

Construcción de un Entorno Analítico (DWH) y Cálculo del CLTV de Cliente

Ingeniería Matemática e Informática - Gestión de Datos

Alejandro Martínez Ronda

Índice

| | |
|--|----------|
| 1. Introducción | 3 |
| 2. Arquitectura y Modelo de Datos | 3 |
| 2.1. Modelo Entidad-Relación | 3 |
| 2.2. Asignación de Claves | 3 |
| 3. Proceso de Integración y Transformación de Datos | 4 |
| 3.1. Migración ETL | 4 |
| 4. Reducción de Dimensionalidad | 4 |
| 5. Modelo de Regresión Logística para CHARN | 4 |
| 5.1. Preparación y Transformación de Datos | 5 |
| 5.2. Ajuste y Evaluación del Modelo | 5 |
| 6. Cálculo del CLTV | 5 |
| 6.1. Cálculo del coeficiente de retención | 5 |
| 6.2. Cálculo del Valor Actualizado | 5 |
| 7. Imágenes y Visualizaciones | 6 |
| 7.1. Distribución del CLTV a 5 años | 6 |
| 7.2. Valor Acumulado por Año | 6 |
| 7.3. Segmentación de Clientes por Quintiles | 7 |
| 8. Conclusiones y Automatización del Proyecto | 7 |
| 9. Apéndices | 8 |
| 9.1. Código Completo y Recursos | 8 |
| 9.2. Estructura del proyecto | 8 |
| 9.2.1. Flujo de Trabajo | 8 |
| 9.2.2. Estructura de Directorios | 9 |

1. Introducción

Este proyecto tiene como objetivo calcular el **Customer Lifetime Value (CLTV)** mediante la integración y transformación de datos provenientes de 19 tablas de Azure SQL y su posterior migración a SQL Server Management Studio. Se emplean técnicas de reducción dimensional y regresión logística para estimar la retención del cliente, elemento clave en el cálculo del CLTV. El enfoque es eminentemente técnico, poniendo especial atención en la calidad de las transformaciones, el modelado de datos y la reproducibilidad del proceso mediante código en Python y consultas SQL.

2. Arquitectura y Modelo de Datos

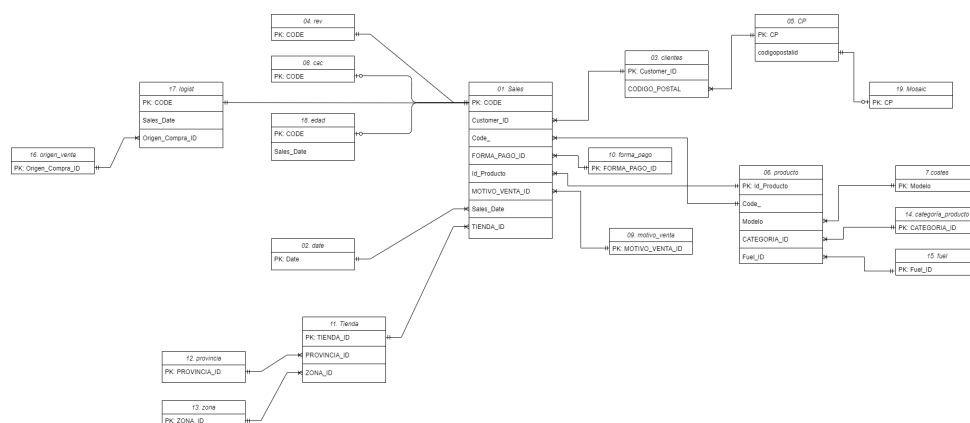
2.1. Modelo Entidad-Relación

El modelo de datos se basa en la integración de las 19 tablas dentro de las siguientes dimensiones:

- **Dim_customer:** Datos demográficos y de comportamiento de clientes.
- **Dim_geo:** Información geográfica de las tiendas.
- **Dim_product:** Detalles de productos y costos asociados.
- **Dim_t:** Dimensión tiempo.
- **Fact:** Registro de transacciones y métricas operativas.

Para construir este modelo, se realizaron consultas en **Azure Data Studio** que permitieron estudiar las relaciones y columnas de cada tabla, definiendo las claves primarias (PK) y foráneas (FK) necesarias. Se generó un diagrama ER en DrawIO, que ilustra la arquitectura del sistema.

Diagrama ER:



2.2. Asignación de Claves

La definición de claves primarias se llevó a cabo mediante consultas SQL en el entorno local de SQL Server Managment.

3. Proceso de Integración y Transformación de Datos

3.1. Migración ETL

Se diseñó un proceso ETL en Python para transferir datos desde Azure SQL a SQL Server Local. Este proceso implica:

- **Extracción:** Ejecución de archivos SQL (por ejemplo, `Dim_customer.sql`, `Dim_geo.sql`, etc.) en Azure SQL.
- **Transformación:** Conversión de tipos de datos (por ejemplo, de `FLOAT` a `INT`), tratamiento de valores nulos y normalización de datos.
- **Carga:** Creación de las tablas en SQL Server local y posterior inserción de registros.

4. Reducción de Dimensionalidad

Para optimizar el análisis y modelado, se realizó una reducción de dimensionalidad estructurada en torno al cliente (`Customer_ID` como clave primaria). Se calcularon métricas agregadas clave que permiten entender su comportamiento y rentabilidad en función de sus interacciones comerciales.

Entre las métricas más relevantes se encuentran:

- **CHARN:** Indicador de cancelación o retención del cliente siendo 1 si la última revisión fue hace más de 400 días y 0 al contrario.
- **Margen generado:** Se calcula a partir del precio de venta y el coste, evaluando la rentabilidad del cliente.
- **Suma de leads:** Número de interacciones en las que un cliente ha sido captado como posible comprador y ha realizado una conversión.
- **Interacción Precio-Unidad:** Relación entre el precio de venta y la antigüedad del vehículo adquirido, útil para analizar patrones de compra.

Se consolidaron estos cálculos en una estructura optimizada, asegurando que cada cliente tuviera un conjunto de métricas representativas. Esta transformación permite mejorar la eficiencia de los modelos predictivos y segmentaciones posteriores, facilitando la integración con el cálculo del CLTV.

5. Modelo de Regresión Logística para CHARN

Antes de proceder al cálculo del CLTV, se ajusta un modelo de regresión logística para predecir la variable **CHARN**. Esta variable se define a partir de la fidelidad del cliente (1 si han transcurrido más de 400 días desde la última revisión, 0 en caso contrario).

5.1. Preparación y Transformación de Datos

Se agrupan y transforman las variables relevantes:

- **Transformaciones:**

- Se aplica el logaritmo al PVP: $\log_PVP = \log_{10}(PVP)$.
- Se calcula la raíz cuadrada de la edad del vehículo: $\sqrt{Car_Age} = \sqrt{Car_Age}$.

- **Selección de Variables:** Se utilizan \log_PVP , $\sqrt{Car_Age}$, $Km_medio_por_revision$ y $PVP_x_Car_Age$ para predecir CHARN.

5.2. Ajuste y Evaluación del Modelo

El modelo se entrena dividiendo el dataset en entrenamiento y prueba, evaluando métricas como precisión, recall, f1-score y ROC AUC.

Los coeficientes obtenidos se almacenan en un CSV y en una tabla de SQL para su posterior uso en el cálculo del CLTV.

6. Cálculo del CLTV

Con el modelo de retención basado en la regresión logística, se procede a calcular el CLTV.

6.1. Cálculo del coeficiente de retención

La retención se estima mediante la siguiente fórmula:

$$\text{Retención} = 1 - \frac{1}{1 + \exp\left(-\left(\beta_0 + \beta_1 \cdot \log(PVP) + \beta_2 \cdot \sqrt{Car_Age}\right)\right)}$$

6.2. Cálculo del Valor Actualizado

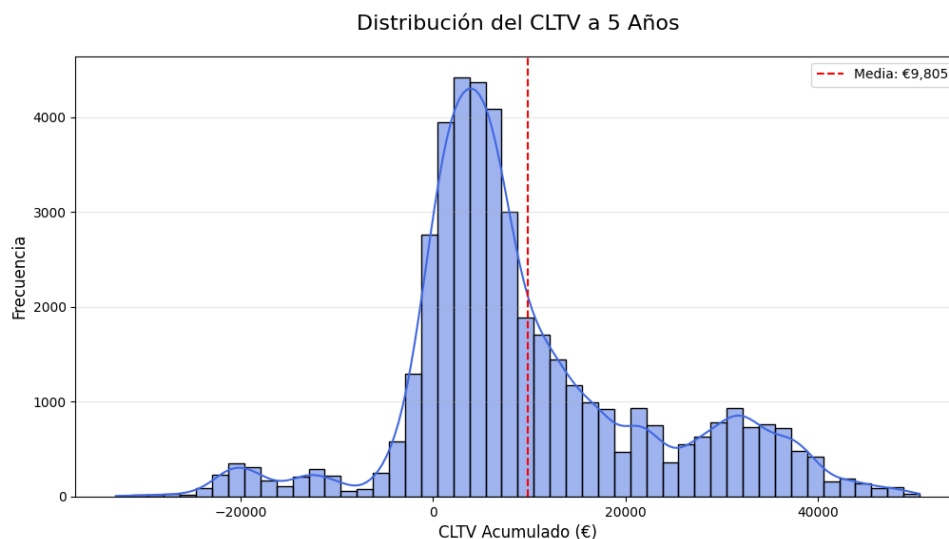
El CLTV se calcula acumulando el margen del cliente durante 5 años, descontado a una tasa del 7%:

$$CLTV = \sum_{t=1}^5 \frac{\text{Margen} \times P(Ret)_t}{(1+r)^t} \quad \text{donde } r = 7\%$$

7. Imágenes y Visualizaciones

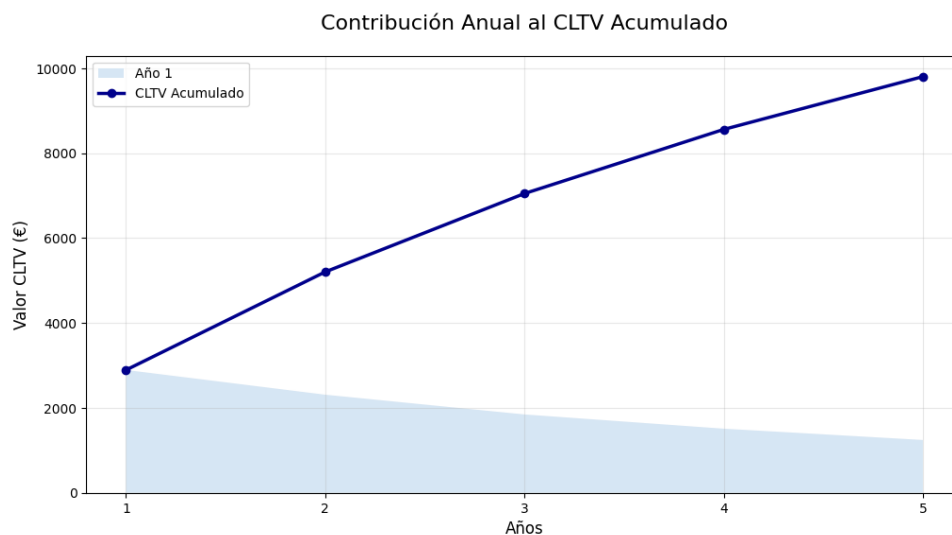
En esta sección se incluyen algunas de las visualizaciones facilitan la comprensión del proceso:

7.1. Distribución del CLTV a 5 años



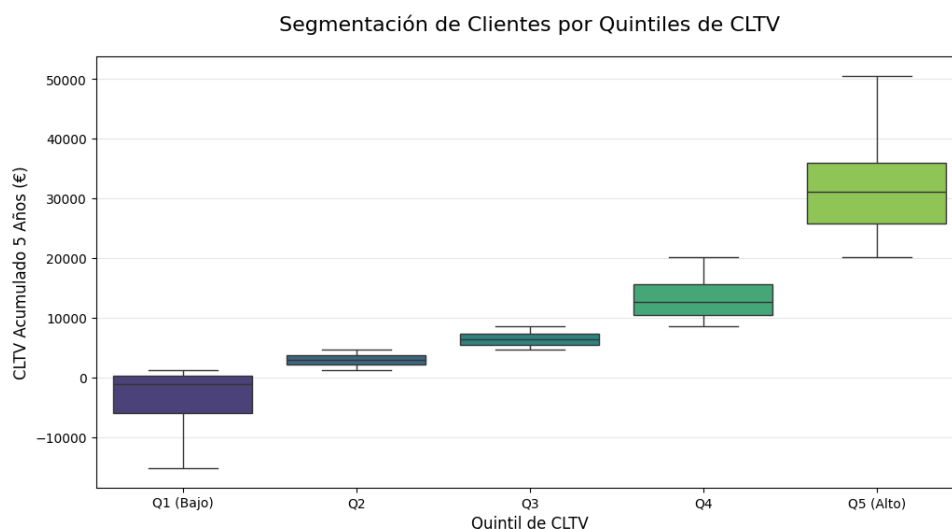
Distribución asimétrica del CLTV con mayoría de clientes entre 0-20k€ (media en 9,805€), presencia de valores negativos (hasta -20k€) y cola larga hacia valores altos (hasta 50k€).

7.2. Valor Acumulado por Año



Decaimiento progresivo de la contribución anual al CLTV (Año 1 ¿50 % del valor total), con crecimiento acumulativo por efecto del descuento (7 %).

7.3. Segmentación de Clientes por Quintiles



Segmentación no lineal: Q1 (pérdidas), Q2-Q3 (baja-media rentabilidad), Q4-Q5 (alto valor con salto significativo entre quintiles).

8. Conclusiones y Automatización del Proyecto

El proceso implementado permite obtener una estimación técnica robusta del CLTV mediante:

- Integración y transformación de datos heterogéneos.
- Reducción dimensional para focalizar el análisis en las variables de interés.
- Aplicación de un modelo de regresión logística para calcular el coeficiente de retención, elemento clave en la fórmula del CLTV.

Se ha desarrollado un sistema semiautomatizado que integra la ejecución de migraciones, transformaciones y el cálculo del CLTV. La automatización se respalda en scripts de Python y consultas SQL, y la organización del repositorio en GitHub facilita la reproducibilidad y mantenimiento del proyecto.

Serian necesarios pocos retoques para convertirlo en un proyecto automatizado al 100

9. Apéndices

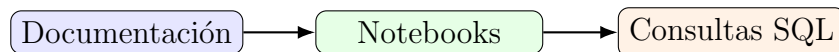
9.1. Código Completo y Recursos




Debido a restricciones de espacio en esta versión impresa, el código completo, scripts SQL, notebooks y diagramas se encuentran disponibles en el repositorio de GitHub:

<https://github.com/alejandromtnz/CLTV-Data-Integration>

9.2. Estructura del proyecto

9.2.1. Flujo de Trabajo



-  **Documentación:** Contiene especificaciones técnicas en pdfs o excel, outputs del modelo de regresión logística y el diagrama Entidad-Relación en png y drawio.
-  **Notebooks:** Implementación del ETL, regresión logística, visualizaciones del CLTV y verificación de la integridad del dato.
-  **SQL:** Consultas para el modelo dimensional (consultas de las 4 dimensiones y Fact), asignación de PKs, reducción de la dimensionalidad, cálculo CLTV y consulta para poder trabajar en PowerBI.

9.2.2. Estructura de Directorios

| |
|---|
| CLTV-Data-Integration/ |
| docs/ |
| ▷ data/ |
| ▷ • Datalake.xlsx # Archivo donde apreciar las columnas y las tablas |
| ▷ diagrams/ |
| ▷ • ER.drawio # Diagrama entidad-relación (DrawIO) |
| ▷ • ER.png # Exportación del diagrama entidad-relación (png) |
| ▷ model_outputs/ |
| ▷ • logistic_coefficients.csv # Coeficientes reultantes de la regresión logística |
| ▷ reports/ |
| ▷ • Caso_uso_DW.pdf # Caso de uso |
| ▷ • CLV_article.pdf # Artículo 1 |
| ▷ • CLV_article2.pdf # Artículo 2 |
| src/ |
| ▷ notebooks/ |
| ▷ • cham_logistic_regression.ipynb # Código regresión loogística |
| ▷ • CLTV_visualization.ipynb # Código visualizaciones CLTV |
| ▷ • environment_migration.ipynb # Código migraciones de Azure a SSMS |
| ▷ • verify_data_integrity.ipynb # Código verificar la no pérdida de datos |
| ▷ sql/ |
| ▷ CLTV/ |
| ▷ • SQLQuery_CLTV.sql # Query para la creación del CLTV |
| ▷ dimensional_model/ |
| ▷ • Dim_customer.sql |
| ▷ • Dim_geo.sql |
| ▷ • Dim_product.sql |
| ▷ • Dim_t.sql |
| ▷ • Fact.sql |
| ▷ dimensionality_reduction/ |
| ▷ • SQLQuery_Customer.sql # Query para la reducción de la dimensionalidad |
| ▷ PKs/ |
| ▷ • SQLQuery_PKs.sql # Query para la creación de PKs |
| ▷ PowerBI/ |
| ▷ • SQLQuery_PowerBI.sql # Query para la consulta necesaria para PowerBI |

Figura 1: Estructura jerárquica del proyecto con los archivos clave resaltados