Informe Final: Proceso ETL, Modelado Dimensional y Análisis CLTV

Frank Llonch Valsells
Universidad Alfonso X el Sabio

29 de marzo de 2025

1. ¿How? (Descripción General del Proceso)

En este informe se describe el flujo completo desde la extracción y consolidación de datos hasta el análisis del Customer Lifetime Value (CLTV).

1. Origen de Datos (Azure)

- Inicialmente, se disponía de 20 tablas en Azure que contenían información de diversas entidades.
- Se diseñó un modelo Entidad-Relación utilizando *draw.io* para identificar las tablas, sus claves primarias (PK) y foráneas (FK), y determinar cómo integrar la información.

2. Combinación en 5 Tablas Clave

- Las 20 tablas fueron analizadas y combinadas lógicamente en 5 tablas finales: 4 dimensiones (por ejemplo, `dim_customer`, `dim_geo`, `dim_time`, `dim_producto`) y 1 tabla de hechos (`fact_sales`).
- Se definieron estas 5 tablas en SQL (ver archivos `dim_customer.sql`, `dim_fact.sql`, etc.) y cada consulta resultante se descargó en formato JSON (por ejemplo, `dim_customer.json`, `dim_fact.json`, etc.).

3. Migración a Entorno Local (Visual Studio en Mac)

- Debido a que se trabajó en Mac, se optó por procesar los JSON localmente utilizando Visual Studio o VS Code.
- Con Python (véase `dimensionator.ipynb`) y librerías como **pandas**, se transformaron los datos (casting de fechas, limpieza, tipificación de columnas, etc.).

4. Carga en PostgreSQL (Modelo Dimensional)

- Usando **sqlalchemy**, se cargaron los datos transformados desde los JSON hacia PostgreSQL, creando así el modelo dimensional.
- Se generó además una tabla/vista centrada en el cliente (`customer_cltv`) para el análisis de retención y CLTV.

5. Análisis y CLTV

- Se desarrolló un modelo de machine learning (regresión logística) para estimar la probabilidad de churn.

- Con esa probabilidad se calculó la **retention_prob** y se aplicaron fórmulas para estimar el CLTV a 1 año y a 5 años.
- Se generaron visualizaciones interactivas (Plotly) y gráficas estáticas (Matplotlib) para extraer insights sobre los clientes.

2. Esquema del Movimiento de Datos (Data Flow)

El movimiento de los datos se puede resumir en el siguiente diagrama conceptual:

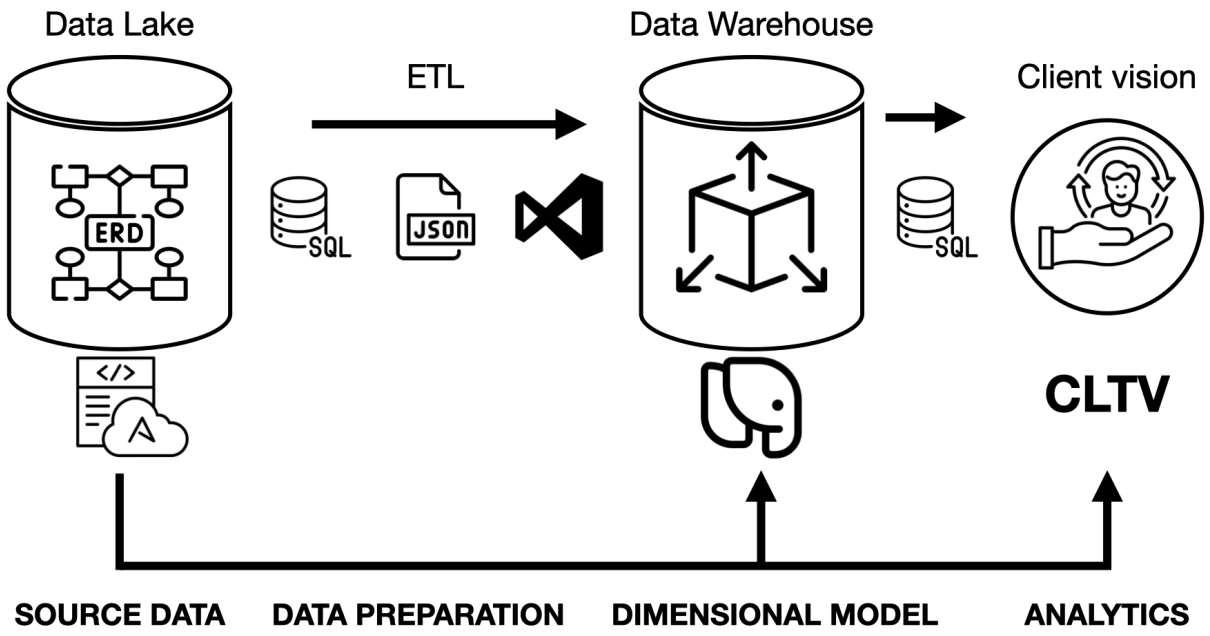

Figura 1: Diagrama del flujo de datos desde Azure hasta el modelo dimensional en PostgreSQL.

Descripción del flujo:

1. Se parte de las 20 tablas originales en Azure.
2. Se diseña un modelo E-R en *draw.io* y se crean consultas para consolidar la información en 5 tablas.
3. Los resultados se descargan en formato JSON.
4. En Visual Studio (Mac) se transforman y limpian los JSON utilizando Python.
5. Finalmente, se cargan los datos en PostgreSQL mediante SQLAlchemy, creando el modelo dimensional y la tabla de clientes (`customer_cltv`).

3. Diagrama del Modelo Entidad-Relación y Modelo Dimensional

3.1. Modelo Entidad-Relación (E-R)

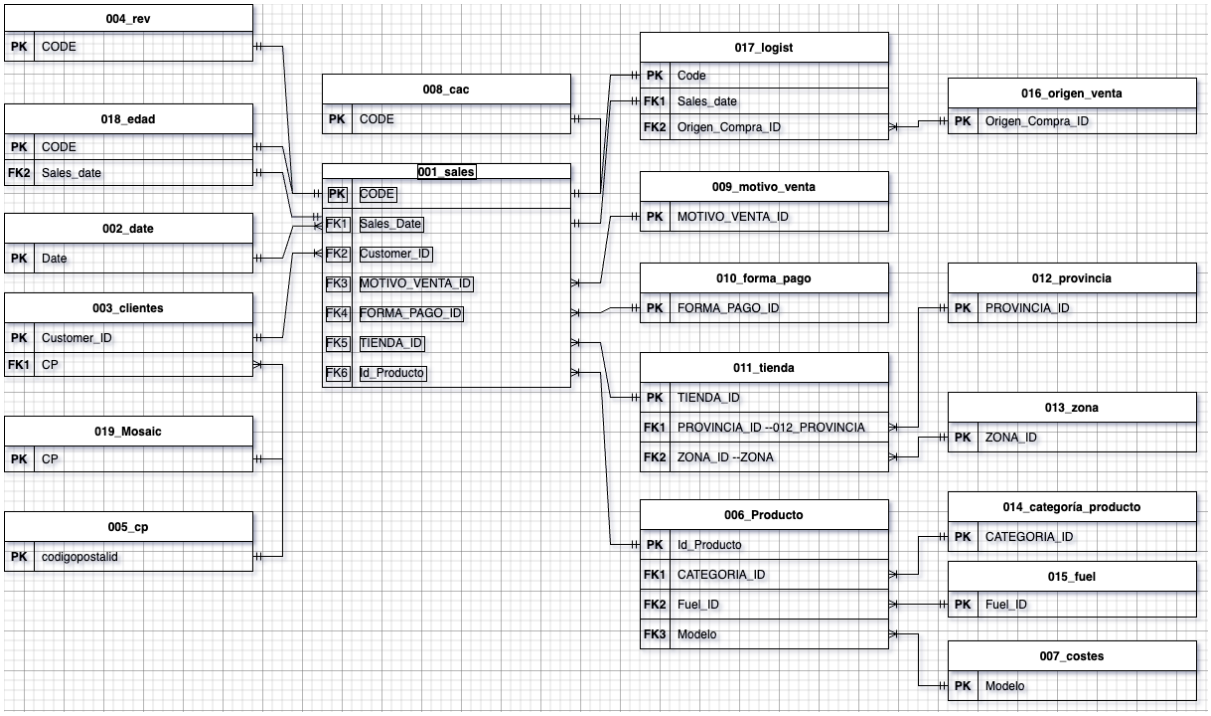

Figura 2: Diagrama E-R de las 20 tablas originales y su consolidación en 5 tablas clave.

3.2. Modelo Dimensional (Star Schema)

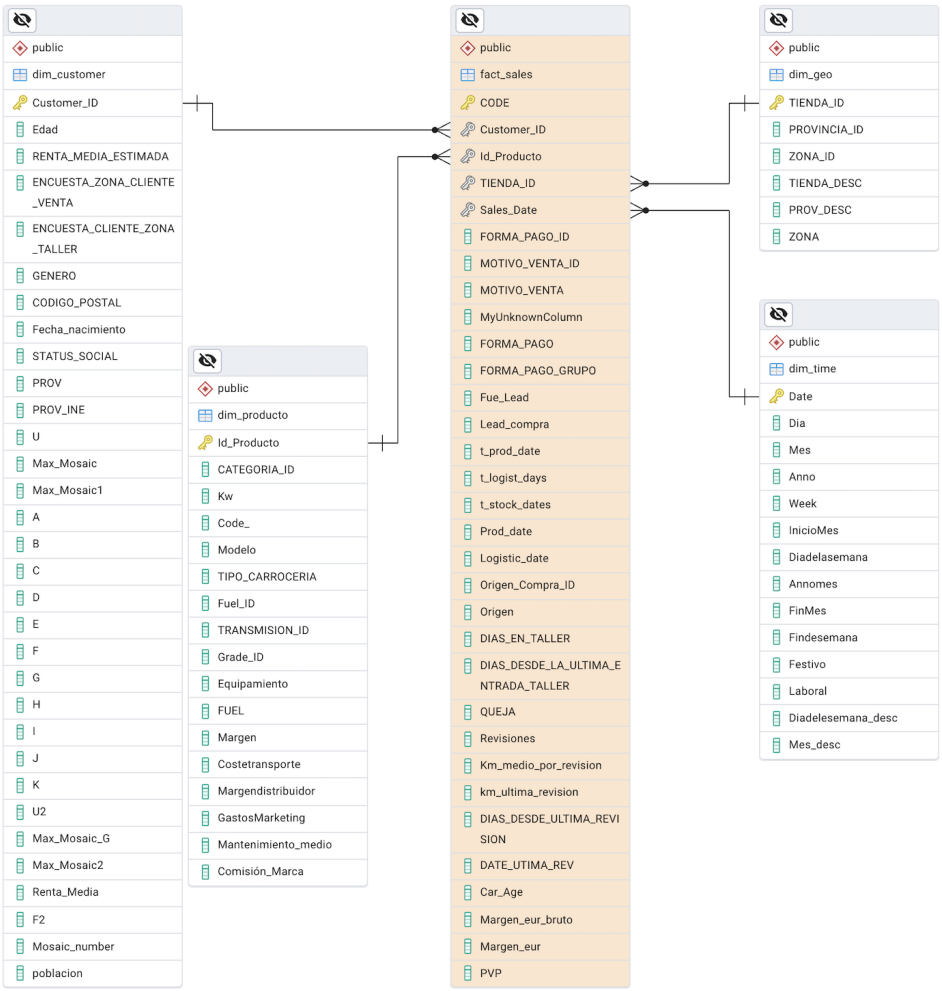

Figura 3: Modelo Dimensional en forma de esquema en estrella.

4. Análisis CLTV, Visión del Cliente y Visualizaciones

Visión Cliente

La visión centrada en el cliente se traduce en una evaluación integral que no solo considera las transacciones históricas, sino también el potencial futuro. La tabla `customer_cltv` permite segment clientes de alto valor (por ejemplo, en el segmento Gold) y diseñar estrategias de retención específicas, a la vez que se detectan clientes con alto riesgo de churn mediante el modelo de regresión logística.

Cálculo del CLTV

El **Customer Lifetime Value (CLTV)** se calcula mediante la siguiente fórmula:

$$CLTV = \sum_{t=1}^{n} \frac{B \times R(t)}{(1+i)^t}$$

donde:

- B es el beneficio obtenido por el cliente (por ejemplo, el promedio de margen por orden, avg_margin_per_order).
- $R(t)$ es la tasa de retención estimada para el año t (derivada del modelo de churn).
- i es la tasa de descuento o riesgo.
- n es el horizonte temporal (por ejemplo, 1 o 5 años).

Cálculo del Churn y Retention

El modelo de **regresión logística** se emplea para estimar la probabilidad de churn:

$$P(Y=1) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\cdots+\beta_k X_k)}}$$

donde $Y=1$ indica que el cliente ha abandonado. La tasa de retención se define como:

$$R = 1 - P(Y=1)$$

Cálculo del Churn Flag

Se define el **churn flag** en función de la variable `DIAS_DESDE_ULTIMA_REVISION`:

$$\text{churn\_flag} = \begin{cases} 1, & \text{si } \texttt{DIAS\_DESDE\_ULTIMA\_REVISION} > 400, \\ 0, & \text{en otro caso.} \end{cases}$$

Visualizaciones e Insights

A continuación se muestran espacios reservados para los gráficos generados (ver carpetas `html/` y `matplotlib_graphs/`):

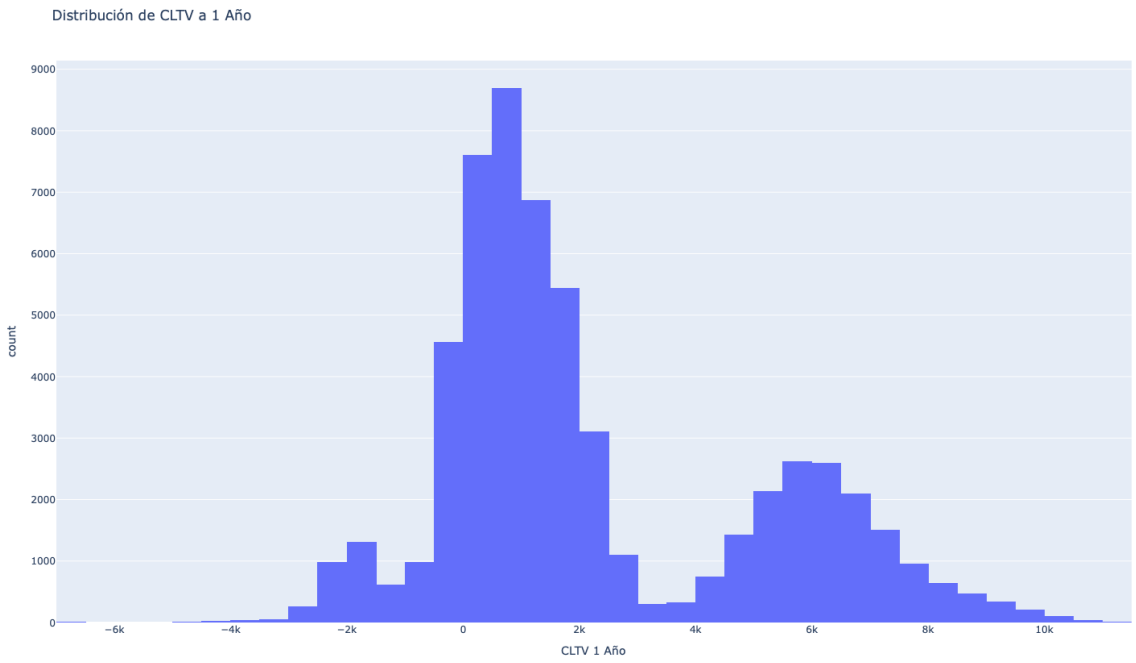

Figura 4: Histograma de CLTV a 1 Año.

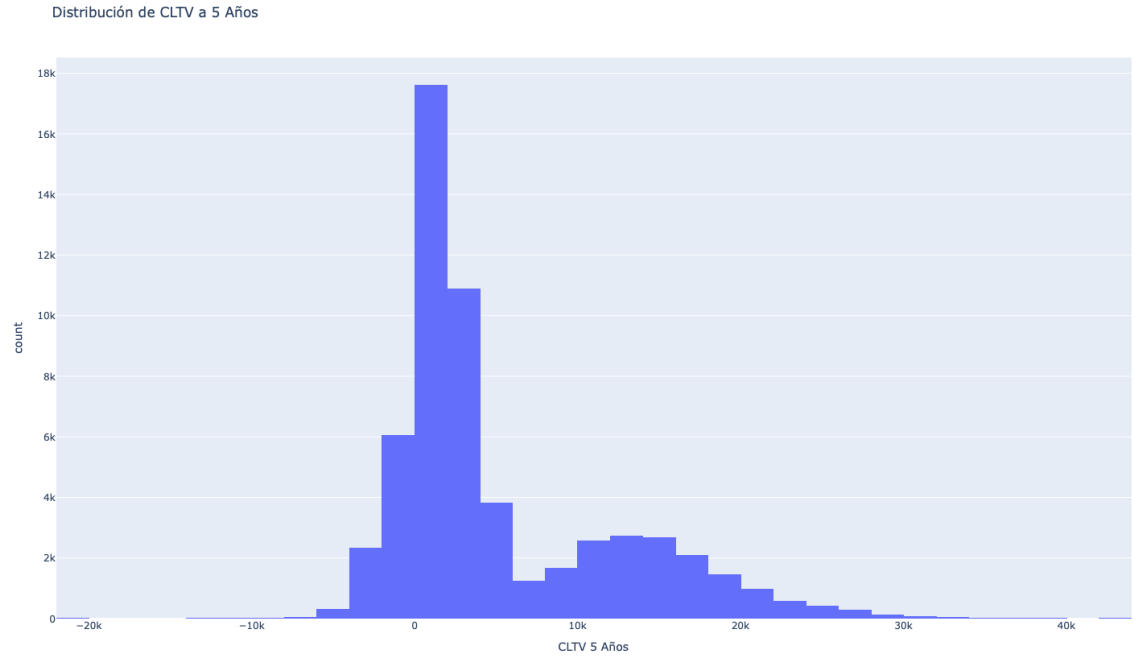

Figura 5: Histograma de CLTV a 5 Años.

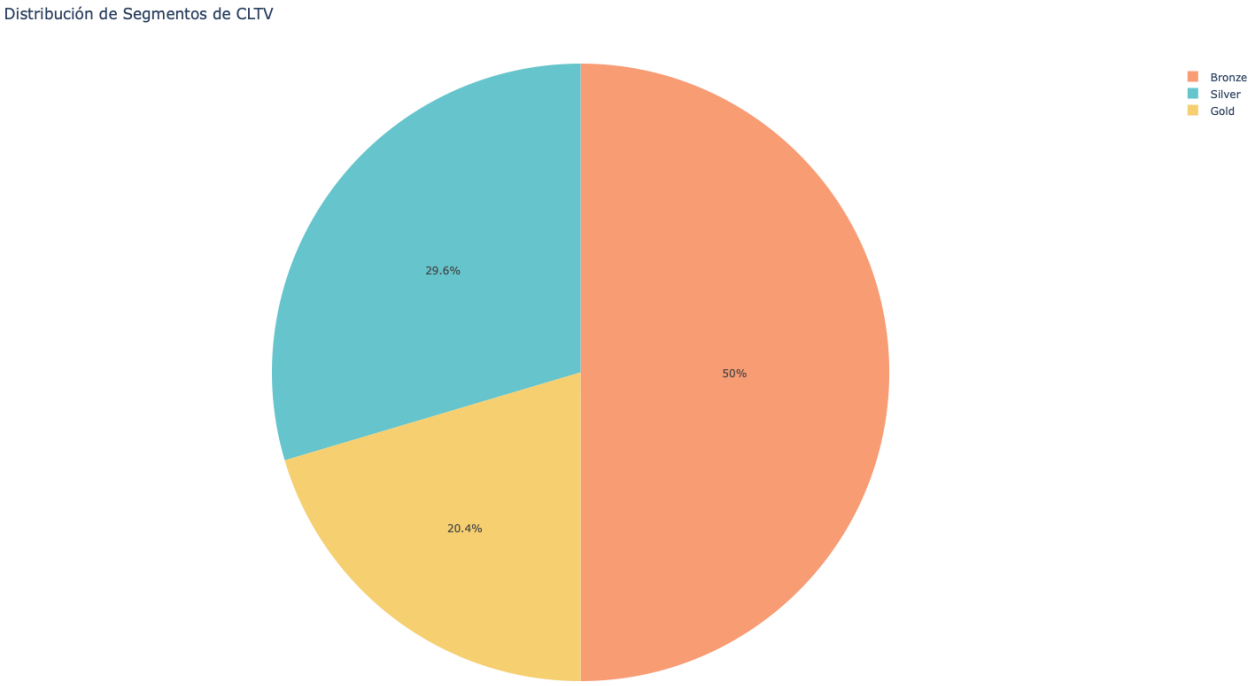

Figura 6: Pie graph: Segmentación en bronze, silver y Gold

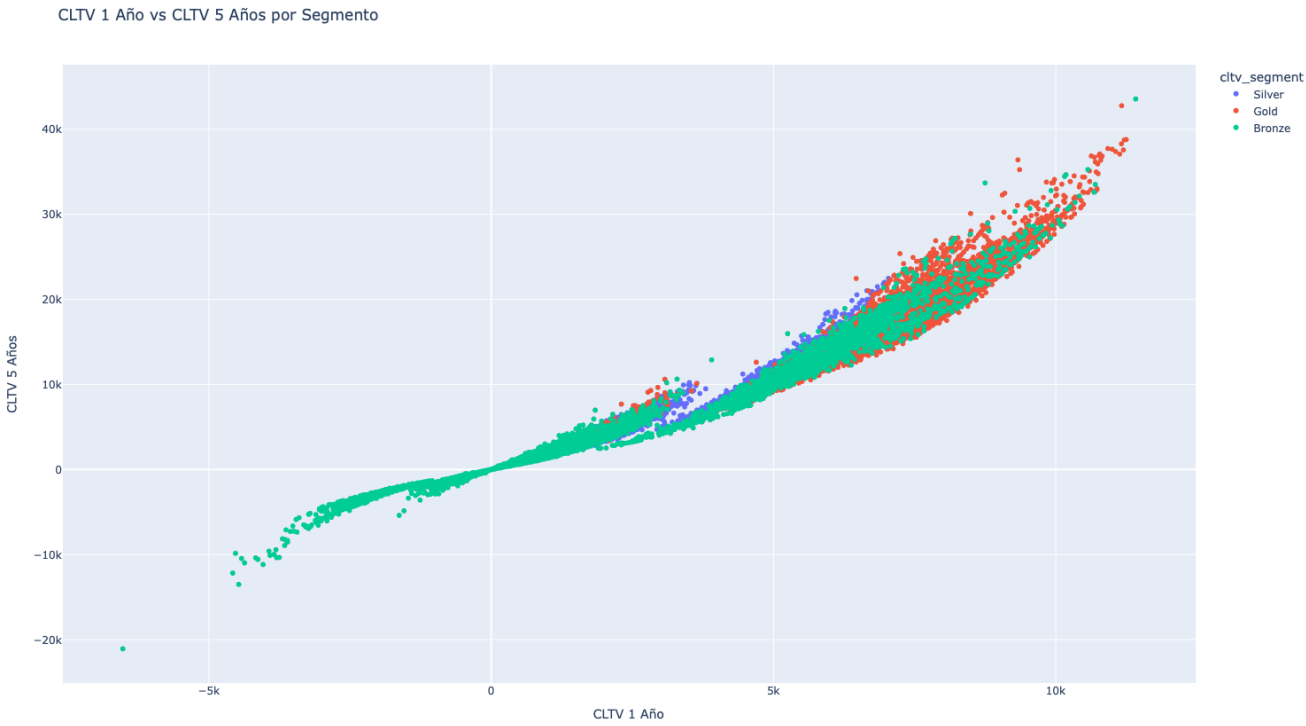

Figura 7: Scatter Plot: CLTV 1 Año vs CLTV 5 Años por Segmento.

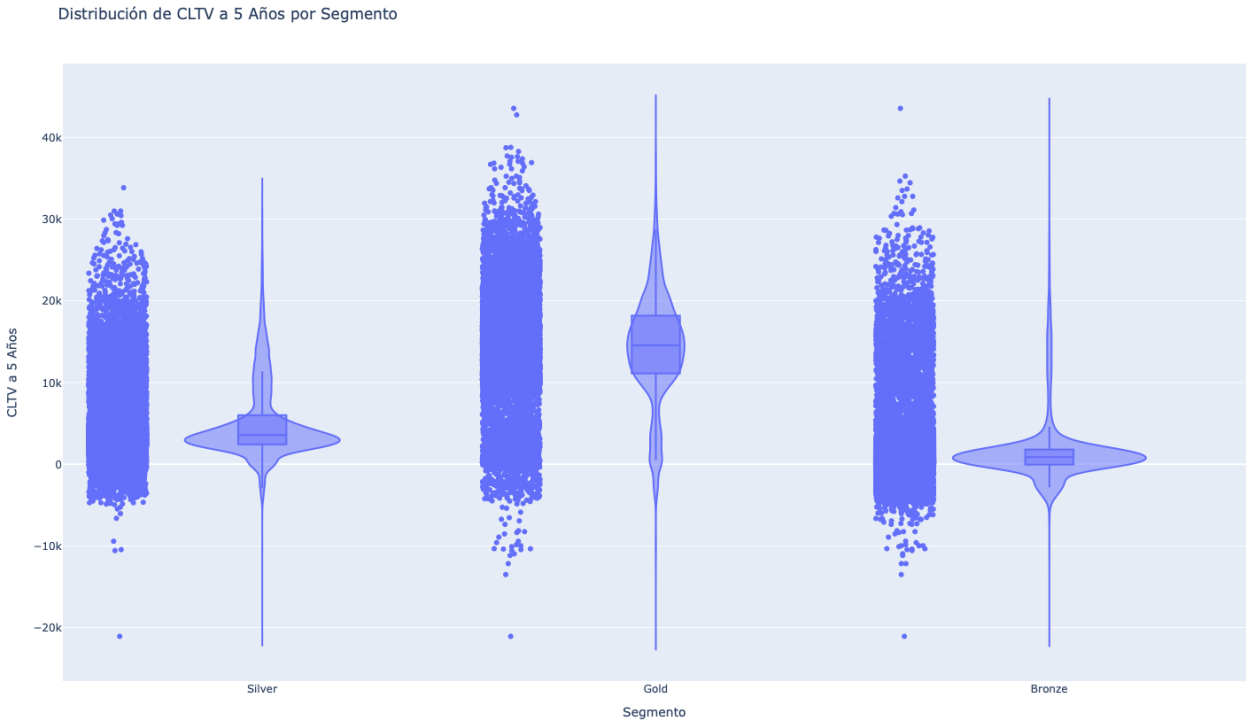

Figura 8: Diagrama Violin: CLTV 5 años por segmentos

5. Conclusiones y Puntos de Mejora

Conclusiones Generales

Se ha construido un pipeline integral que abarca:

1. La extracción de 20 tablas en Azure, seguido del diseño de un modelo E-R
2. La consolidación de la información en 5 tablas clave, descargadas en formato JSON.
3. La transformación y limpieza de los datos mediante Python, y su posterior carga en un modelo dimensional en PostgreSQL.
4. La creación de una tabla centrada en el cliente (`customer_cltv`) para el análisis de retención y cálculo de CLTV.
5. La aplicación de un modelo de regresión logística para estimar la probabilidad de churn y, a partir de ella, calcular la tasa de retención y el CLTV a diferentes horizontes temporales.

Puntos de Mejora

- **Integración en Azure:** Idealmente, todo el proceso ETL y la creación del modelo dimensional se debería realizar directamente en un entorno Azure, evitando la descarga manual de JSON y facilitando la escalabilidad y el mantenimiento.
- **Automatización del Pipeline:** Se podría orquestar el pipeline utilizando herramientas como **Apache Airflow** o **Azure Data Factory** para ejecutar procesos de manera periódica y sin intervención manual.
- **Optimización del Modelo de Churn:** Explorar algoritmos más complejos (Random Forest, XGBoost) y técnicas de balanceo (oversampling/undersampling) para mejorar el desempeño (precision y recall) del modelo de churn.
- **Dashboard Interactivo:** La creación de un dashboard profesional en **Power BI** o **Tableau** permitiría la visualización dinámica de los insights, facilitando la toma de decisiones.
- **Validación y Refinamiento:** Continuar evaluando y refinando tanto el pipeline ETL como los modelos de predicción para adaptarse a cambios en el comportamiento de los clientes y en el entorno de datos.

Referencias y Anexos

- **SQL:** `dim_customer.sql`, `dim_fact.sql`, `dim_geo.sql`, `dim_producto.sql`, `dim_time.sql`.
- **JSON:** `dim_customer.json`, `dim_fact.json`, `dim_geo.json`, `dim_producto.json`, `dim_time.json` (descargados desde Azure tras combinar las 20 tablas en 5).
- **Python Notebooks:** `dimensionator.ipynb` (contiene el proceso ETL, la creación de `customer_cltv`, el cálculo del CLTV y las visualizaciones).
- **Carpetas de Resultados:**
  - `html/`: Gráficos interactivos en HTML generados con Plotly.
  - `matplotlib_graphs/`: Gráficos en PNG generados con Matplotlib/Seaborn.

Fin del Informe