Proyecto ETL 3

Santiago Gomez Castro

Juan Carlos Quintero

MIguel Angel Ruales

Docente: Javier Alejandro Vergara Zorrilla

Universidad Autónoma de Occidente

Facultad de Ingeniería

Santiago de Cali

2024

Contexto

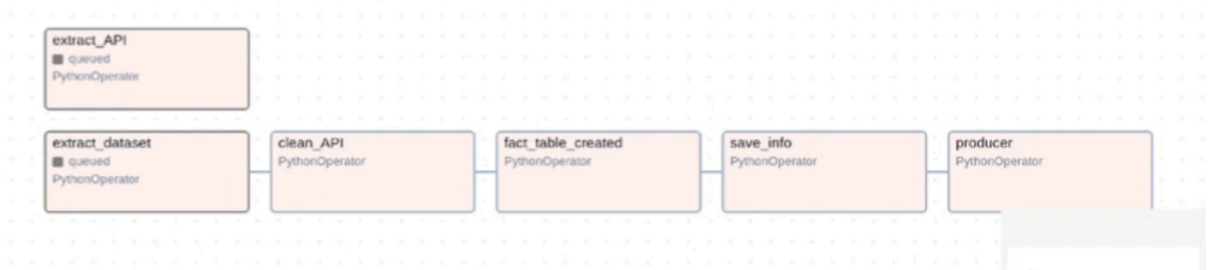
Mediante el uso de airflow se busca realizar la extracción de datos de nuestra base de datos Postgresql y de la API, procedería una serie de limpieza y transformación de la información para obtener nuestra tablas de hechos y dimensionales, para el último realizar una transmisión streaming de los datos a un dashboard actualizable donde se mostrará la información extraída, también se implemente un sistema de validación de datos para la extracción correcta.

Herramientas usadas

- **Python:** Se usó python para la creación de scripts de subida de datos a la base de dato y Dags para el funcionamiento de Airflow
- **Jupyter:** Se emplean Notebooks de jupyter para el EDA de ambos dataset, donde se limpia, transforma la información y a su vez se crean gráficas para entender más sencillamente la información.
- **Ubuntu:** Una máquina virtual Ubuntu para poder ejecutar el proyecto en un ambiente linux ya que Airflow lo requiere para su correcto funcionamiento.
- **Apache-Ariflow:** Gestor de flujos de datos para la modificación, transformación y carga de los datos de forma secuencial.
- **Poetry:** Ambiente de desarrollo en Python para la gestión de las librerías requeridas.
- **Git y Github:** Gestores de versión de código para guardar y compartir el proyecto.
- **Looker:** Las visualizaciones principales se crearon usando PowerBI
- **SQLalchemy:** Esta librería nos permite la conexión a la base de datos para la obtención de los datos y luego actualizarlos después de haber pasado por el EDA.
- **Pandas:** Librería para el análisis de los datos y su manipulación.
- **Dontev:** Libreria para acceder a las credenciales de la base de datos y no exponerlas en el código.
- **PostgreSQL:** Base de datos relacional que nos facilitara el guardado y gestión de los datos.
- **Ngrok:**Se emplea Ngrok para crear un sub túnel para exponer nuestra base de datos local a la máquina Ubuntu.
- **Apache-kafka:** Nos permite hacer streaming de datos para el envío de los datos al Dashboard en looker
- **Docker:** Contenedores Docker para Kafka y ZooKeeper

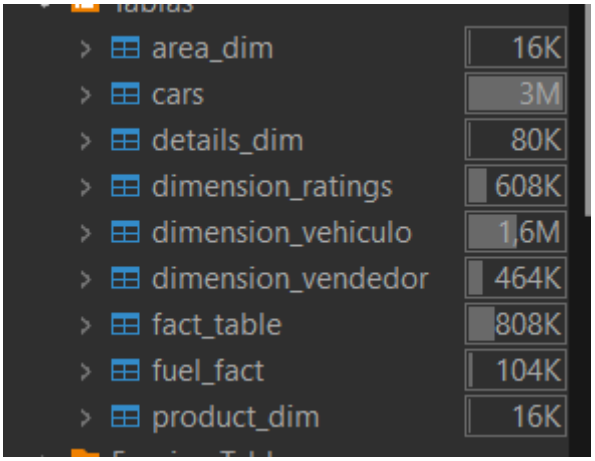
Airflow

Estos son los Dags usados en airflow para todo el proceso de ETL, a continuación se realizar una explicación de la funcionalidad de este workflow.



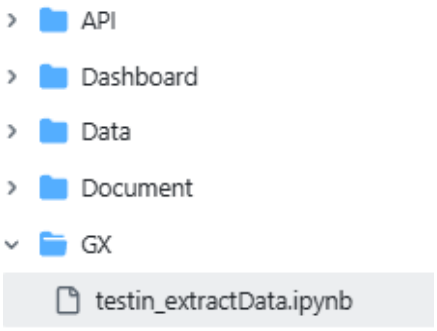
El workflow empieza con la extracción de los datos de la base de datos y la API, esta información se guarda en Dataframe que se envía en formato JSON al Dag siguiente, el Dag siguiente es “clean_api” donde se limpiaria la información de la API para poder ser manipulada, “fact_table_created” crea las tablas dimensionales y tablas de hechos que son guardados en la base de datos en el Dag “save_info”, por último con el Dag “producer” se realiza una transmisión de los datos al Dashboard en looker.

extract_API
extract_dataset >> clean_api >>fact_table_created >> save_info >> producer
Datos ya guardados en la base de datos:



Validación de Datos con Great Expectations

Para asegurar la calidad de los datos recuperados desde la API (extract_API), se utilizó la librería Great Expectations. El archivo para las validaciones es un cuaderno de jupyter dentro de una carpeta “GX”:



Este proceso garantiza que los datos sean adecuados para su posterior uso, validando las siguientes condiciones clave:

- **Estructura de columnas**
  Validar que las columnas obtenidas sean exactamente iguales a las columnas esperadas. Esto asegura consistencia en el esquema de los datos antes de continuar con el pipeline.

- **Tipos de datos**
  Verificar que cada columna tenga el tipo de dato adecuado. Este paso es crítico, ya que los datos aún no han sido transformados y podrían contener formatos no estandarizados.

- **Datos faltantes**
  Validar que ninguna columna contenga valores faltantes, excepto la columna `value`. Dado que el proceso ocurre antes de la limpieza, esta columna puede incluir algunos datos nulos.

Preparación del Proyecto

Antes de ejecutar las validaciones, es necesario iniciar un proyecto de **Great Expectations**. Para ello, basta con ejecutar el siguiente comando en la consola:

```
great_expectations init
```

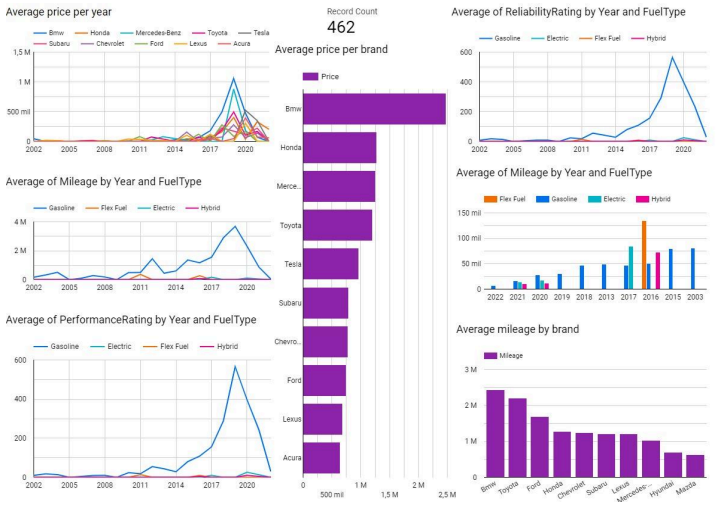
Ejecución de las Validaciones

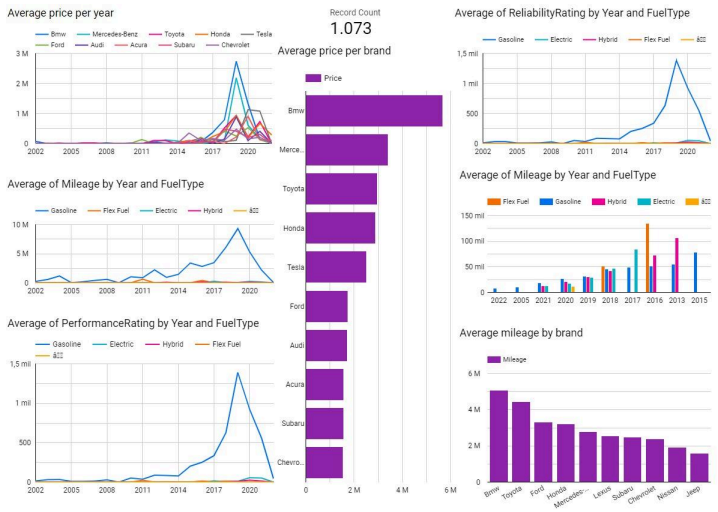
Al ejecutar el código del cuaderno correspondiente, se procesarán los datos y se generarán logs detallados de las expectativas definidas. Además, se mostrará un porcentaje que indica el nivel de éxito de las validaciones realizadas:

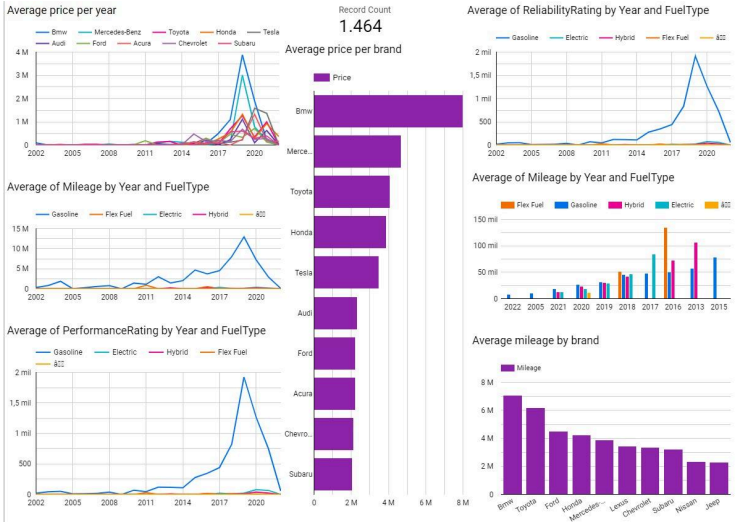
```
        "run_id" : {
            "run_name": null,
            "run_time": "2024-11-14T11:37:56.860088-05:00"
        },
        "batch_kwargs": {
            "ge_batch_id": "cd8516be-a2a6-11ef-9586-a81e84abb5fb"
        },
        "batch_markers": {},
        "batch_parameters": {},
        "validation_time": "20241114T163756.860088Z",
        "expectation_suite_meta": {
            "great_expectations_version": "0.18.8"
        }
    }
}
Porcentaje de expectativas exitosas: 100.00%
```


Dashboard

Se utilizó kafka para hacer un dashboard a tiempo real con looker, se crearon dos scripts: producer.py y consumer.py los cuales se encargan del streaming de los datos, el producer.py envía al consumer una fila cada cierto tiempo y el consumer se encarga de irlos subiendo a la base de datos postgresql progresivamente.
A continuación podemos ver el dashboard en 3 momentos distintos del streaming de datos.







Interpretaciones:

Average price per year

Este gráfico muestra la evolución del precio promedio de los vehículos por marca a lo largo de los años. Se observan picos significativos entre 2017 y 2020, indicando posibles eventos económicos, innovaciones tecnológicas o cambios en las políticas de precios.

Average price per brand

Este gráfico de barras refleja el precio promedio por marca. BMW aparece consistentemente como la marca con los precios más altos, lo que sugiere un enfoque en vehículos de lujo.

Average of Mileage by Year and FuelType

Este gráfico ilustra el kilometraje promedio por tipo de combustible y año. Los vehículos a gasolina dominan, especialmente en los años recientes. Se observa una tendencia creciente hasta 2017, seguida por una disminución.

Average mileage by brand

Se representa el kilometraje promedio por marca, donde BMW, Toyota y Ford destacan como las marcas con los mayores promedios, lo que podría estar relacionado con la calidad de fabricación o el uso esperado de sus vehículos.

Average of PerformanceRating by Year and FuelType

Este gráfico muestra la calificación de rendimiento promedio por año y tipo de combustible. Los vehículos a gasolina destacan, pero los eléctricos y otros tipos emergen en menor proporción.

Average of ReliabilityRating by Year and FuelType

Este gráfico presenta el índice de confiabilidad promedio. Se observa un aumento hasta 2020, especialmente en los vehículos a gasolina, seguido por una disminución, probablemente debido al envejecimiento del parque automotriz.

Conclusiones:

Demanda de vehículos premium:
Las marcas premium como BMW lideran en precios y rendimiento, lo que indica una demanda sólida en este segmento, posiblemente impulsada por consumidores que priorizan calidad y estatus.

Dominancia de vehículos a gasolina:
A pesar de los avances en tecnología eléctrica, los vehículos a gasolina siguen siendo dominantes en términos de kilometraje, rendimiento y confiabilidad.

Tendencias futuras:
Si bien los eléctricos están en crecimiento, es necesario más tiempo para que compitan en igualdad de condiciones con los vehículos a gasolina, especialmente en términos de kilometraje y confiabilidad.

Evolución del mercado:
El mercado automotriz está en un periodo de transición, donde los vehículos híbridos y eléctricos están comenzando a ganar terreno. Esto se alinea con tendencias globales hacia la sostenibilidad.