

CREDI FIEL

DATATHON

OPTIMIZACIÓN DE COSTOS Y COBRANZA A
CLIENTES.

JACK IN THE CODE

Ana Lidia Arteaga Bretón

Daniela Ortiz Blanco

Fernanda Ojeda Tamayo

René Cumplido Feregrino



NUEVA BASE DE DATOS

Consolidación multi-anual de datos

Todos los años la
volvimos una sola
base de datos.

A B A UNION B

1	3	→	1
2	4		2
3	5		3
			4
			5

Estandarización de columnas

Estandarizamos los nombres.

Nombre Cliente

nombre_cliente

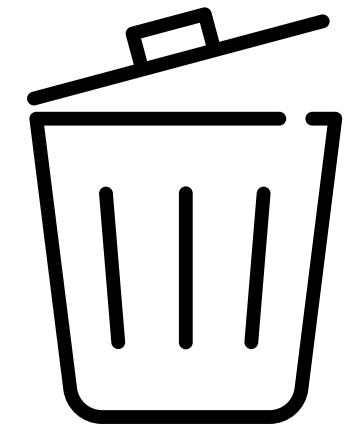
Integración relacional por claves

Concentramos toda la información relevante por cada intento de cobro.

The diagram shows two source tables at the top. The left table has columns A, B, and C. The right table has columns C, D, and E. Arrows from the B column of the left table and the D column of the right table point to the B and D columns of a target table at the bottom. The target table has columns B, C, and D. The C column in the target table is shaded with diagonal lines, indicating it is a derived column.

Selección de variables predictivas

Eliminamos todo lo
que hace ruido en
nuestro modelo.





**MINIMIZACIÓN DE COSTO POR
COMISIONES Y OPTIMIZACIÓN DE ENVÍOS**

CREDIFIEL

QUE HACE NUESTRO MODELO

Realizamos un modelo de minimización con programación lineal usando PuLP.

- El objetivo fue seleccionar el intento de cobro más económico por crédito, entre múltiples opciones, minimizando el costo total de domiciliación.
- Se usaron variables binarias $x[i]$ para decidir si un intento se selecciona o no.

```
[ ] # Modelo
    modelo = LpProblem("Minimizacion_Costos_Domiciliacion", LpMinimize)

    # Variables binarias de decisión
    x = LpVariable.dicts("x", df_validos['var_id'], cat=LpBinary)

    # Función objetivo: minimizar el costo total
    modelo += lpSum(x[i] * df_validos.loc[i, 'costo'] for i in df_validos['var_id']), "CostoTotal"

    # Restricción: solo un intento por crédito
    for credito in df_validos['idcredito'].unique():
        indices = df_validos[df_validos['idcredito'] == credito]['var_id']
        modelo += lpSum(x[i] for i in indices) == 1, f"UnEnvio_{credito}"
```

 modelo.solve()

 1

FUNCIÓN OBJETIVO

$\sum x_i \cdot \text{costo}$

CREDIFIEL

OUTPUT

OTORGA

Una guía para el usuario sobre las estrategias cuya implementación permite minimizar los costos asociados a las comisiones.

El algoritmo implementado otorga dos distintos outputs:

- Un csv que otorga las estrategias consideradas como las más eficientes.
- Se imprimen en la consola las 5 estrategias que generar la menor cantidad de costos.

Estrategias más eficientes:				
	emisora	banco	tipoenvio	num_creditos \
0	BANAMEX CUENTA	BANAMEX	CUENTA	452
1	BANORTE CLABE TRADICIONAL	BANORTE	TRADICIONAL	7245
2	BANORTE CLABE INTERBANCARIO	AZTECA	INTERBANCARIO	1
3	BANORTE CLABE INTERBANCARIO	HSBC	INTERBANCARIO	4
4	BBVA CLABE TRADICIONAL	BBVA MEXICO	TRADICIONAL	7646
	costo_total	costo_promedio		
0	791.0	1.75		
1	18112.5	2.50		
2	4.5	4.50		
3	18.0	4.50		
4	61168.0	8.00		



**MAXIMIZACIÓN DE COBRANZA A
CLIENTES**

CREDIFIEL

CONTENIDO

Realizamos un modelo de maximización con programación lineal usando PuLP.

Preparación de datos

- Se eliminaron registros sin monto cobrado.
- Se creó un identificador único por intento (var_id).
- Se usaron variables binarias x[i] para decidir si un intento se selecciona o no.

```
# === 3. Crear el modelo ===
modelo = LpProblem("Maximizacion_Cobranza", LpMaximize)

# Variables binarias por intento
x = LpVariable.dicts("x", df['var_id'], cat=LpBinary)

# Función objetivo: Maximizamos la suma de lo cobrado
modelo += lpSum(x[i] * df.loc[i, 'montocobrado'] for i in df['var_id']), "TotalCobrado"

# Restricción: una sola selección por idcredito
for credito in df['idcredito'].unique():
    indices = df[df['idcredito'] == credito]['var_id']
    modelo += lpSum(x[i] for i in indices) <= 1, f"UnIntento_{credito}"

[ ] # === 4. Resolver ===
    modelo.solve()
```

⇒ 1

FUNCIÓN OBJETIVO


$$\sum \text{monto_cobrado} * x[i]$$

CREDIFIEL

OUTPUT

El modelo otorga los resultados en dos formatos distintos:

- Un csv que cuenta con todas las estrategias prioritarias para cada crédito individual.
- Se imprime en la consola las 5 mejores estrategias con todos sus datos.

 Estrategias óptimas para maximizar la cobranza:

	emisora	banco
5	BANORTE CLABE TRADICIONAL	BANORTE
12	BBVA CLABE INTERBANCARIA	BANORTE
52	Bancomer Tradicional ...	BBVA MEXICO
19	BBVA CLABE TRADICIONAL	BBVA MEXICO
55	SANTANDER CUENTA	SANTANDER
56	Santander Excepciones Cuenta ...	SANTANDER
3	BANAMEX TRADICIONAL REINTENTO	BANAMEX
7	BBVA CLABE EN LINEA	BBVA MEXICO
54	SANTANDER CLABE TRADICIONAL	SANTANDER
40	Banamex Reintentos ...	BANAMEX

	tipoenvio	num_creditos	total_cobrado	monto_promedio
5	TRADICIONAL	3723	3399358.24	913.069632
12	INTERBANCARIO	3866	2094208.71	541.699097
52	TRADICIONAL	3380	1997873.01	591.086689
19	TRADICIONAL	1793	1723836.28	961.425700
55	TRADICIONAL	2596	1522867.03	586.620582
56	TRADICIONAL	2726	1331772.01	488.544391
3	TRADICIONAL	2187	1331732.33	608.931107
7	EN LINEA	1935	1308053.63	675.996708
54	TRADICIONAL	1304	1185054.64	908.784233
40	REINTENTO	3213	789537.42	245.732157

PERMITE

Generar un plan de acción y dar claridad al operador de que estrategias priorizar para cada crédito.

CREDI FIEL

SUGERENCIAS

- Les sugerimos considerar una extensión del modelo actual hacia un enfoque estocástico, que permita tomar decisiones considerando la incertidumbre en tiempos de respuesta o montos cobrados.
- Contar con una base que se alimente en tiempo real facilitaría la toma de decisiones y permitiría aplicar modelos más avanzados y adaptativos.
- Vimos varios errores que parecen venir de la carga manual, como IDs que no coinciden entre tablas o columnas con nombres largos y diferentes, lo que complica mucho el análisis.
- También observamos que la misma variable en diferentes datasets tenían distinto formato.

¡GRACIAS!