Universidad de Antioquia

Ingeniería de Sistemas

Introducción a la inteligencia artificial

PROYECTO ENTREGA 2

PRESENTA:
Jonatan Stiven Restrepo Lora

Tutor Principal:
Raul Ramos Pollan

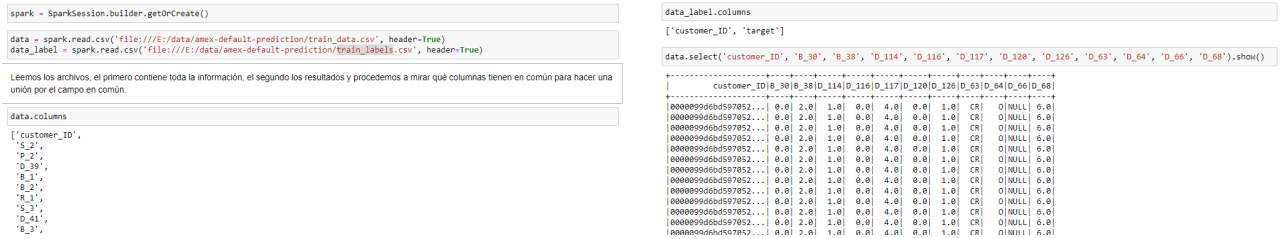
Resumen

Para comenzar con la tarea predictiva, se inició el trabajo con el conjunto de datos suministrados. Sin embargo, estos datos resultaron ser bastante voluminosos. Por lo tanto, se decidió seleccionar aleatoriamente una muestra de aproximadamente 6000 registros para su posterior entrenamiento.

Estos registros se guardaron y se sometieron a una evaluación con el propósito de identificar sus variables categóricas y determinar qué operaciones eran necesarias para su procesamiento. Luego, después de un análisis y filtro, se procedió a llevar a cabo la tarea de limpieza, que incluyó la imputación de valores faltantes de acuerdo a los resultados de los análisis previamente diseñados.

Selección de muestras

Se comenzó por cargar los conjuntos de datos, **train_data** y **train_labels**, utilizando la herramienta de pandas. Sin embargo, debido al tamaño de los archivos, esta tarea resultó ser inviable. Como solución, se optó por utilizar Spark, dado que permite manejar volúmenes de datos más grandes.



Cuadro 1: Comparación de datos

Luego, se procedió a identificar las columnas similares con el fin de fusionar los datos y trabajar con un solo conjunto que incluyera tanto las variables predictoras como el objetivo. Se consideró la variable **customer_ID** como candidata para este proceso. Sin embargo, al intentar llevar a cabo la operación, se presentaron errores, como el inesperado aumento en el tamaño del conjunto de datos. Esto motivó la necesidad de primero dividir la muestra para poder gestionarla de manera adecuada y proceder a guardarla en un **.xslx**

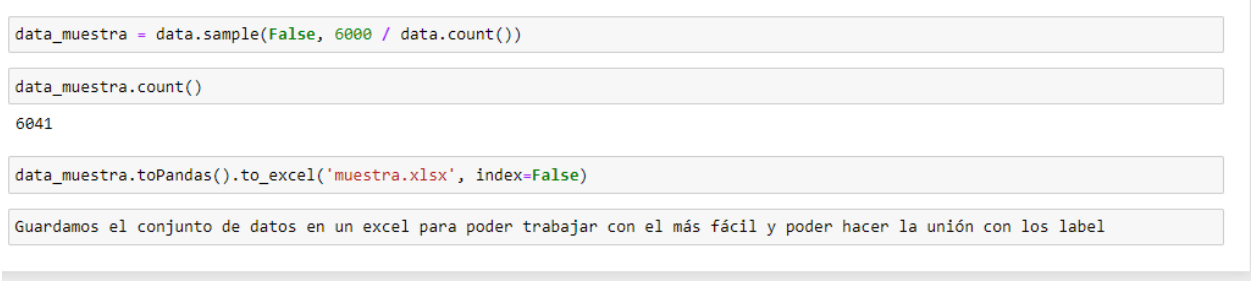


Figura 1: Guardado de los datos

Merge Datasets

Una vez los datos divididos, procedemos a hacer un merge de la muestra extraída y **train_labels** para así poder agregar el objetivo correspondiente al **customer_ID** de nuestro conjunto de datos para proceder a guardarlo dado que este sera el dataset con el que estaremos trabajando.

```
print(f'cantidad de registros: {len(pd_label)}')
```

```
cantidad de registros: 458913
```

```
data_full = pd_muestra.merge(pd_label, on='customer_ID')
data_full.head()
```

	customer_ID	S_2	P_2	D_39	B_1	B_2	R_1	S_3	D_41	B_3	...	D_137	D_138
0	000adf2938f771f75a581b65107024eddeae70684778c0...	2017-04-25	0.885999	0.009377	0.008894	1.006219	0.008716	0.065198	0.001650	0.009946	...	NaN	NaN
1	001a152e1893ab8372e7c9627c9de2e024399f2660d5d8...	2017-11-10	0.750890	0.037378	0.045959	1.001278	0.000312	0.087636	0.004212	0.001730	...	NaN	NaN
2	001e2ceaf1421f1477c0de9ba1c9357b9d278f7b670ab7...	2018-03-26	0.607227	0.002678	0.001271	0.812964	0.005415	NaN	0.005049	0.006548	...	NaN	NaN
3	0034f7e366a41d2500643c7dd0faa6302ce944743ccdf5...	2018-01-07	0.775546	0.003486	0.234352	0.040793	0.256024	0.172966	0.000081	0.225639	...	NaN	NaN
4	00394e07aa3f71174f8bedfd16d64f194c80ad9445e17f...	2017-04-27	0.358659	0.001814	0.028545	1.004951	0.009079	0.765318	0.003546	0.009843	...	NaN	NaN

5 rows × 191 columns

```
print(f'cantidad de registros: {len(data_full)}')
```

```
cantidad de registros: 6041
```

Una vez comprobado que el merge lo altero el DF y se procede a guardarlos en un csv y txt

```
data_full.to_excel('data_full.xlsx', index=False)
```

```
data_full.to_csv('data_full.txt', sep='\t', index=False)
```

Figura 2: Muestra completa

Limpeza de Datos

En el proceso de limpieza de datos, comenzamos por identificar las variables categóricas, a pesar de que ya se habían especificado en el conjunto de datos original como 'B_30', 'B_38', 'D_114', 'D_116', 'D_117', 'D_120', 'D_126', 'D_63', 'D_64', 'D_66', 'D_68'. Estas variables se dividieron en dos categorías: nominales y ordinales. En primer lugar, se llevó a cabo la limpieza y transformación de las variables nominales.

Dentro de las variables nominales, se identificaron `'S_2', 'D_63' 'D_64'.` Se observó que `'S_2'` representaba fechas, por lo que se procedió a ajustar el tipo de dato utilizando la biblioteca pandas. Para las demás variables nominales, se completaron los valores faltantes utilizando la moda de los datos.

Posteriormente, se aplicó el método de codificación One-Hot Encoding para organizar estas variables categóricas.

```
df = pd.get_dummies(df, columns=['D_63', 'D_64'])
```

```
df.rename(columns={'D_64_-1': 'D_64_I'}, inplace=True)
```

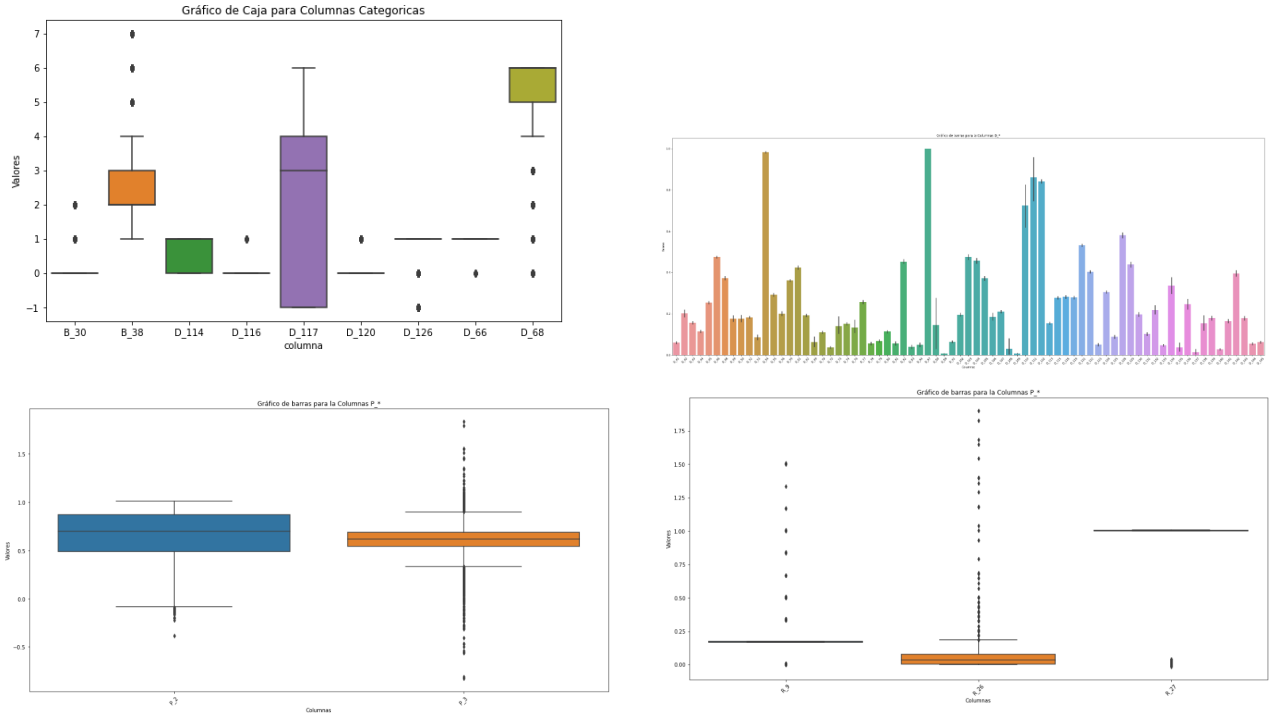
```
df[['D_63_CO', 'D_63_CR', 'D_63_CL', 'D_63_XZ', 'D_63_XZ', 'D_63_XM', 'D_63_XL', 'D_64_O', 'D_64_U', 'D_64_R',
    'D_64_I', 'target']]
```

	D_63_CO	D_63_CR	D_63_CL	D_63_XZ	D_63_XZ	D_63_XM	D_63_XL	D_64_O	D_64_U	D_64_R	D_64_I	target
0	1	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	1	0	0	0
2	1	0	0	0	0	0	0	1	0	0	0	1
3	1	0	0	0	0	0	0	1	0	0	0	0
4	0	0	1	0	0	0	0	1	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...
6036	1	0	0	0	0	0	0	0	1	0	0	0
6037	1	0	0	0	0	0	0	0	1	0	0	1
6038	1	0	0	0	0	0	0	0	0	1	0	1
6039	1	0	0	0	0	0	0	0	1	0	0	0
6040	0	1	0	0	0	0	0	1	0	0	0	0

6041 rows × 12 columns

Figura 3: Manejo de variables categoricas

Luego, procedimos a analizar el resto de las variables para verificar si contenían valores atípicos, con el fin de determinar el método más apropiado para completar los valores faltantes. Si se identificaban datos atípicos y, después de examinar la distribución de los datos, se determinaba que llenar los valores faltantes con la moda era la mejor opción para evitar una desviación significativa.



Cuadro 2: Analisis de los grupos

Concluciones

Se ha completado la separación y limpieza de los datos de trabajo. Los siguientes pasos a seguir serán la evaluación de las relaciones entre las variables y el objetivo con el fin de determinar qué algoritmo de clasificación puede ser más adecuado para abordar la pregunta en cuestión.

- Folder de trabajo de los notebooks
  Nota: *los notebooks fueron trabajados en local, si los va a ejecutar en colab asegurece que la ruta de los archivos sea la correcta.*

- video explicatorio