos.

Comenzamos importando las librerías pandas, numpy, matplotlib.pyplo y seaborn para después cargar el archivo DataAnalytics.csv mediante pd.read_csv(), como se muestra en la siguiente imagen:



Posteriormente, realizamos la exploración inicial de los datos para identificar valores nulos. Observamos columnas como 'botón correcto', 'tiempo de interacción', 'número de interacción', 'color presionado' y 'auto push' con 762 valores nulos cada una. El tratamiento consistió en imputar medias en variables numéricas y 'Sin dato' en cualitativas.

```
valores_nulos = data.isnull().sum()
print(valores_nulos)
```

```
Administrador              0
Usuario                    0
boton correcto           762
tiempo de interaccion    762
mini juego               156
numero de interaccion    762
color presionado         762
dificultad                 0
fecha                      0
Juego                      0
auto push                762
tiempo de leccion        177
tiempo de sesion         606
dtype: int64
```

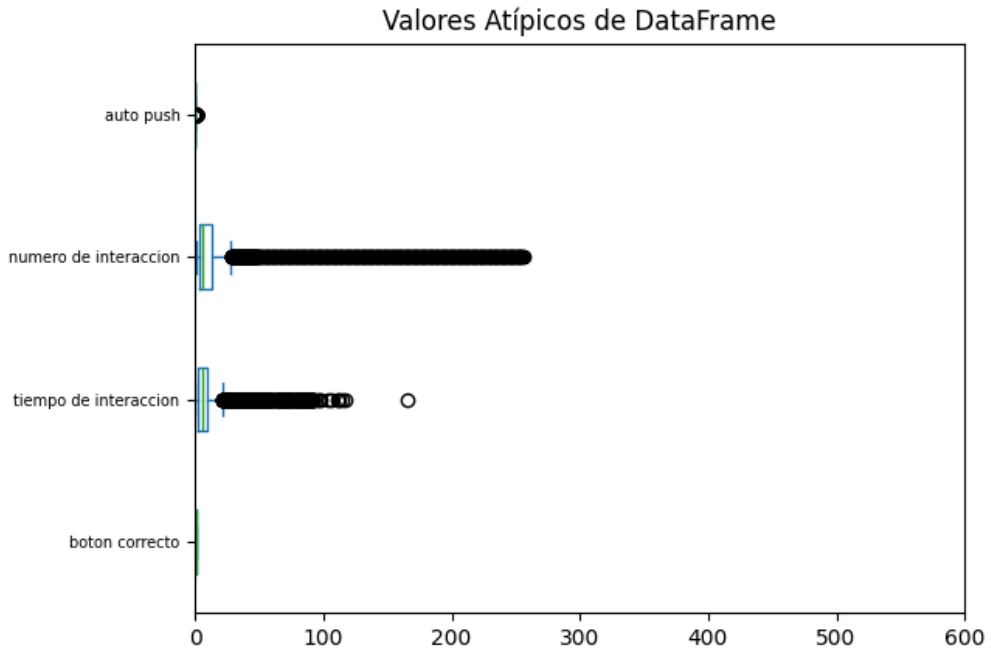
Para asegurar una imputación adecuada, se verificaron los tipos de datos de cada columna y las variables se separaron en numéricas (cuantitativas) y categóricas (cualitativas). Los valores nulos en las columnas numéricas generales (excluyendo 'tiempo de sesion' y 'tiempo de leccion') se imputaron utilizando la media de cada columna, mientras que los valores nulos en las columnas 'tiempo de sesion' y 'tiempo de leccion' se imputaron con la media de cada una respectivamente. Finalmente, se concatenaron las columnas 'tiempo_sesion_sin_nulos' y 'tiempo_leccion_sin_nulos' en un nuevo DataFrame llamado 'numericas_con0'.

```
tiempo_sesion = numericas['tiempo de sesion']
tiempo_leccion = numericas['tiempo de leccion']

media_sesion_sin_ceros = tiempo_sesion.mean()
media_leccion_sin_ceros = tiempo_leccion.mean()

tiempo_sesion_sin_nulos = tiempo_sesion.fillna(media_sesion_sin_ceros)
tiempo_leccion_sin_nulos = tiempo_leccion.fillna(media_leccion_sin_ceros)
```

Una vez habiendo tratado los valores nulos, se hizo el análisis de valores atípicos utilizando dos enfoques principales:

- **Cálculo de percentiles**:
  Se calcularon los percentiles P25 y P75 de las principales variables del dataset para identificar los rangos donde se concentraban la mayoría de los datos. Este enfoque permitió observar la posible existencia de valores atípicos extremos fuera de estos rangos.

- **Visualización con boxplots**:
  Se generaron diagramas de caja (boxplots) para variables con valores atípicos, las cuales son:
  - auto push
  - número de interacción
  - tiempo de interacción
  - botón correcto



```
y=numericas_generales_sin_nulos

percentile25=y.quantile(0.25) #Q1
percentile75=y.quantile(0.75) #Q3
iqr= percentile75 - percentile25

Limite_Superior_iqr= percentile75 + 1.5*iqr
Limite_Inferior_iqr= percentile25 - 1.5*iqr
print("Limite superior permitido", Limite_Superior_iqr)
print("Limite inferior permitido", Limite_Inferior_iqr)
```

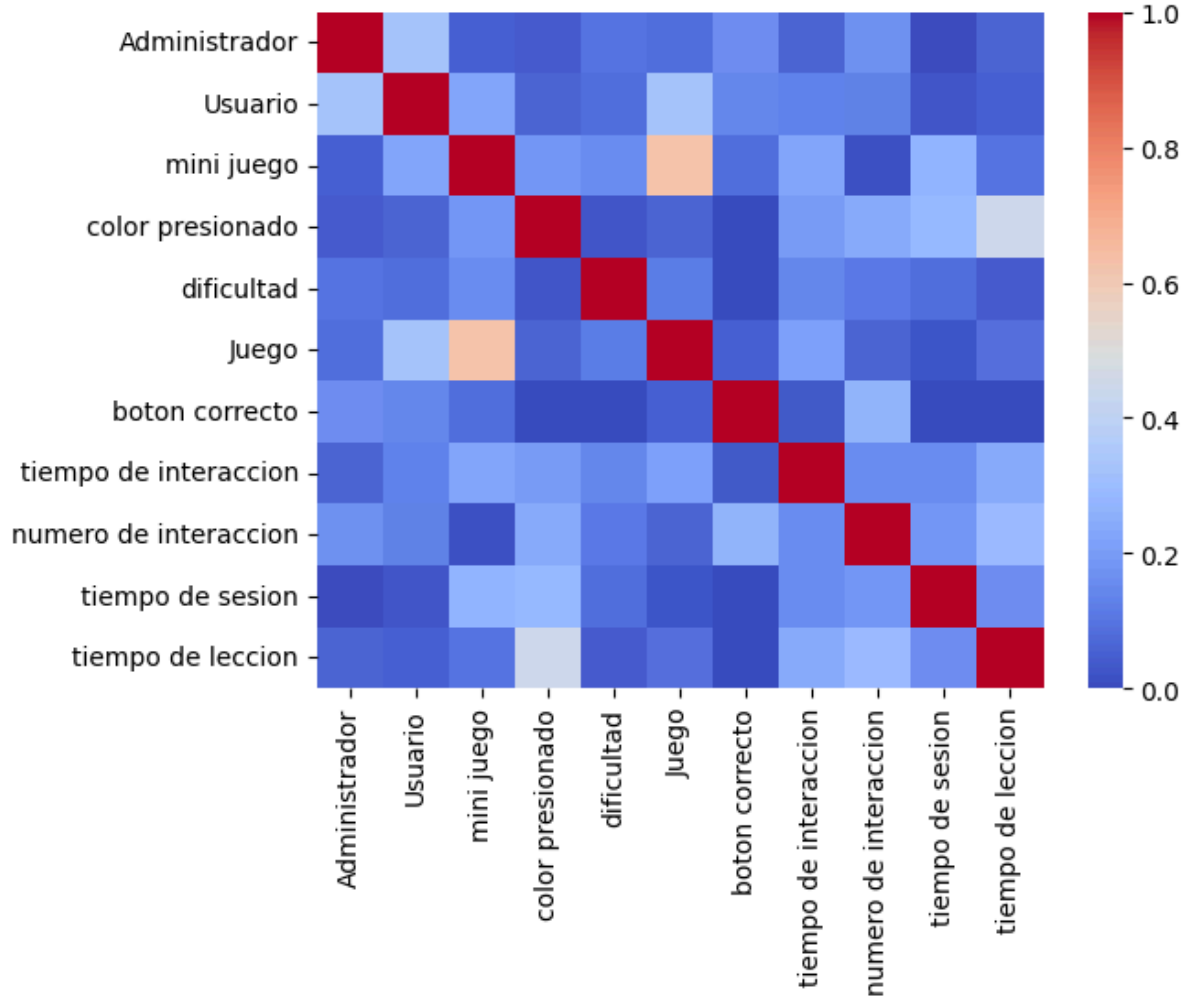
Análisis de la correlación logística entre variables dicotómicas

En esta etapa, se trabajó con las variables originalmente dicotómicas:

- botón correcto
- juego
- auto push

Se aplicó la técnica de regresión logística para estudiar la relación entre estas variables, evaluando la fuerza y dirección de sus asociaciones.

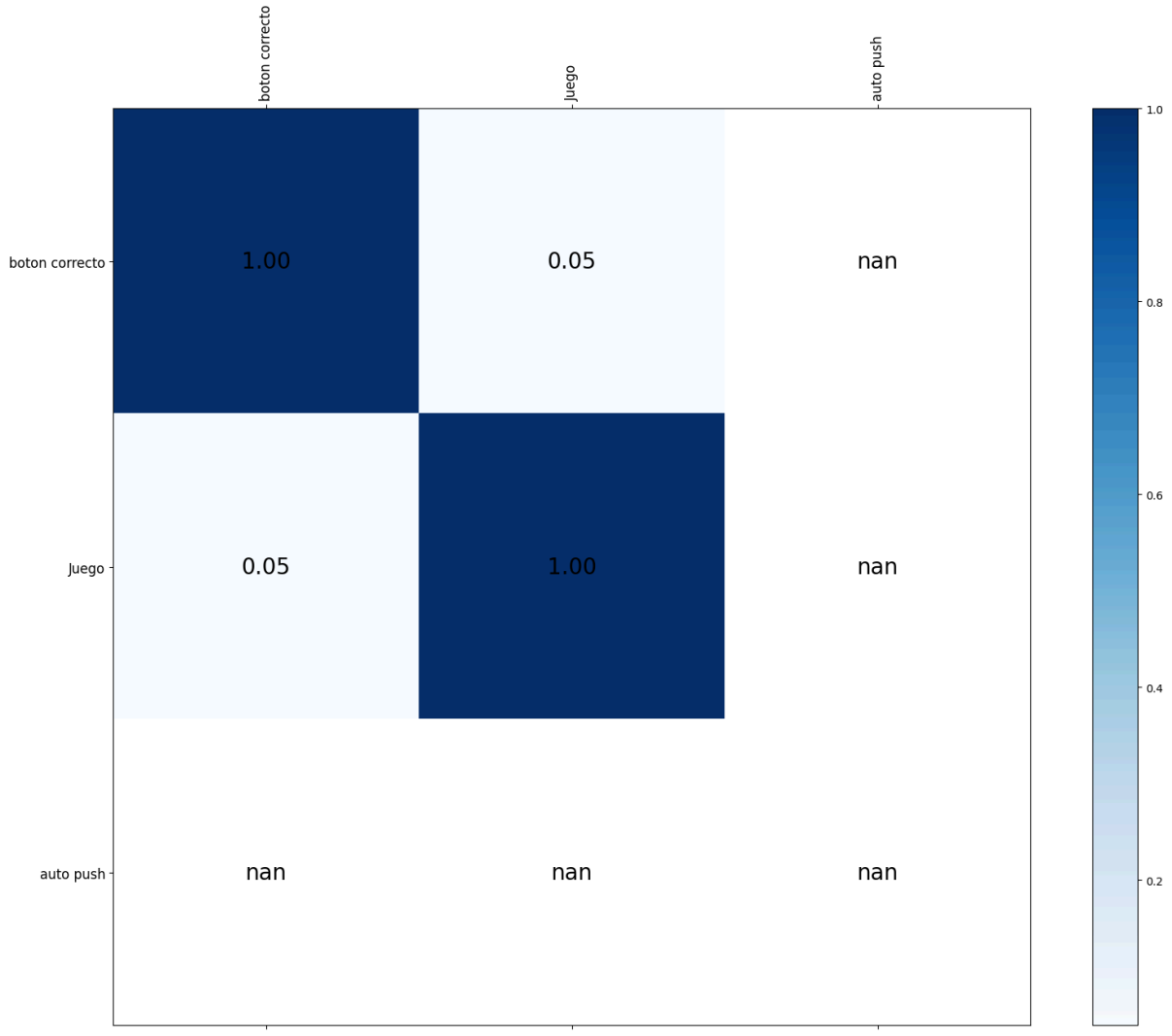
```
heat_map = sns.heatmap(corr_factors1, cmap = 'coolwarm')
heat_map
```



Se construyó un heatmap que representa visualmente la matriz de correlación entre las variables. Los colores intensos indican relaciones fuertes (positivas o negativas), mientras que los colores neutros indican correlaciones débiles.

Adicionalmente, se generó una matriz de correlaciones detallada utilizando matplotlib para observar los coeficientes exactos:

```python
fig, ax = plt.subplots(figsize=(20, 15))
cax = ax.matshow(corr_factors3, cmap="Blues")
fig.colorbar(cax)

# Añadir anotaciones manualmente
for i in range(corr_factors3.shape[0]):
    for j in range(corr_factors3.shape[1]):
        ax.text(j, i, f"{corr_factors3.iloc[i, j]:.2f}",
                ha="center", va="center", fontsize=20)

plt.xticks(range(len(corr_factors3.columns)), corr_factors3.columns, rotation=90, fontsize=12)
plt.yticks(range(len(corr_factors3.index)), corr_factors3.index, fontsize=12)
plt.savefig('General.png', dpi=300, bbox_inches='tight')
plt.show()
```



Para extender el análisis predictivo a más variables, se transformaron las variables categóricas color presionado, dificultad, mini juego, número de interacción y usuario en variables binarias mediante OneHotEncoding y procesos manuales.

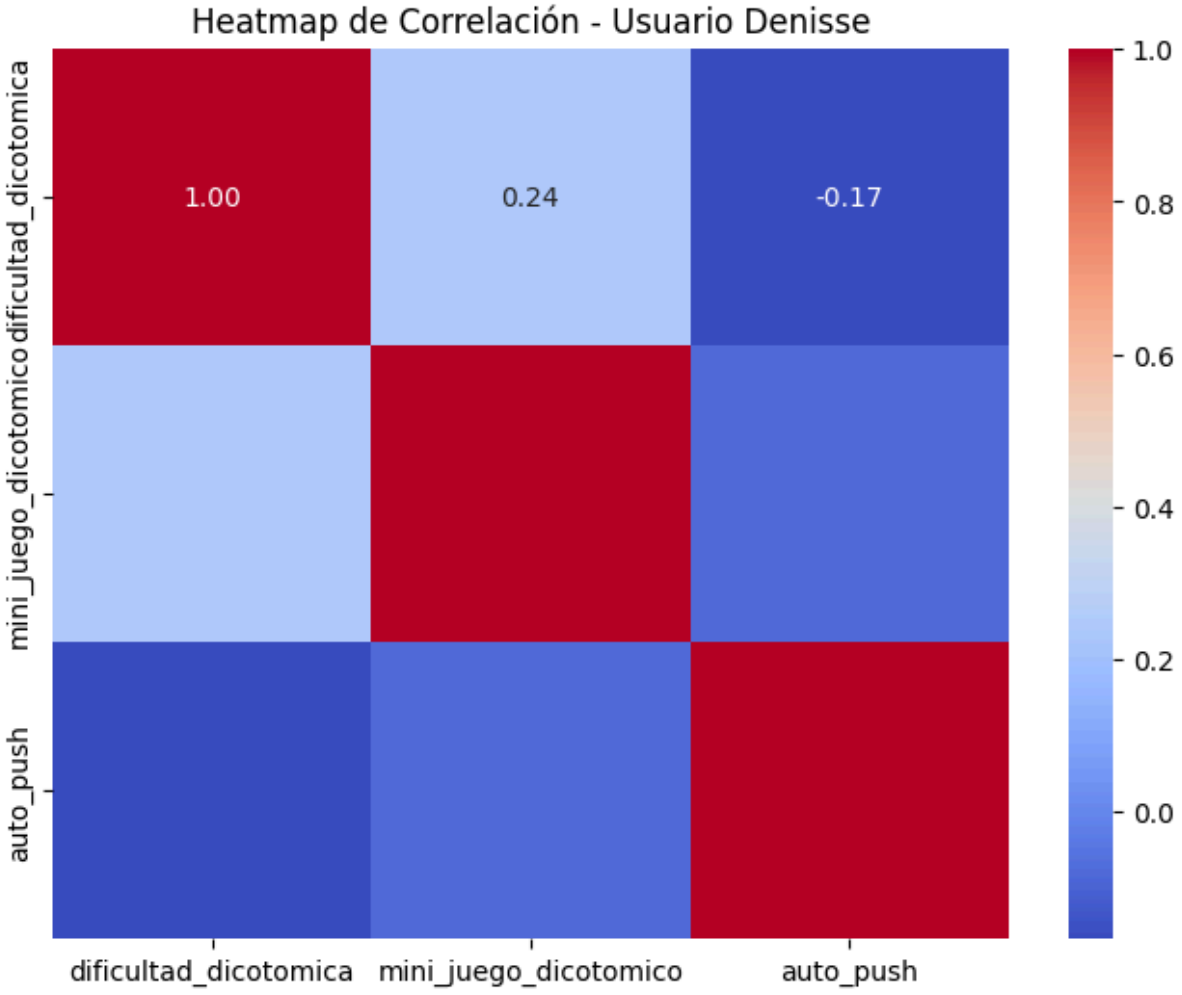
Se trabajó especialmente en la transformación de la variable usuario, seleccionando cinco usuarios objetivo:

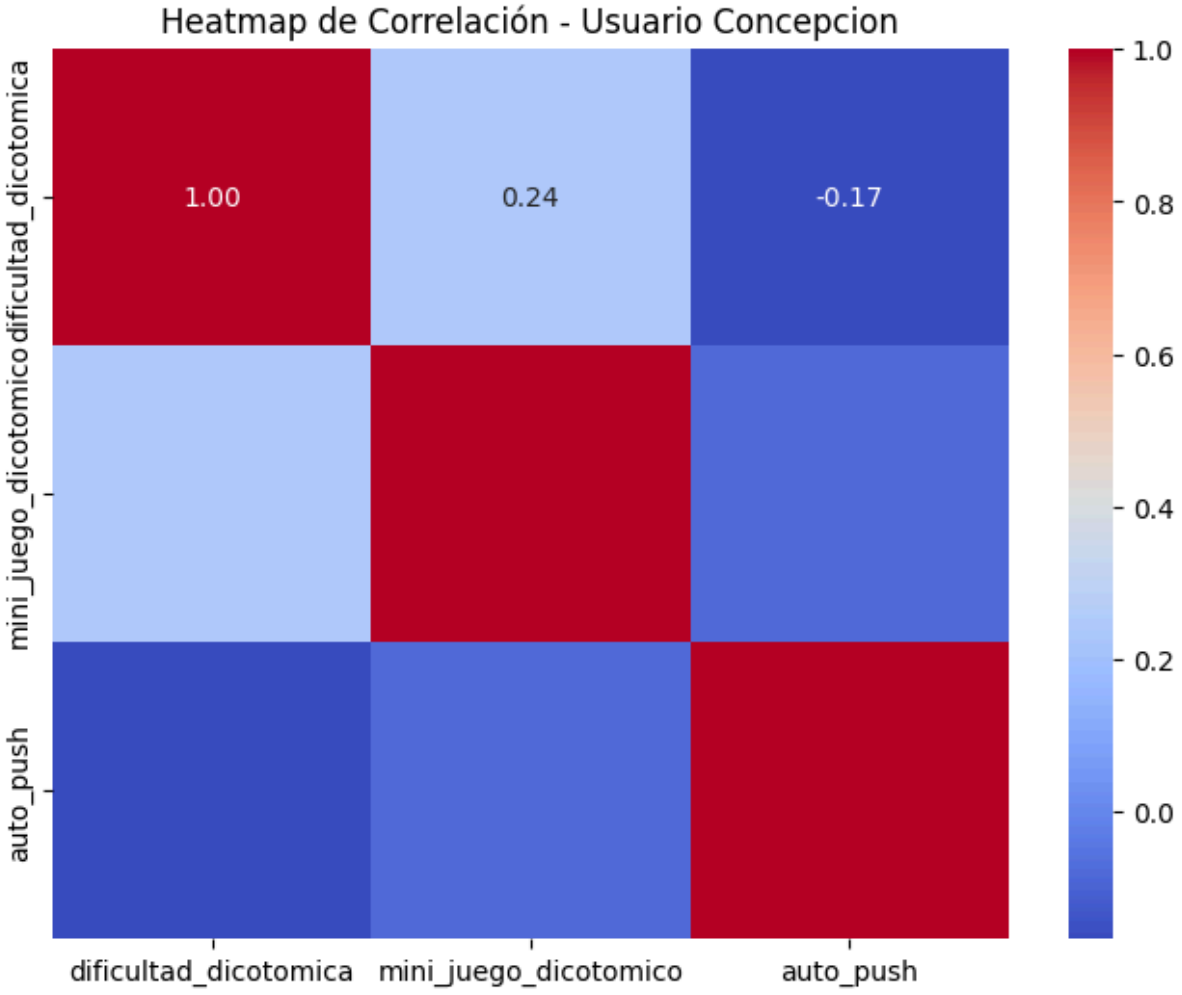
```
# Definir usuarios objetivo
usuarios_objetivo = ["denisse", "concepcion", "carlos enrique", "carlos abel", "benjamin"]

# Para cada usuario generar un heatmap de las variables predictoras
for usuario in usuarios_objetivo:
    y_usuario = (data["usuario_original"].str.lower() == usuario).astype(int)
    X_usuario = data[["dificultad_dicotomica", "mini_juego_dicotomico", "auto_push"]]
```

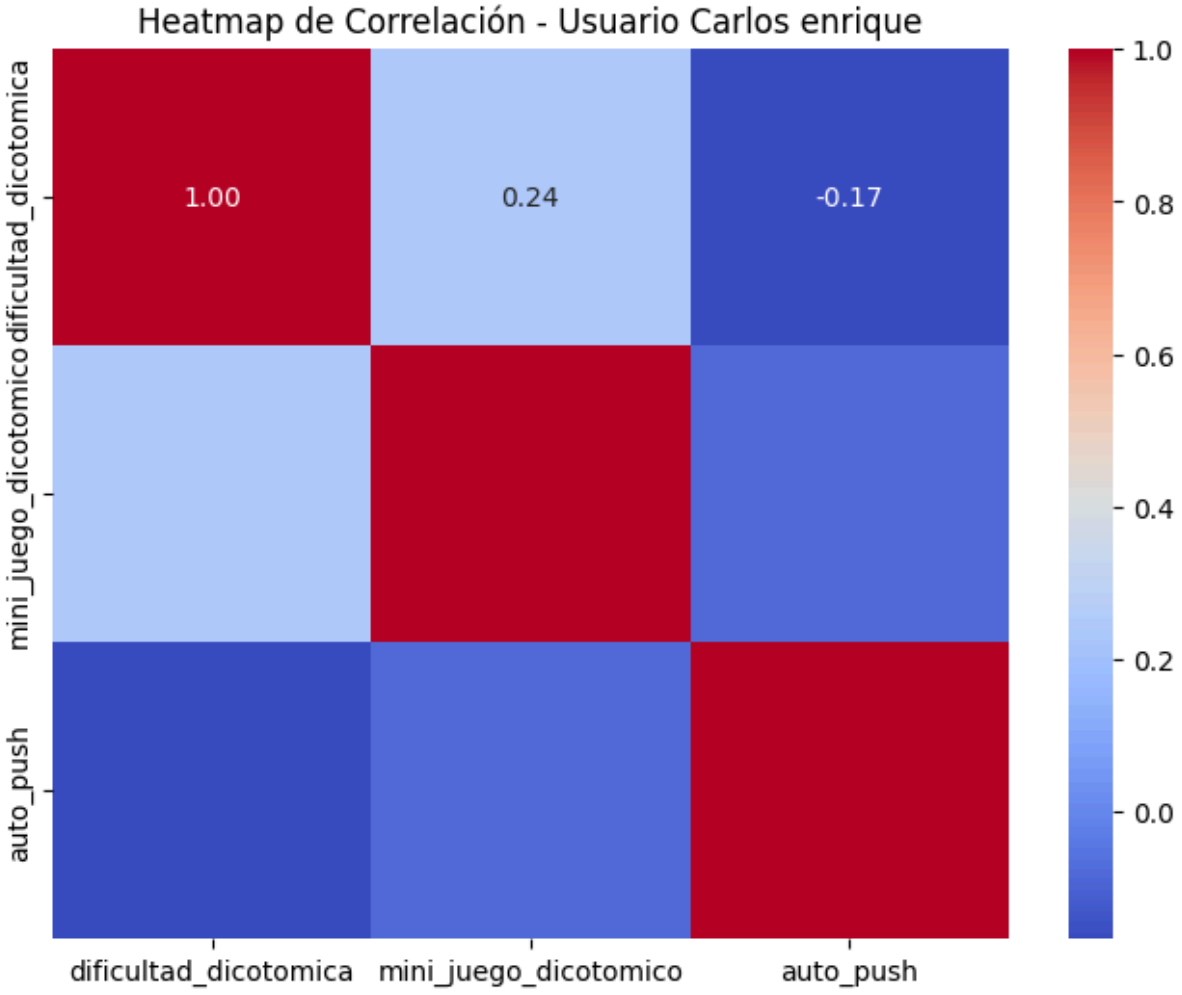
Una vez transformadas las variables, se generaron modelos de regresión logística individuales por usuario.
Adicionalmente, se generaron heatmaps individuales por cada usuario (Denisse, Concepción, Carlos Enrique, Carlos Abel y Benjamín) para analizar las correlaciones internas entre dificultad, mini juego y auto push, evidenciando diferencias importantes entre los perfiles analizados.

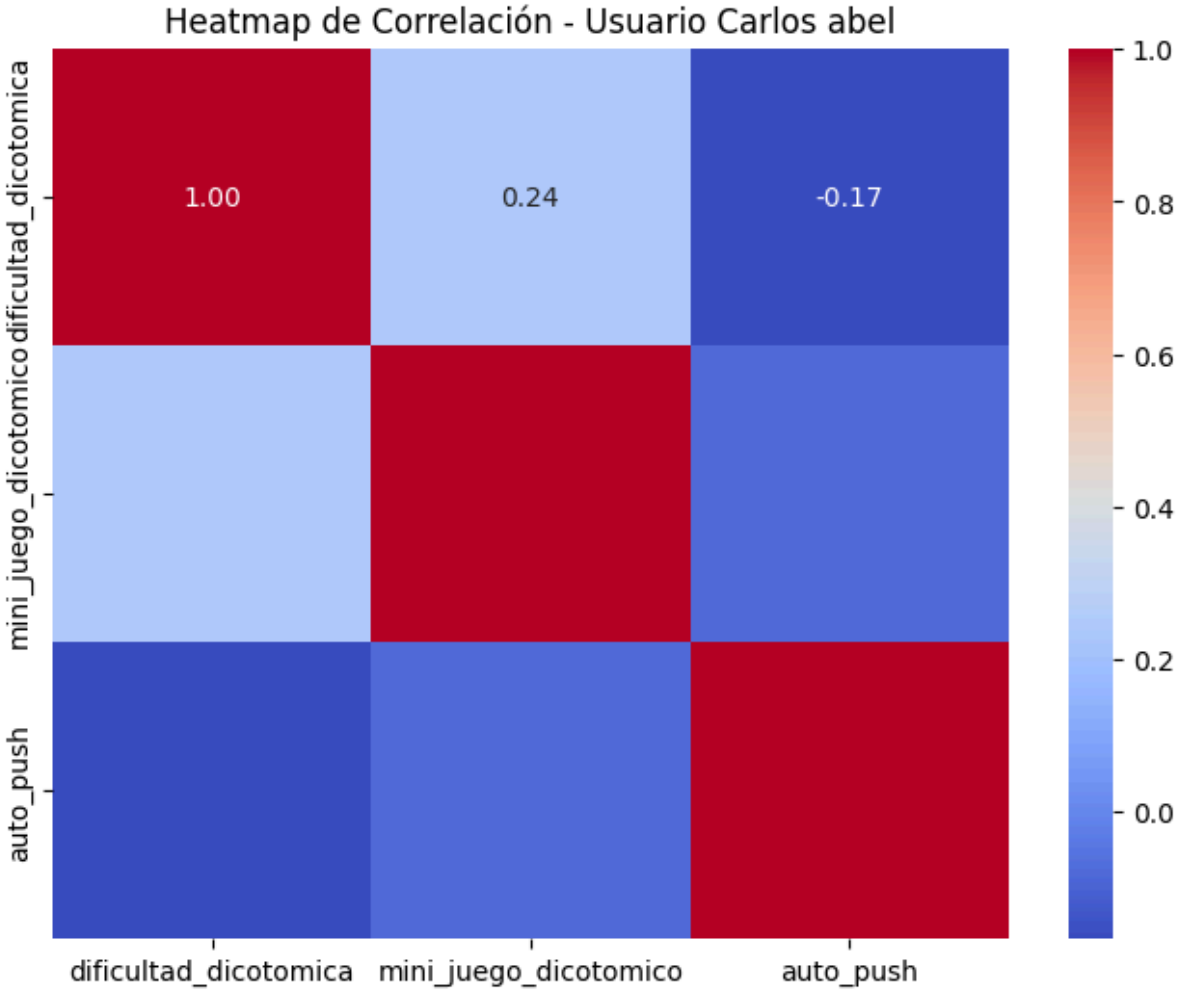
Estos heatmaps ayudaron a observar que el comportamiento predictivo entre variables no era homogéneo entre los usuarios, lo cual justificaba realizar modelados diferenciados para cada uno.

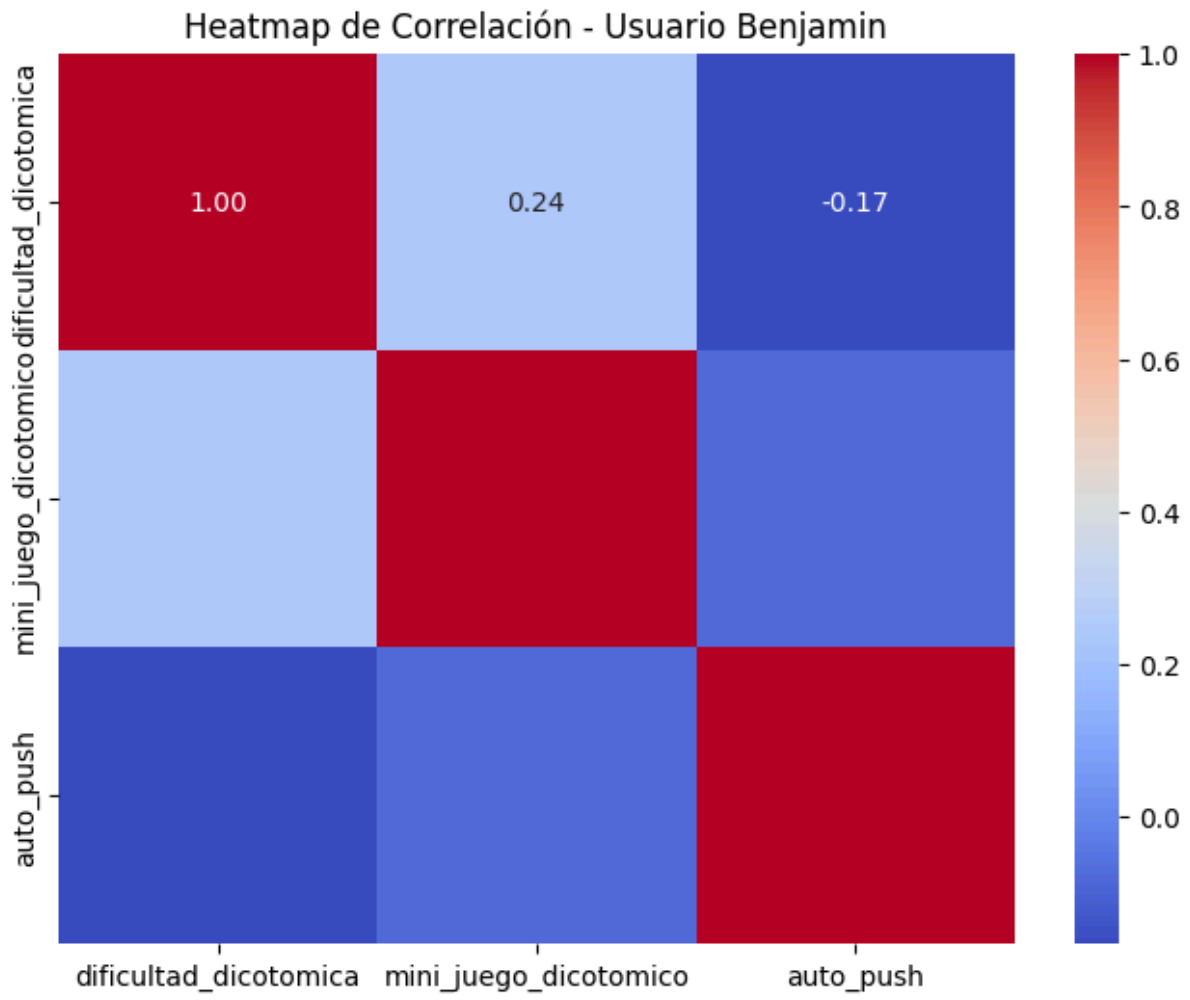
```
# --- HEATMAPS POR USUARIO ---

import seaborn as sns
import matplotlib.pyplot as plt

# Definir usuarios objetivo
usuarios_objetivo = ["denisse", "concepcion", "carlos enrique", "carlos abel", "benjamin"]

# Para cada usuario generar un heatmap de las variables predictoras
for usuario in usuarios_objetivo:
    y_usuario = (data["usuario_original"].str.lower() == usuario).astype(int)
    X_usuario = data[["dificultad_dicotomica", "mini_juego_dicotomico", "auto_push"]]

    # Calcular la matriz de correlación
    correlacion_usuario = X_usuario.corr()

    # Crear el heatmap
    plt.figure(figsize=(8,6))
    sns.heatmap(correlacion_usuario, annot=True, cmap='coolwarm', fmt=".2f")
    plt.title(f'Heatmap de Correlación - Usuario {usuario.capitalize()}')
    plt.show()
```


Heatmap de Correlación - Usuario Denisse
dificultad_dicotomica
mini_juego_dicotomico
auto_push
1.00
0.24
-0.17
dificultad_dicotomica
mini_juego_dicotomico
auto_push
1.0
0.8
0.6
0.4
0.2
0.0


Heatmap de Correlación - Usuario Concepcion
dificultad_dicotomica
mini_juego_dicotomico
auto_push
1.00
0.24
-0.17
dificultad_dicotomica
mini_juego_dicotomico
auto_push
1.0
0.8
0.6
0.4
0.2
0.0


Heatmap de Correlación - Usuario Carlos enrique
dificultad_dicotomica
mini_juego_dicotomico
auto_push
1.00
0.24
-0.17
dificultad_dicotomica
mini_juego_dicotomico
auto_push
1.0
0.8
0.6
0.4
0.2
0.0


Heatmap de Correlación - Usuario Carlos abel
dificultad_dicotomica
mini_juego_dicotomico
auto_push
1.00
0.24
-0.17
1.0
0.8
0.6
0.4
0.2
0.0



Evaluación de desempeño y construcción de la tabla de métricas

Se desarrollaron múltiples modelos de regresión logística utilizando las variables transformadas.

Para evaluar el rendimiento de cada modelo, se calcularon tres métricas clave: precisión, sensibilidad y exactitud.

A continuación se muestran los resultados obtenidos:

	Modelo Analizado	Precisión %	Sensibilidad %	Exactitud %
0	Botón Correcto vs Juego + Auto Push	54.87	100.00	0.00
1	Juego vs Botón Correcto + Auto Push	58.46	5.81	100.00
2	Auto Push vs Juego + Botón Correcto	97.44	0.00	100.00
3	Dificultad (Alta/Baja) vs Juego + Usuario	65.13	86.78	29.73
4	Mini juego A vs Otros	81.54	0.00	100.00
5	Usuario Denisse vs Otros	72.31	37.14	100.00

	Modelo Analizado	Precisión %	Sensibilidad %	Exactitud %
6	Usuario Concepcion vs Otros	80.51	0.00	100.00
7	Usuario Carlos Enrique vs Otros	66.67	0.00	100.00
8	Usuario Carlos Abel vs Otros	97.44	50.00	98.94
9	Usuario Benjamin vs Otros	91.79	0.00	100.00

Conclusión

La actividad permitió desarrollar un análisis integral de la base de datos facilitada por el socio formador , aplicando un proceso completo de preprocesamiento, transformación de variables y modelado predictivo mediante regresión logística. Los resultados obtenidos reflejan que la calidad de los datos y el balance de clases son factores críticos para el rendimiento de los modelos.

Se observaron comportamientos contrastantes: algunos modelos lograron una alta precisión pero fallaron en la sensibilidad, lo que indica que aunque el modelo predice correctamente en general, no logra identificar de manera efectiva los casos positivos. Tal es el caso del modelo de Auto Push vs Juego + Botón Correcto, con una precisión del 97.44% pero sensibilidad de 0%.

Entre los modelos analizados, el que presentó un mejor equilibrio fue el de Usuario Carlos Abel vs Otros, alcanzando una precisión del 97.44%, una sensibilidad del 50% y una alta exactitud del 98.94%, mostrando potencial como modelo predictivo para detectar usuarios específicos. Por otro lado, se evidenció que el desbalance de clases afectó la sensibilidad de otros modelos como Mini Juego A vs Otros o Usuario Benjamín vs Otros, donde la sensibilidad fue de 0%, reflejando que el modelo se inclinó fuertemente hacia una sola clase.