Detección de fraudes en siniestros de ARL: Implementación de técnicas de Machine Learning en Positiva Compañía de Seguros

Martínez H. Andrés F
Facultad de Inegniería y Ciencias Básicas
Universidad Central
Maestría en Analítica de Datos
Curso de Automatización e integración de datos
Bogotá, Colombia
amartinezh3@ucentral.edu.co

May 24, 2024

Contents

1 Introducción

Dentro de la economía colombiana, el sector asegurador es uno de los más fuertes. Durante el 2023, esta industria emitió un total de $60.6 billones pesos, representando un crecimiento del 7% con respecto al 2022 (La República, 2024). Esto equivale a un crecimiento según el indicador de penetración (calculado por la diferencia entre el valor consolidado de las primas y el PIB calculado del 1% para el 2023), ya que pasó del 3.24% al 3.42%. (Portafolio, 2024).

Por otra parte, los eventos que fueron atendidos por las aseguradoras también aumentaron en un 3%, lo cual representó $21.89 billones de pesos. (Portafolio, 2024). La siniestralidad más alta, se presenta en los seguros de vida (Gráfico 1) del cuál se pagaron cerca de $2.471 mil millones de pesos.

Dentro del análisis de la ocurrencia de estos siniestros, es importante tener en cuenta el factor fraude en la industria, que permanece muy latente dentro de los valores pagados. El sector asegurador ha implementado diferentes acciones durante el 2022 para prevenir estos fraudes y así se pudieron controlar cerca de 41.000 posibles eventos, cuyo pago hubiese significado entregar cerca de $202.000 millones en reclamaciones ilegales (Fasecolda, 2024).

Sin embargo, el riesgo de fraude puede haberse materializado en algunos casos, aún con los controles definidos para evitarlo. Los seguros más vulnerables en este sentido resultan siendo el SOAT, riesgos laborales, autos, sustracción y salud. La mayor parte de estos sucesos corresponden a eventos que no sucedieron. Para estos casos las estrategias más utilizadas fueron la simulación del siniestro, el uso de información falsa o adulterada, los cobros dobles o el uso de pólizas que están a nombre de terceros (Fasecolda, 2024).

2 Características del proyecto de investigación que hace uso de Integración y Automatización de Datos para IA

2.1 Titulo del proyecto de investigación

Detección de fraudes en siniestros de ARL: Implementación de técnicas de Machine Learning en Positiva Compañía de Seguros.

2.2 Objetivo general

Diseñar un sistema automatizado que utilice técnicas de Machine Learning y análisis estadístico supervisado para predecir, identificar, y analizar posibles fraudes en los siniestros de las pólizas de ARL de Positiva Compañía de Seguros.

2.2.1 Objetivos especificos

- Recolectar datos históricos de reclamaciones y transacciones en siniestros de ARL gestionados por Positiva Compañía de Seguros durante los últimos

cinco trimestres (2023-2024), con el fin de identificar patrones y tendencias indicativas de fraude.

- Construir una base de datos relacional para la captura, almacenamiento, tratamiento y visualización de los registros recolectados, utilizando herramientas que permitan la gestión de datos en tiempo real.
- Establecer un flujo de trabajo integrado que conecte las herramientas utilizadas para la recolección, organización, tratamiento, análisis y visualización de los datos requeridos para predecir posibles fraudes.
- Diseñar e implementar algoritmos y modelos de análisis de datos, incluyendo técnicas estadísticas supervisadas y de machine learning, para detectar los principales factores de riesgo de fraude en las pólizas de ARL.
- Desarrollar un tablero de control automátizado que presente de manera clara y concisa los resultados del análisis de riesgo de fraude, destacando los factores de riesgo identificados, y proporcionando herramientas interactivas para el análisis detallado de los datos.

2.3 Alcance

Positiva Compañía de Seguros S.A. es una empresa colombiana dedicada a ofrecer seguros a personas naturales y jurídicas, que cuenta con más de 60 años de experiencia en el mercado, cuando se constituyó Seguros Tequendama de Vida, la cual fue adquirida en 1995 por La Previsora S.A. Tras la integración de operaciones con ARP Seguro Social en 2008, ambas empresas formaron la actual compañía. Actualmente, Positiva cuenta con una base de 6 millones de personas aseguradas (equivalente al 8% del mercado) en toda Colombia, a través de productos que protegen riesgos laborales, salud, vida, rentas, etc.

El presente estudio pretende, tomando como referencia una base de datos ofrecida por la empresa que contiene los registros de las pólizas de ARL vendidas por la compañía y que fueron cobradas por siniestros (riesgos materializados) durante los cuatro trimestres del 2023 y el primero de 2024 en Colombia, identificar los diferentes patrones que se presentan en los eventos de fraude en el cobro de estas pólizas. De esta manera, se espera generar una herramienta que ayude a la compañía a identificar y generar acciones para mitigar los riesgos asociados con los cobros ilegales de estos siniestros.

2.4 Pregunta de investigación

¿Cuáles son los principales agentes identificados en los casos de fraude externo reportados por Positiva Compañía de Seguros durante el periodo 2023-2024, y cuáles son los factores de riesgo asociados que deben ser considerados en el diseño e implementación de medidas preventivas y de detección de fraudes en la compañía?

2.5 Hipótesis

A partir del uso de los datos obtenidos por Positiva Compañía de Seguros derivados de eventos fraudulentos ocurridos con el pago de sus pólizas, se pueden generar herramientas que permitan determinar las correlaciones existentes entre los factores de ocurrencia y muestren estos resultados en un dashboard en tiempo real, junto con algunos indicadores generales de control. Con esto, se busca entregar un instrumento que apoye análisis más profundos, los cuales permitirán tomar decisiones que mitiguen la materialización de estos riesgos.

3 Reflexiones sobre el origen de datos e información

Con base en el enfoque del presente estudio, y para dar claridad en la obtención y manejo de los datos a utilizar, es necesario establecer:

3.1 ¿Cuál es el origen de los datos e información?

El presente estudio se realiza tomando como base un dataset aportado por Positiva Compañía de Seguros S.A., con fines educativos y con compromiso de manejarla conforme a los requisitos exigidos por la ley para tal fin. Esta base de datos contiene los registros de las pólizas activas que presentaron alguna reclamación durante el periodo 2023-2024, y para su manejo, se han identificado 42 variables, de las cuales 28 son cualitativas y 14 son cuantitativas.

En la primera fase del proceso, en la cual se estandarizará la base de datos y se eliminarán los datos que no se utilizarán, quedarán 39 variables.

3.2 ¿Cuáles son las consideraciones legales o éticas del uso de la información?

Es importante establecer el marco legal bajo el cual se realizará el presente estudio, pues este estará bajo los lineamientos establecidos para la manipulación y uso de los datos recibidos por parte de la compañía. En principio, todo el proceso estará en función al cumplimiento de las siguientes normas:

Ley 1581 de 2012: Esta ley establece el marco general para la protección de datos en Colombia. Regula la recolección, almacenamiento, uso, circulación y supresión de datos personales, y garantiza los derechos de las personas sobre su información personal.

Decreto 1377 de 2013: Este decreto establece disposiciones adicionales a la Ley 1581 de 2012. Define aspectos como el registro de las bases de datos, los procedimientos para el ejercicio de los derechos de los titulares de la información, y las responsabilidades de los encargados y responsables del tratamiento de los datos.

Al mismo tiempo, es importante aclarar que, para garantizar el cumplimiento de estas leyes y el acceso a la documentación de sus procesos, los resultados obtenidos serán compartidos exclusivamente con la compañía y todos los datos fueran anonimizados, esto a fin de proteger la información de sus afiliados.

3.3 ¿Cuáles son los retos de la información y los datos que utilizara en Integración y Automatización de Datos para IA?

La base de datos a utilizar contiene una mezcla de 43 variables, entre variables cuantitativas y cualitativas, que describen las pólizas reclamadas por siniestros ocurridos durante un periodo de tiempo establecido. Esta base de datos es alimentada de forma continua cada trimestre (3 meses) por la misma compañía,

esto con el fin de asegurar una trazabilidad de los riesgos materializados y ayudar a establecer su afectación a los indicadores de las pólizas de ARL.

El reto en este proyecto es diseñar un tablero de control que se actualice de forma constante, e integre y analice los datos que se vayan generando en tiempo real. De esta manera, se asegura la disponibilidad de los principales indicadores de riesgo actualizados en todo momento y servirá como herramienta en el diseño de estrategias de prevención de estos riesgos.

3.4 ¿Qué espera de la utilización de la Integración y la Automatización de Datos para IA en su proyecto?

Con la realización del presente estudio, se espera generar una herramienta dinámica que la empresa Positiva Compañía de Seguros pueda utilizar para la estructuración de sus procesos de prevención de fraude en el cobro de pólizas de ARL de sus afiliados en un futuro. De esta manera, se aplicarán conocimientos y técnicas de tratamiento de bases de datos para identificar los principales ecosistemas de fraude que se presentan actualmente en esta industria y los diferentes agentes que participan en él, fungiendo como un punto de partida en el uso de análisis posteriores más profundos.

4 Diseño de integración y Automatización de Datos para IA

Como primer paso en el desarrollo del presente estudio, y con el fin de conocer el flujo de trabajo que se pretende construir, se genera un diagrama de integración y automatización de datos para IA. Con este, se pretende entregar al lector un panorama global de este proyecto, que le permitirá identificar de manera orgánica la integración de las diferentes herramientas utilizadas en el diseño del proyecto.

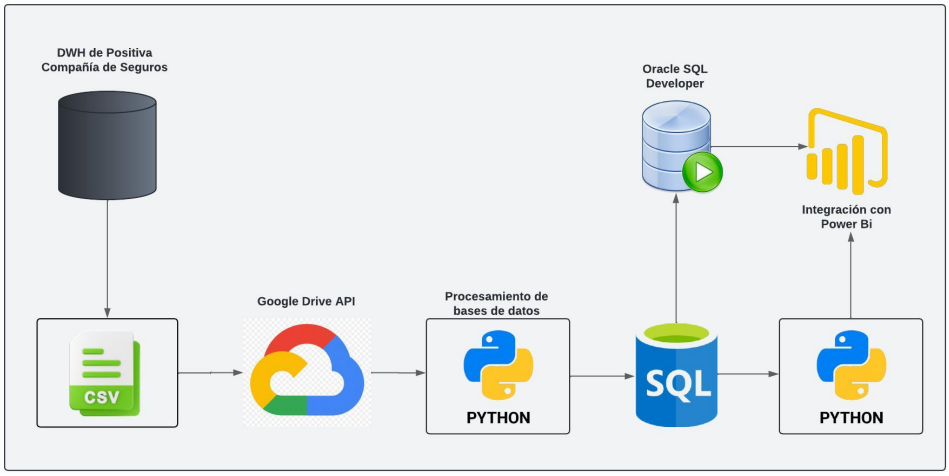


Figure 1: Diseño de integración y automatización de datos para IA. (Elaboración propia)

4.1 Fuente de datos

- Datos entregados por Positiva Compañía de Seguros como fuente primaria y custodiados en su DWH (Bodega de datos). Esta herramienta está integrada en el ecosistema de Google.
- Actualización de datos almacenados a través de archivos .csv o .xlsx.

4.2 Procesamiento de datos

- Con el fin de asegurar la actualización trimestral de los datos de forma automática, se usará Google API y un cliente de está última que permite su integración con Python.
- Para garantizar la calidad de los datos, y la imputación de registros nulos, se usará Python. Principalmente, se utilizarán sus librerías: Pandas,

Numpy, y Pyjanitor.

- Para la construcción de los modelos estadísticos y de Machine Learning, se utilizará Python, y se usarán principalmente sus librerías: Pandas, Numpy, SciPy, Scikit-learn, TensorFlow, Matplotlib, y Seaborn.
- Posterior a la estandarización y preparación de las bases de datos, se usará Oracle SQL developer como herramienta de almacenamiento y manipulación de los mismos.

4.3 Difusión de resultados

- los resultados entregados por los modelos predictivos serán visualizados a través de un dashboard de Power Bi. Esto se hará a través de su función "Scripts de Python".
- Los indicadores descriptivos iniciales también se visualizarán en Power Bi utilizando el conector nativo de bases de datos de Oracle.

5 Integración de datos

Para lograr una integración exitosa de los datos, se ha diseñado un flujo de trabajo que pretende aprovechar algunas herramientas proporcionadas por el entorno de Google y combinarlas con Oracle SQL developer, el cual es un procesador de bases de datos relacionales que permitirá almacenar, gestionar y tratar todos los registros de forma eficiente y segura.

5.1 Extracción de datos en GCP

El acceso inicial a los datos se hará mediante la descarga de un archivo plano (.csv) con las variables necesarias para el análisis. Esta información se encuentra almacenada en la bodega de datos de Positiva Compañía de Seguros, y se descargará a través de GCP (Google Cloud Platform).

Usando consultas SQL a través de BigQuery, se construirá el archivo plano necesario para alimentar las herramientas de análisis posterior. Este archivo será migrado a Google Cloud Storage, a fin de asegurar su disponibilidad y la replica de su información en cualquier fase del flujo de trabajo. Todo esto se realiza con base en un script de Python (Anexos. Figure 2).

Por el tipo de datos que se usarán (estructurados) y el bajo volumen de estos, el flujo de trabajo se ha construido basado en herramientas muy sencillas y accesibles, pero que aseguran la integridad y la confidencialidad de todos los datos en cualquier punto del proceso.

5.2 Carga de datos a Google Colab

Posterior a la creación del archivo plano con los datos, estos se deben integrar con Google Colab directamente. Desde Google Cloud se usa un script de Python

(Anexos. Figure 3) para almacenar de forma automática el archivo en Google Drive, y desde allí se podrá integrar directamente con Google Colab desde cualquier terminal que se vaya a usar.

5.3 Acceso y limpieza de datos en Google Colab

La limpieza de la base de datos se hace con el fin de proveer un procesamiento previo a los datos antes de ser usados o almacenados. Esta limpieza garantiza la consistencia, precisión, confiabilidad, y eficiencia de los registros. Además, permitirá eliminar los datos confidenciales que no se usarán en el estudio, pero cuyo tratamiento está regido por la ley.

Al finalizar este proceso y con la base ya tratada, se generarán dos copias identicas de la misma: una será usada para entrenar y probar los modelos predictivos en Google Colab y la otra será migrada a Oracle SQL Developer, para constuir los indicadores iniciales y estructurar la información actual obtenida (Anexos. Figure 4).

5.4 Carga de datos limpios a Oracle SQL Developer

En este proyecto, se usará Oracle SQL Developer como motor para crear una base de datos estructurada que permita almacenar todos los registros generados en las consultas anteriores. Mediante el tratamiento de esta base de datos, se van a generar una serie de indicadores iniciales de las pólizas de ARL en tiempo real. Posteriormente, y a través de su integración con PowerBi, se podrán visualizar en un tablero de control interactivo.

La carga de los datos en el Developer se hará usando un script de Python (Anexos. Figure 5), y más adelante se automatizará su actualización trimestral.

6 Automatización de datos

La iteración necesaria para construir los modelos predictivos y de Machine Learning se hará mediante la alimentación de la base de datos con nuevos registros obtenidos cada trimestre. Inicialmente se tomarán 4 trimestres del 2023 y el primer trimestre del 2024.

6.1 Automatización del proceso de exportación para cada trimestre

Con el propósito de asegurar la automatización del proceso de consulta, la exportación de datos cada trimestre, y la posterior alimentación de la base de datos, se construye un job usando Google Cloud Scheduler. Esto permite que, a través de una cloud function, se exporten los datos de BigQuery a Cloud Storage en el momento deseado.

Para esto, se debe crear una cloud function. Esta debe contener el código para exportar los datos obtenidos de la consulta de BigQuery a Cloud Storage

(Anexos. Figure. 6), y después, a través de un trigger HTTP (Anexos. Figure. 7), se configurará la activación de la función mediante una solicitud HTTP.

Por último, se configura el job en Cloud Scheduler para llamar la cloud function periódicamente (Anexos. Figure. 8).

6.2 Automatización de la carga a Oracle

Para asegurar que los nuevos registros se carguen en Oracle de manera automática, se debe incluir en la cloud function la función "upload_to_oracle_and_save_copy()". Esta se ejecutará después de que se hayan limpíado los datos en Google Cloud y se hayan guardado en Google Drive.

7 Inteligencia artificial

7.1 Modelos predictivos

El componente de inteligencia artificial del proyecto se integra con los modelos de Machine learning y los métodos estadísticos que se utilizarán para predecir cuáles siniestros pueden constituir fraude según los datos entregados para cada variable utilizada en el estudio.

Para esto, se entrenarán los módelos con los datos recolectados en 5 periodos diferentes (iteraciones), tomando para cada uno muestras para train y para test. Estos modelos serán programados en Python a través de Google Colab, y el cargue de las bases de datos con las que se probarán se hará posterior a la limpieza de los datos que se van a usar.

7.2 Diseño del Dashboard

Los resultados obtenidos se visualizarán a través de un dashboard de PowerBi, utilizando la conexión nativa que tiene la versión de escritorio con los scripts de Python, y de igual manera, se configurará su actualización automática cada tres meses, usando el editor de consultas.

Para minimizar el riesgo de presentar datos erroneos o desactualizados, se va a configurar una alerta de actualización del dashboard, a fin de recibir una notificación en caso de que se presente alguna falla o algún otro problema en la actualización.

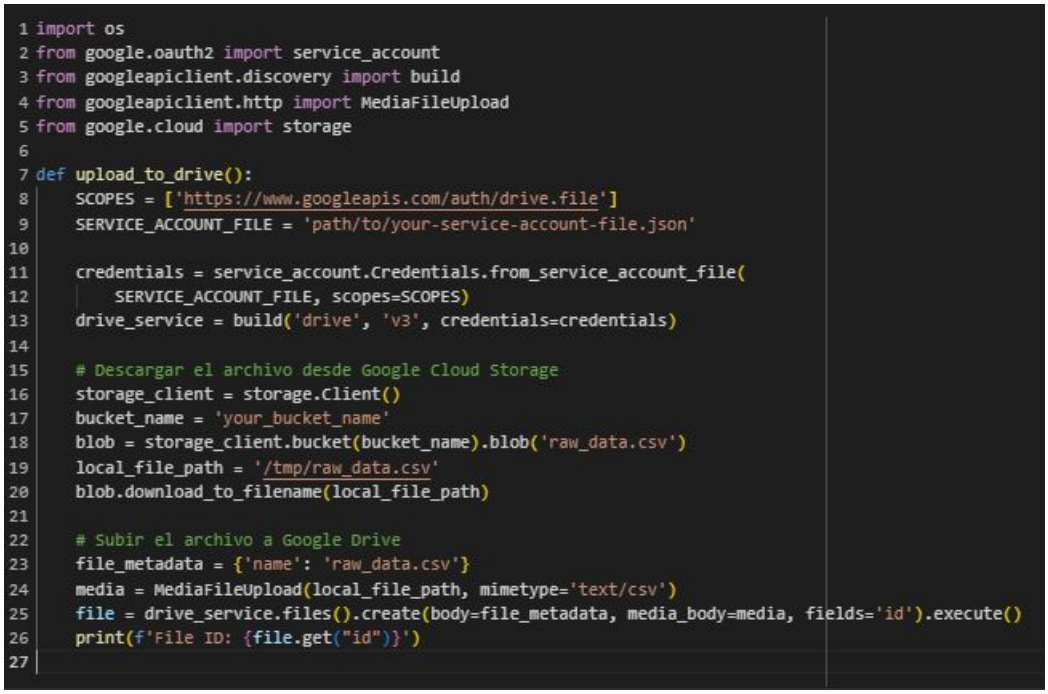
8 Bibliografía

- Revista Fasecolda, 2024. *Mercado mundial de seguros 2021 - 2023.* Extraído el 10 de marzo de 2024 de https://revista.fasecolda.com/index.php/revfasecolda/article/download/830/787/144
- Swiss Re Institute, 2024. *Sigma 3/2023 - World insurance: Stirred, and not shaken.* Extraído el 10 de marzo de 2024 de https://www.swissre.com/institute/research/sigmaresearch/sigma-2023-03.html
- Fasecolda, 2024. *Más colombianos protegidos, la meta de 2024.* Extraído el 10 de marzo de 2024 de https://www.fasecolda.com/cms/wp-content/uploads/2024/02/ComunicadoCifras-2024-F.pdf
- La República, 2024. *Sura, Bolívar y Alfa, aseguradoras líderes en primas de seguros de personas en 2023.* Extraído el 10 de marzo de https://www.larepublica.co/finanzas/surabolivar-y-alfa-las-aseguradoras-lideres-en-seguros-de-personas-el-ano-pasado-3810082
- Portafolio, 2024. *Un colombiano gasta en promedio $970.000 en seguros al año.* Extraído el 10 de marzo de 2024 de https://www.portafolio.co/mis-finanzas/ahorro/balance-del-sector-asegurador-en-2023-59825
- Senado de Colombia, 2012. *Ley estatutaria 1581 de 2012.* Extraido el 01 de abril de 2024 de http://www.secretariasenado.gov.co/senado/basedoc/ley_1581_2012.html
- Gobierno de Colombia, 2013. *Decreto 1377 de 2013.* Extraido el 01 de abril de 2024 de https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=53646.

9 Anexos

```python
from google.cloud import bigquery
from google.cloud import storage

def export_data_to_gcs():
    bigquery_client = bigquery.Client()

    # Configuración de la consulta en BigQuery

    query = """
    SELECT *
    FROM `your_project.your_dataset.your_table`
    """
    query_job = bigquery_client.query(query)

    # Configuración de la exportación del archivo a Cloud Storage
    bucket_name = 'your_bucket_name'
    destination_blob_name = 'raw_data.csv'
    destination_uri = f'gs://{bucket_name}/{destination_blob_name}'

    extract_job = bigquery_client.extract_table(
        query_job.destination,
        destination_uri,
        job_config=bigquery.ExtractJobConfig(destination_format='CSV')
    )
    extract_job.result()
    print(f'Data exported to {destination_uri}')

```

Figure 2: Modelo de script de Python a utilizar para generar las consultas SQL y la extracción de información en archivo plano. (Elaboración propia)



```python
import os
from google.oauth2 import service_account
from googleapiclient.discovery import build
from googleapiclient.http import MediaFileUpload
from google.cloud import storage

def upload_to_drive():
    SCOPES = ['https://www.googleapis.com/auth/drive.file']
    SERVICE_ACCOUNT_FILE = 'path/to/your-service-account-file.json'

    credentials = service_account.Credentials.from_service_account_file(
        SERVICE_ACCOUNT_FILE, scopes=SCOPES)
    drive_service = build('drive', 'v3', credentials=credentials)

    # Descargar el archivo desde Google Cloud Storage
    storage_client = storage.Client()
    bucket_name = 'your_bucket_name'
    blob = storage_client.bucket(bucket_name).blob('raw_data.csv')
    local_file_path = '/tmp/raw_data.csv'
    blob.download_to_filename(local_file_path)

    # Subir el archivo a Google Drive
    file_metadata = {'name': 'raw_data.csv'}
    media = MediaFileUpload(local_file_path, mimetype='text/csv')
    file = drive_service.files().create(body=file_metadata, media_body=media, fields='id').execute()
    print(f'File ID: {file.get("id")}')

```

Figure 3: Modelo de script de Python a utilizar para cargar los archivos a Google Colab desde Google Cloud. (Elaboración propia)

```python
from google.colab import drive
drive.mount('/content/drive')

import pandas as pd

# Leer el archivo .csv desde Google Drive
file_path = '/content/drive/My Drive/raw_data.csv'
df = pd.read_csv(file_path)

# Limpieza de datos
# Ejemplo de procesos de limpieza:
# 1. Eliminar filas duplicadas
df_cleaned = df.drop_duplicates()

# 2. Eliminar filas con valores nulos en columnas críticas
df_cleaned = df_cleaned.dropna(subset=['column1', 'column2'])

# 3. Conversión de tipos de datos
df_cleaned['column3'] = pd.to_datetime(df_cleaned['column3'])

# Guardar los datos limpios de nuevo en Google Drive
cleaned_file_path = '/content/drive/My Drive/cleaned_data.csv'
df_cleaned.to_csv(cleaned_file_path, index=False)
print("Data cleaned and saved to Google Drive")
```

Figure 4: Modelo de script de Python a utilizar para hacer una limpieza y un procesamiento previo a los datos. (Elaboración propia)

```python
import cx_Oracle
import pandas as pd

def upload_to_oracle_and_save_copy():
    dsn_tns = cx_Oracle.makedsn('host', 'port', service_name='service_name')
    connection = cx_Oracle.connect(user='username', password='password', dsn=dsn_tns)
    cursor = connection.cursor()

    # Leer el archivo CSV limpio desde Google Drive
    cleaned_file_path = '/content/drive/My Drive/cleaned_data.csv'
    df_cleaned = pd.read_csv(cleaned_file_path)

    # Inserta los datos limpios en la base de datos Oracle
    for index, row in df_cleaned.iterrows():
        cursor.execute("""
        INSERT INTO your_table (column1, column2, ...)
        VALUES (:1, :2, ...)
        """, (row['column1'], row['column2'], ...))

    connection.commit()
    cursor.close()
    connection.close()

    # Guardar una copia de los datos limpios en Google Drive para análisis estadístico
    analysis_copy_path = '/content/drive/My Drive/analysis_copy_cleaned_data.csv'
    df_cleaned.to_csv(analysis_copy_path, index=False)
    print("Data uploaded to Oracle and copy saved to Google Drive")


```

Figure 5: Modelo de script de Python a utilizar para cargar los datos procesados desde Google Colab a Oracle SQL Developer. (Elaboración propia)

```python
def export_data_to_gcs(event, context):
    from google.cloud import bigquery
    from google.cloud import storage

    bigquery_client = bigquery.Client()

    # Configuración de la consulta de BigQuery
    query = """
    SELECT *
    FROM `your_project.your_dataset.your_table`
    """
    query_job = bigquery_client.query(query)
    results = query_job.result()

    # Convertir los resultados a un DataFrame
    df = results.to_dataframe()

    # Guardar los resultados en un archivo CSV local
    local_file_path = '/tmp/raw_data.csv'
    df.to_csv(local_file_path, index=False)

    # Subir el archivo a Google Cloud Storage
    storage_client = storage.Client()
    bucket_name = 'your_bucket_name'
    destination_blob_name = 'raw_data.csv'

    bucket = storage_client.bucket(bucket_name)
    blob = bucket.blob(destination_blob_name)
    blob.upload_from_filename(local_file_path)

    print(f'Data exported to gs://{bucket_name}/{destination_blob_name}')

```

Figure 6: Modelo de script de Python a utilizar para construir la cloud function de actualización. (Elaboración propia)

```
gcloud functions deploy [data_exporter] \
  --runtime python39 \
  --trigger-http \
  --region=us-central1

```

Figure 7: Modelo de script de Python a utilizar para crear el trigger HTTP de la función. (Elaboración propia)

```
gcloud scheduler jobs create http actual_data \
  --schedule="0 0 1 */3 *" \
  --uri=https://us-central1-[YOUR_PROJECT_ID].cloudfunctions.net/[FUNCTION_NAME] \
  --http-method=POST \
  --message-body="" \
  --time-zone=America/Bogota
```

Figure 8: Modelo de script de Python a utilizar para crear el job en Cloud Scheduler. (Elaboración propia)