Desarrollo de un sistema de información para la recolección y el análisis de datos en una empresa exportadora de café

Fernando Alfonso Orozco Fragozo

Jose Hernando Acuña Olivella

Asesor

Andrés Felipe Hernández Giraldo

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Especialización en Ciencia de Datos y Analítica

2025

Dedicatoria

A Dios, por darme la fuerza, la salud y la sabiduría necesarias para culminar este ciclo académico.

A mi familia, por su amor incondicional, su paciencia y su apoyo constante en cada etapa de mi vida. A mis padres, por enseñarme que la perseverancia y el esfuerzo son el verdadero camino hacia las metas.

A mis amigos y compañeros, por compartir conmigo momentos de aprendizaje, motivación y compañía en este proceso.

Y a todas aquellas personas que, de alguna forma, contribuyeron con su palabra, su ejemplo o su ayuda para que hoy este sueño sea una realidad.

Resumen

El presente proyecto tiene como objetivo desarrollar un Sistema de Información (SI) para una empresa exportadora de café, con el fin de optimizar la recolección y la gestión de los datos comerciales de la misma. La información base fue proporcionada por la empresa en un archivo de Excel que contiene registros de ventas y despachos —denominados internamente como "plan de embarques"—, los cuales constituyen el conjunto de datos de sus operaciones.

En una primera fase, los datos son depurados y estructurados en un entorno Jupyter Notebook, usando el lenguaje de programación Python, con el propósito de garantizar su consistencia y calidad. A continuación, se aplican algoritmos de Machine Learning (Aprendizaje Automático) con el fin de identificar cuál de ellos ofrece un mejor desempeño en la predicción de tendencias relevantes de variables en el conjunto de datos. Aunque estos modelos no serán integrados al sistema de información, forman parte del análisis exploratorio y predictivo de los datos que nos proporcionó la empresa.

El sistema es desarrollado con el Framework Django, integrando los datos en una Base de Datos Relacional, usando el SGBD MySQL, lo cual permite la centralización de la información y un acceso organizado y eficiente. Además, se incorporan gráficos para la visualización de métricas clave, con el fin de facilitar la comprensión de los datos y con ello facilitar la toma de decisiones.

Para validar el funcionamiento del sistema, en el documento se mostrarán capturas de pantalla que muestran el funcionamiento de la aplicación. Como resultado, se espera entregar una herramienta tecnológica funcional que fortalezca la capacidad de gestión de la empresa, incremente su eficiencia operativa y facilite que la empresa tome sus decisiones basándose en los datos de esta.

Palabras clave: Sistema de Información (SI), Base de Datos Relacional, Django, Machine Learning (Aprendizaje Automático), Framework.

Abstract

The objective of this project is to develop an Information System (IS) for a coffee export company in order to optimize the collection and management of its commercial data. The base information was provided by the company in an Excel file containing sales and shipping records—internally referred to as the "SHIPPING PLAN"—which constitute the dataset of its operations.

In the first phase, the data is cleaned and structured in a Jupyter Notebook environment, using the Python programming language, in order to ensure its consistency and quality. Next, machine learning algorithms are applied to identify which one offers the best performance in predicting relevant trends in variables in the dataset. Although these models will not be integrated into the information system, they form part of the exploratory and predictive analysis of the data provided to us by the company.

The system is developed with the Django framework, integrating the data into a relational database using the MySQL DBMS, which allows for the centralization of information and organized and efficient access. In addition, graphs are incorporated for the visualization of key metrics, in order to facilitate the understanding of the data and thus facilitate decision-making.

To validate the functioning of the system, the document will show screenshots that demonstrate the operation of the application. As a result, we expect to deliver a functional technological tool that strengthens the company's management capacity, increases its operational efficiency, and makes it easier for the company to make decisions based on its data.

Keywords: Information System (IS), Relational Database, Django, Machine Learning, Framework.

Tabla de Contenido

Lista de Tablas

Lista de Figuras

Introducción

En el sector exportador de café, la capacidad para recopilar, organizar y analizar datos se ha convertido en un factor clave para la eficiencia operativa y la toma de decisiones. A pesar de ello, muchas empresas aún gestionan su información mediante archivos dispersos y procesos manuales, lo que limita la trazabilidad, incrementa los errores y reduce la capacidad de respuesta ante las dinámicas del mercado.

Este proyecto plantea el desarrollo de un Sistema de Información orientado a centralizar y estructurar datos operativos a partir de los registros de los datos de una empresa en hojas de cálculo. La metodología incluye la depuración de los datos usando el lenguaje de programación Python en un Jupyter Notebook, su almacenamiento en una base de datos relacional con MySQL y la creación de una aplicación web con Django para gestionar la información de forma segura, organizada y accesible.

Complementariamente, se aplican modelos de aprendizaje automático para identificar patrones y evaluar tendencias en algunas de las variables del conjunto de datos. Aunque estos modelos no se incorporan al sistema final, enriquecen el análisis exploratorio. Finalmente, se integran gráficos que facilitan la visualización de algunos aspectos clave, fortaleciendo la capacidad de tomar decisiones basadas en datos en la empresa.

Planteamiento del Problema

Las empresas exportadoras de café enfrentan desafíos crecientes en un entorno global caracterizado por la volatilidad de los precios internacionales, los desafíos climáticos, la demanda de productos diferenciados y los problemas logísticos. Estos retos se ven agravados cuando las organizaciones no cuentan con un Sistema de Información (SI) que permita centralizar, analizar y visualizar los datos operativos de forma eficiente y oportuna. La ausencia de tales herramientas limita la toma de decisiones basada en datos, reduce la capacidad de adaptación y afecta directamente la competitividad.

Incluso en empresas con trayectoria consolidada y crecimiento sostenido, es común encontrar una gestión de datos fragmentada, apoyada en archivos dispersos y procesos manuales. A pesar de que la digitalización representa una estrategia clave, en el sector cafetero especialmente para pequeñas y medianas empresas exportadoras, existe una escasa aplicación de soluciones informáticas que realicen una gestión de datos optima, dado que la mayoría de las iniciativas tecnológicas en este campo se centran en grandes compañías o en soluciones especializadas, sin abordar el desarrollo de sistemas accesibles, adaptables y basados en tecnologías abiertas.

En este contexto, se identifica la necesidad de crear un Sistema de Información que permita a una empresa exportadora de café almacenar sus datos operativos en una base de datos relacional, utilizando un framework de desarrollo ágil que facilite la construcción eficiente y escalable de la aplicación. Con el objetivo de adoptar un enfoque integral, resulta fundamental llevar a cabo un proceso riguroso de limpieza, transformación y análisis exploratorio de los datos como etapa previa a la aplicación de algoritmos de Machine Learning, con el objetivo de encontrar patrones relevantes y tendencias que puedan ser útiles para la empresa.

¿Cómo desarrollar un Sistema de Información para una empresa exportadora de café, a partir del análisis de sus datos, almacenando los mismos en una base de datos relacional para consultarlos a través de una aplicación web desarrollada con un framework basado en python, incorporando visualizaciones de aspectos clave del conjunto de datos de la empresa?

Justificación

En la actualidad, la competitividad en la industria del café exige que las empresas adopten herramientas tecnológicas avanzadas que optimicen sus operaciones, mejoren su capacidad de análisis y permitan una toma de decisiones rápida y basada en los datos de los que disponen. Por ello las empresas exportadoras de café, enfrentan una creciente necesidad de organizar, analizar y utilizar eficientemente los datos generados en las etapas comerciales y logísticas de su cadena de valor, específicamente en los procesos de ventas y embarque del café. Sin embargo, en muchos casos, los procesos manuales de recopilación de datos y la falta de un sistema centralizado generan demoras, errores y dificultan la obtención de información para la toma de decisiones.

En este contexto, el desarrollo de un sistema de información que permita recolectar datos desde su origen mediante formularios web (como ventas y embarques), y almacenarlos en una base de datos centralizada, en una gran mejora porque evita la duplicidad de la información, facilita el acceso simultáneo por parte de múltiples usuarios y mejora la gestión de respaldos, lo cual es fundamental para garantizar la seguridad de los datos en la empresa. Además, parte de la información almacenada puede consultarse para generar visualizaciones gráficas que faciliten el análisis de datos relevantes.

Este proyecto, además de ser técnicamente viable, permite desarrollar una idea que puede ser replicada por otras empresas exportadoras que tenga tamaños similares, o modelos de negocios parecidos promoviendo el desarrollo tecnológico a través del uso de software libre (Python, MySQL, Django).

Objetivos

Objetivo General

Crear un sistema de información en Django que permita la recolección, consulta y gestión de datos operativos en una empresa exportadora de café, a partir de un conjunto de datos previamente limpiados y estructurados en una base de datos mediante técnicas de análisis y procesamiento en Python.

Objetivos Específicos

Realizar un proceso de depuración y transformación de los datos históricos proporcionados por la empresa.

Aplicar algoritmos de aprendizaje supervisado en la predicción de variables operativas clave.

Implementar visualizaciones de datos relevantes que faciliten el monitoreo de indicadores operativos.

Marco Conceptual y Teórico

Para el desarrollo del proyecto nos sustentamos en el siguiente marco conceptual y teórico.

Sistemas de Información y su Rol en la Empresa

Un Sistema de Información (SI) es un conjunto de elementos interrelacionados que recolectan, procesan, almacenan y distribuyen datos con el fin de apoyar la toma de decisiones, la coordinación y el control dentro de una organización. Estos sistemas integran personas, hardware, software, datos y procesos, y son esenciales para el funcionamiento eficaz de las empresas modernas.

En el contexto de una empresa exportadora de café, un SI permite centralizar información de aspectos clave como la logística, las ventas y las certificaciones del café. Esto facilita la visualización y la captura de los datos. Además, un sistema bien diseñado reduce errores, mejora la eficiencia operativa y aumenta la capacidad de respuesta ante los cambios o eventos inesperados.

Gestión y Preparación de los Datos

La gestión de datos comprende todas las acciones necesarias para recolectar, organizar, transformar y mantener los datos en condiciones óptimas para su análisis. En el ciclo de vida del dato, las etapas iniciales son fundamentales para garantizar resultados confiables.

En este proyecto, los datos fueron tratados inicialmente en un entorno Jupyter Notebook utilizando Python, donde se identificaron y gestionaron valores nulos y campos que no eran necesarias para el análisis o contenían información de carácter reservado para la empresa. Asimismo, se codificaron variables categóricas y se generaron nuevas variables a partir de transformaciones sobre campos temporales. Estas acciones permitieron estructurar los datos en

un formato compatible para realizar el análisis exploratorio de datos y la aplicación de algoritmos de machine learning.

Bases de Datos Relacionales

Una base de datos relacional es un sistema estructurado que organiza los datos en tablas con filas y columnas, facilitando su acceso, manipulación y consulta mediante el lenguaje SQL. MySQL es uno de los sistemas de gestión de bases de datos más utilizados por su estabilidad, eficiencia y compatibilidad con aplicaciones web (Coronel & Morris, 2018).

En el proyecto, se creó una base de datos relacional para almacenar la información procesada desde los *dataframes*, facilitando su integración posterior en el sistema de información. Esta base de datos permite consultas rápidas, centraliza la información y evita la duplicidad y fragmentación de esta, proporcionando así un soporte robusto para las funcionalidades del sistema.

Ciencia de Datos y Aprendizaje Automático

La ciencia de datos combina estadística, y la informática para extraer valor de grandes volúmenes de información (Provost & Fawcett, 2013). Dentro de este campo, el Aprendizaje Automático (*Machine Learning*) se refiere al uso de algoritmos que pueden aprender de los datos y hacer predicciones o clasificaciones automáticas (Geron, 2019).

En el presente trabajo se aplicaron algoritmos supervisados como regresión lineal y árboles de decisión. La regresión permitió estimar el volumen exportado de café en kilogramos a partir de variables logísticas y comerciales. Las métricas utilizadas incluyeron el error cuadrático medio (MSE), el coeficiente de determinación (R^2), el MAE, la matriz de confusión y el reporte de clasificación, lo que permitió evaluar objetivamente el rendimiento de los modelos.

Visualización de Datos

La visualización de datos es una herramienta fundamental para comprender patrones, detectar anomalías y comunicar hallazgos de forma efectiva (Knaflic, 2015). Los gráficos permiten concentrar información clave de una manera muy sencilla, lo que facilita el manejo de información y la toma de decisiones (Few, 2006).

En este proyecto, se utilizaron librerías como Seaborn y Matplotlib para construir gráficos de exploración durante el análisis inicial. Además, se planifica la integración de gráficos relevantes dentro del sistema web para presentar indicadores clave de manera clara.

Desarrollo Web con Django

Django es un Framework web de alto nivel basado en Python, que facilita el desarrollo rápido de aplicaciones seguras y escalables. Su arquitectura modelo-vista-controlador (MVC) permite una clara separación entre la lógica de negocio, la presentación y el acceso a datos (Holovaty & Kaplan-Moss, 2009).

En el marco del proyecto, Django se utilizará para construir el sistema de información que permitirá ingresar, consultar y visualizar la información contenida en la base de datos relacional. El sistema se ejecutará inicialmente en entorno local (localhost), y se mostrará su funcionamiento a través de la documentación de capturas del mismo en este trabajo. Esta implementación contribuirá a modernizar la gestión operativa de la empresa, facilitando el acceso a información crítica para la toma de decisiones.

Metodología

La presente investigación se enmarca dentro del enfoque cuantitativo, ya que se fundamenta en la recolección, transformación y análisis de datos estructurados provenientes de registros históricos operativos de una empresa exportadora de café. Se trata de una investigación aplicada, dado que busca resolver un problema real mediante la implementación de un sistema de información que integre técnicas de procesamiento de datos, visualización y modelos predictivos.

Fases del Proyecto

La metodología que se siguió para el desarrollo del proyecto se estructuró en tres etapas principales:

1. Recolección, Limpieza y Carga de los Datos: Se obtuvieron datos operativos correspondientes a ventas y despachos del periodo 2022–2023, suministrados por una empresa exportadora de café mediante un archivo de Excel. Esta información fue cargada en un Jupyter Notebook utilizando la librería *pandas*, donde se aplicaron técnicas de limpieza y análisis exploratorio. Posteriormente, los datos fueron cargados en una base de datos relacional previamente estructurada mediante un script SQL, utilizando los DataFrames como fuente para poblar las tablas definidas.

2. Análisis Predictivo con Aprendizaje Supervisado: A partir de los datos contenidos en la hoja ‘Ventas’, se aplicaron tres algoritmos de aprendizaje supervisado con el fin de predecir variables clave del negocio. La regresión lineal se utilizó para estimar el volumen exportado en kilogramos (KG ESTÁNDAR), obteniendo un R^2 de 0.99, lo que evidencia un ajuste casi perfecto y una alta capacidad predictiva. Para la clasificación de embarques como de alto volumen (True/False), definidos como aquellos que superan los 10.000 kg, se implementaron un

Árbol de Decisión y un modelo k-Nearest Neighbors (k = 3), ambos con una precisión del 100 %. No obstante, el Árbol de Decisión se destaca por su robustez y facilidad de interpretación, mientras que el modelo k-NN, aunque igual de preciso, es más sensible a variaciones en los datos. En conjunto, los resultados muestran que la regresión lineal es ideal para predicciones numéricas, y el Árbol de Decisión para tareas de clasificación confiables y explicables.

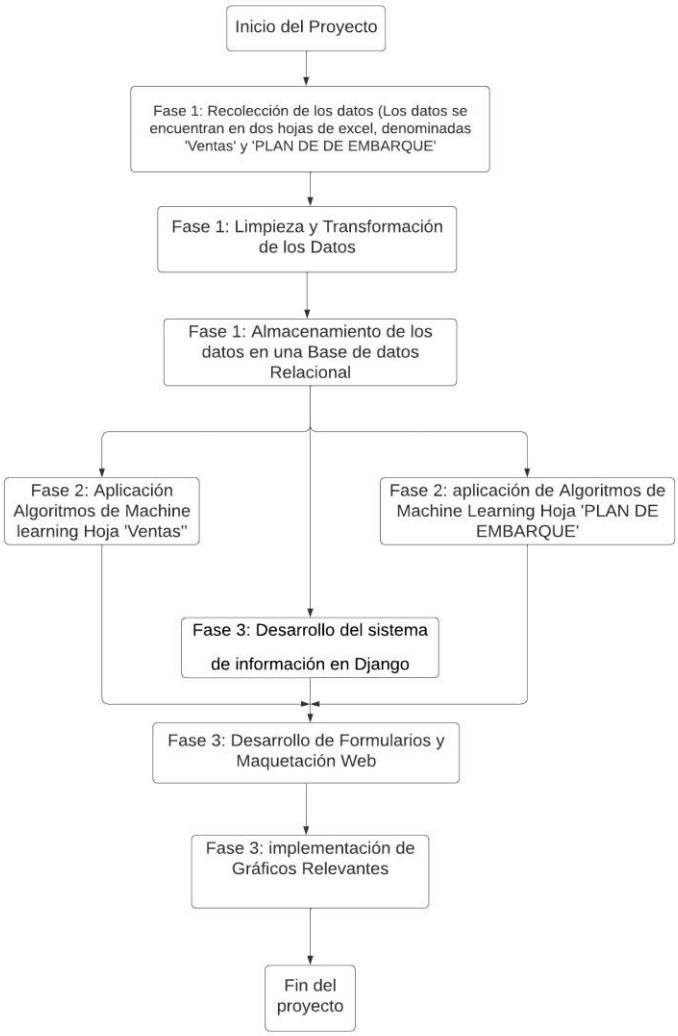
En el caso de los datos registrados en la hoja 'PLAN DE EMBARQUE', se aplicaron tres modelos de aprendizaje supervisado con el objetivo de predecir el número de sacos exportados. La regresión lineal presentó el mejor desempeño general, con un error absoluto medio (MAE) de 30.72 y un coeficiente de determinación (R^2) de 0.4985, lo que indica un ajuste razonable y una adecuada capacidad para modelar relaciones lineales. El Árbol de Decisión obtuvo resultados ligeramente inferiores (MAE de 32.11 y R^2 de 0.4042), aunque con la ventaja de capturar posibles relaciones no lineales entre las variables. Por su parte, el modelo k-Nearest Neighbors (k = 3) fue el que mostró el menor rendimiento (R^2 de 0.2650), evidenciando una menor sensibilidad a valores extremos y predicciones más planas. En conjunto, se concluye que la regresión lineal es la alternativa más sólida para estimaciones numéricas en este conjunto de datos.

3. Desarrollo del Sistema de Información: Utilizando el Framework *Django*, se desarrolló una aplicación web conectada a la base de datos, con funcionalidades que permiten el registro, consulta y visualización de los datos operativos. El sistema incluye formularios web y gráficos de aspectos relevantes de las operaciones de la empresa, generados usando Django con librerías de Python, lo que proporciona una solución centralizada y dinámica para apoyar la gestión y el análisis de información en la empresa.

Para ilustrar esta metodología en la Figura 1, realizamos un diagrama fases del proyecto en forma de diagrama de flujo.

Figura 1

Diagrama con las Fases del Proyecto en Forma de Diagrama de Flujo



Depuración y Transformación de los Datos

El conjunto de datos que vamos a analizar corresponde a una empresa exportadora de café que almacena la información de sus operaciones en un archivo de Excel, en el proceso de recolección de los datos la empresa nos entregó ese archivo el cual tenía por nombre ‘Plan de Embarque 2022-2023.xlsx’ , el mismo tiene dos hojas, la primera se llama 'Ventas' y la segunda se llama 'Plan De Embarque', ambas contienen el registro de la información comercial de la empresa, en la hoja 'Ventas' se encuentra la información macro de los contratos generales de las ventas de café, y en la hoja 'Plan De Embarque' se encuentra la información de la ejecución de los contratos, para abordar el proceso de depuración y transformación de los datos usaremos un jupyter notebook, que nos permitirá realizar las operaciones sobre el conjunto de datos usando el lenguaje de programación Python, como analizaremos dos conjuntos de datos empezaremos primero por la hoja denominada ‘Ventas’ y posteriormente realizaremos un proceso similar sobre la hoja denominada ‘Plan De Embarque’.

Hoja ‘Ventas’

A continuación, realizaremos el proceso de depuración y transformación del conjunto de datos contenidos, en la hoja denominada ‘Ventas’ del archivo proporcionado por la empresa llamado ‘Plan de Embarque 2022-2023.xlsx’.

Carga y Comprensión de los Datos

Para realizar el proceso de carga de los datos y el procesamiento posterior que vamos a realizar sobre el conjunto de datos nos valemos de las librerías mostradas en la Figura 2.

Figura 2

Librerías de Python Usadas en el Procesamiento de los Datos

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Una vez que hemos importado las librerías necesarias procedemos a cargar el conjunto de datos de datos tal como se muestra en la Figura 3, para facilitar el análisis hemos cambiado el nombre original del conjunto de datos por el nombre ‘InformacionEmpresa.xlsx’.

Figura 3

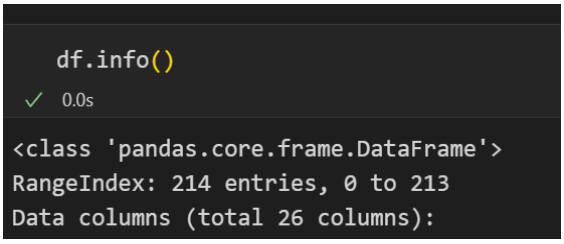
Carga del Conjunto de Datos

```
df=pd.read_excel('informacionEmpresa.xlsx',sheet_name='Ventas',header=0)
✓ 0.2s
```

Ahora procedemos a usar el método info() de la biblioteca pandas para obtener un resumen de la información del dataframe tal como se muestra en la Figura 4.

Figura 4

Resumen de la Información del Dataframe



Vemos que el mismo tiene 214 registros y 26 columnas, la siguiente parte del análisis será realizar un proceso de depuración de las columnas irrelevantes o vacías.

Depuración de Datos

En la Tabla 1, mostramos las columnas que serán eliminadas del conjunto de datos, entre ellas tenemos la columna 'Estado de contrato', la cual tiene casi todos sus valores nulos salvo uno, procederemos a eliminar esa columna y a eliminar aquellas columnas que no aportan mucho valor al análisis o que tienen que ver con información reservada de la empresa, como son 'Referencia interna', y 'Contrato de venta'.

Tabla 1

Columnas a Eliminar Hoja 'Ventas'.

Nombre de la columna	Non-Null Count	Dtype
Estado de contrato	1 non-null	object
Referencia interna	166 non-null	object
Contrato de venta	166 non-null	object

Nota. Esta tabla muestra los nombres de las columnas que serán eliminadas en la hoja denominada 'Ventas'

El proceso de eliminación de las columnas mencionadas en el párrafo anterior se muestra en la Figura 5, donde se hace una lista de las columnas y luego se aplica el método drop(), de la biblioteca pandas.

Figura 5

Eliminación de las Columnas Innecesarias Para el Análisis

```
# Eliminar columnas específicas del DataFrame
columnas_a_eliminar = ["ESTADO DE CONTRATO", "REFERENCIA INTERNA", "CONTRATO DE VENTA"]
df= df.drop(columns=columnas_a_eliminar)
✓ 0.0s
```

A continuación, procedemos a eliminar los espacios en blanco de los nombres de las columnas tal como se muestra en la Figura 6, para mejorar la legibilidad y la eficiencia del código.

Figura 6

Eliminación de los Espacios en Blanco de los Nombres de las Columnas

```
df.columns=df.columns.str.strip()
✓ 0.0s
```

No se eliminaron más columnas porque, aunque algunas tienen pocos datos, los datos que tienen pueden ser relevantes.

Transformación de los Datos

Con el propósito de analizar las tendencias estacionales a lo largo de los meses, extraemos de la columna denominada 'MES DE EMBARQUE', el mes y creamos una nueva columna, llamada 'MES', tal como se muestra en la Figura 7, con el propósito de observar

posibles tendencias mensuales, dado de que la información solo abarca el año 2022 y el año 2023, no vemos necesario crear una nueva columna para el año.

Figura 7

Creación de una Columna Para Observar Tendencias Estacionales

```
df['MES EMBARQUE'] = pd.to_datetime(df['MES DE EMBARQUE'], errors='coerce')
df['MES'] = df['MES EMBARQUE'].dt.month
```
✓ 0.0s

Dado que pretendemos obtener informaciones estadísticas, es importante asegurarnos que las columnas numéricas estén en formato numérico, para ello como se muestra en la Figura 8, usamos el meto to_numeric() de la librería pandas.

Figura 8

Conversión de las Columnas en Numéricas

```
# Asegurar que las columnas numéricas estén bien tipificadas
df["SACOS"] = pd.to_numeric(df["SACOS"], errors="coerce")
df["KG ESTÁNDAR"] = pd.to_numeric(df["KG ESTÁNDAR"], errors="coerce")
df["UNIDAD DE EMPAQUE (Kg)"] = pd.to_numeric(df["UNIDAD DE EMPAQUE (Kg)"], errors="coerce")
```
✓ 0.0s

Análisis de los Datos

Una vez hechos los procesos de limpieza y transformación procedemos a analizar el conjunto de datos, y extraer conclusiones relevantes, para ello lo primero que hacemos es obtener el resumen estadístico de las variables numéricas, tal como se muestra en la Figura 9.

Figura 9

Código Para Obtener Resumen Estadístico de las Variables Numéricas

```
df[["SACOS", "KG ESTÁNDAR", "UNIDAD DE EMPAQUE (Kg)"]].describe().T
```

A continuación, en la Tabla 2 procedemos a mostrar los resultados obtenidos del resumen estadístico de las variables numéricas.

Tabla 2

Resumen Estadístico

	Count	Mean	Std	Min	25%	50%	75%	Max
Sacos	166.0	271.49	138.25	9.0	275.0	275.0	285.0	855.0
Kg Estándar	214.0	14635.09	10142.81	0.0	1137.5	19250.0	19950.0	59850.0
Unidad de Empaque (kg)	166.0	112.53	200.56	20.0	70.0	70.0	70.0	1000.0

Nota. Resumen estadístico de las variables numéricas de la hoja ‘Ventas’

Del resumen estadístico, vemos que los datos que se encuentran en la variable 'SACOS', y que se corresponde al tamaño de las ventas de café en sacos, el valor está muy concentrado en 275, pero hay valores atípicos y esto se explica porque la empresa además de vender a grandes y medianos compradores también tiene como clientes a pequeñas cafeterías por ello el tamaño mínimo de envió que se encontró fue de 9, pero hay envíos de 20, de 13 y de 10 sacos, en cuanto a los datos en la variable 'Unidad De Empaque (Kg)', vemos que hay uniformidad en torno a los 70kg, sin embargo en el resumen estadístico, el valor más pequeño es 20kg que tiene que ver con la explicación anterior de las ventas a pequeñas cafeterías y por el extremo superior el valor máximo es de 1000 eso tiene que ver con clientes que realizan pedidos embazados en bolsas tipo

big bags, que es una bolsa especial donde se empaca el café al vacío, y por ello puede conservarlo mejor por más tiempo.

Uno de los hallazgos relevantes del conjunto de datos es conocer que países son los mayores compradores de café de la empresa, para ello ejecutamos el código que se ve en la Figura 10.

Figura 10

Código Para Obtener Mayores Compradores de café

```
# Agrupar por país y comprador, sumar sacos
agrupado = (
    df.groupby(["PAIS"])["SACOS"]
    .sum()
    .sort_values(ascending=False)
    .reset_index()
)

# Mostrar los 10 mayores (puedes ajustar el número)
print(agrupado.head(10))
```

En la Tabla 3 vemos que países son los mayores compradores de café de la empresa.

Tabla 3

Mayores Compradores de Café

Pais	Sacos
United States	13009.0
Colombia	9797.0
Belgium	8640.0

Pais	Sacos
France	3985.0
United Kingdom	1975.0
Panama	1595.0
Alemania	1375.0
Canada	1375.0
Costa Rica	855.0
Italy	855.0

Nota. Mayores compradores de café organizados por país, según la información de la hoja 'Ventas'

Vemos que entre los países que más compran café están Estados Unidos, Bélgica, Francia, Reino Unido y también se encuentra Colombia, porque la empresa les vende a otras empresas que se encuentran en Colombia.

Ahora realizaremos una agrupación en el conjunto de datos, para conocer cuáles son las empresas en los países compradores a las que se les vende el café, el código para realizar dicha agrupación se muestra en la Figura 11.

Figura 11

Código Para Agrupar los Países y las Empresas Compradoras

```
# Agrupar por país y comprador, sumar sacos
agrupado = (
    df.groupby(["PAIS", "IMPORTADOR / EXPORTADOR"])["SACOS"]
    .sum()
    .sort_values(ascending=False)
    .reset_index()
)

# Mostrar los 10 mayores (puedes ajustar el número)
print(agrupado.head(10))
```

En la Tabla 4 se muestra el resultado de la ejecución del código de la Figura 12.

Tabla 4

Empresas Compradores de Café en los Países Destino

Pais	Importador / Exportador	Sacos
Colombia	Expocafe	7800.0
United States	Sucafina NA Inc	7700.0
Belgium	Efico Nv	4660.0
Belgium	Briz Coffees NV	3980.0
France	Meo-Fichaux	3435.0
United States	Armenia Coffee Corporation	2475.0
United States	Volcafe USA	2200.0
United Kingdom	D.R Wakefield & Co. Ltda	1395.0
Alemania	Hamburg Coffee Company Hacofco	1375.0

Pais	Importador / Exportador	Sacos
Canada	Ken Gabbay Coffee Ltd.	1375.0

Nota. Empresas compradoras de café en los países destino según la información de la hoja 'Ventas'

Vemos que varias empresas que compran el café para el extranjero son empresas que tienen sede en Colombia y en el exterior de ahí se deriva que el nombre de varias de ellas sea en español.

Otro hallazgo importante es conocer con que tipo de certificaciones se vende el café a los países compradores, para ello vamos a ejecutar el código que se muestra en la Figura 12.

Figura 12

Código Para Obtener las Certificaciones con las que se Comercializa el Café

```
# Asegurar que 'SACOS' sea numérico
df["SACOS"] = pd.to_numeric(df["SACOS"], errors="coerce")

# Agrupar por país y tipo de certificación ('CERT'), sumando los sacos
agrupado_pais_cert = (
    df.groupby(["PAIS", "CERT"])["SACOS"]
    .sum()
    .sort_values(ascending=False)
    .reset_index()
)

# Mostrar los 10 primeros resultados
agrupado_pais_cert.head(10)
```

El resultado de la ejecución del código mostrado en la Figura 12 se muestra en la Tabla 5 Vemos que la mayoría del café comercializado se vende sin certificación.

Tabla 5

Certificaciones del Café Comercializado

Pais	CERT	Sacos
Belgium	Sin certificación	6545.0
Colombia	Sin certificación	4983.0
United States	FTO	4950.0
United States	ORGANICO	3505.0
Colombia	FTO	2570.0
France	Sin certificación	2280.0
United Kingdom	RFA	1395.0
United States	RFA	1375.0
United States	Orgánico - RFA	1240.0
Colombia	Orgánico	1144.0

Otro hallazgo relevante desde el punto de vista logístico es desde que puertos, salen las ventas hacia los respectivos países, el código para obtener dichos puertos se muestra en la Figura 13.

Figura 13

Código Para Agrupar Los Puertos Desde Donde Sale El Café Vendido Al Exterior

```
# Convertir 'SACOS' a numérico
df["SACOS"] = pd.to_numeric(df["SACOS"], errors="coerce")

# Agrupar por país y por INCONTERMS, sumando los sacos
agrupado_pais_inconterms = (
    df.groupby(["PAIS", "INCOTERMS"])["SACOS"]
    .sum()
    .sort_values(ascending=False)
    .reset_index()
)

# Mostrar los 10 primeros resultados
agrupado_pais_inconterms.head(10)
```

El resultado de la ejecución mostrado en la Figura 13 se muestra en la Tabla 6.

Tabla 6

Puertos Desde los que Sale la Carga a los Países Destino

País	INCOTERMS	Sacos
United States	FOB-NBA	8264.0
Belgium	FOB-CTGENA	5705.0
Colombia	EXW	4490.0
United States	FOB-BUN	3025.0
Belgium	FOB-NBA	2935.0
France	FOB-CTGENA	1995.0
United Kingdom	FOB-NBA	1955.0
France	FOB-NBA	1415.0

País	INCOTERMS	Sacos
Canada	FOB-NBA	1375.0
Alemania	FOB-NBA	1375.0

Nota. Puertos desde los que sale la carga a los países de destino.

Vemos que la mayoría de los envíos hacia los Estados Unidos salen del puerto de Barranquilla, seguido del puerto de Buenaventura, mientras que los envíos a Europa salen de los puertos de Cartagena y Barranquilla.

Otro hallazgo significativo es el tipo de empaque en el que se envía el café vendido a los compradores, a continuación en la Figura 14 mostramos el código para realizar dicho hallazgo.

Figura 14

Código Para Encontrar el Tipo de Empaque Según País de Destino

```
# Convertir 'SACOS' a numérico
df["SACOS"] = pd.to_numeric(df["SACOS"], errors="coerce")

# Agrupar por país y tipo de empaque, sumando los sacos
agrupado_pais_empaque = (
    df.groupby(["PAIS", "TIPO DE EMPAQUE"])["SACOS"]
    .sum()
    .sort_values(ascending=False)
    .reset_index()
)

# Mostrar los 10 primeros resultados
agrupado_pais_empaque.head(10)
```

En la Tabla 7 mostramos el resultado de la ejecución del código de la Figura 14.

Tabla 7

Tipo de Empaque Según el País de Destino

País	Tipo de Empaque	Sacos
United States	Fique	12720.0
Colombia	Fique	9512.0
Belgium	Fique	7100.0
France	Fique	3985.0
United Kingdom	Fique	1935.0
Belgium	Granel	1500.0
Alemania	Fique	1375.0
Canada	Fique	1375.0
Panama	F. Arte Especial	1035.0
Costa Rica	F. Arte Especial	855.0

Vemos que el tipo de empaque que predomina en la mayoría de los envíos es el Fique, que suele ser empaque tradicional.

Por último, en la Figura 15, mostramos el código para obtener la tendencia de ventas mensuales de café, este dato es importante porque le puede servir a la empresa para tomar decisiones sobre su inventario.

Figura 15

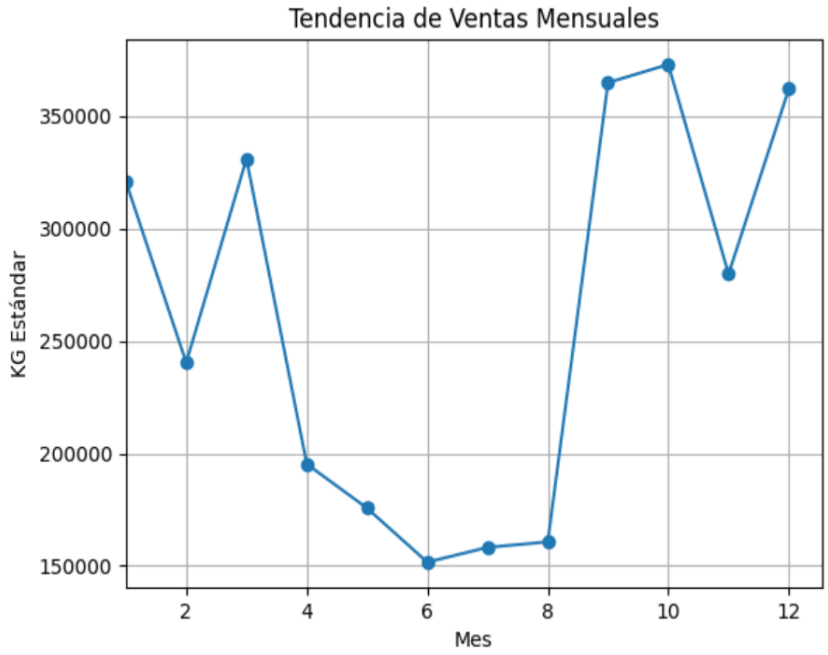
Código Para Obtener Tendencias de Ventas Según el Mes

```
df.groupby('MES')['KG ESTÁNDAR'].sum().plot(kind='line', marker='o')
plt.title('Tendencia de Ventas Mensuales')
plt.xlabel('Mes')
plt.ylabel('KG Estándar')
plt.xlim(left=1)
plt.grid(True)
plt.show()
```

En la Figura 16 se muestra el resultado de la ejecución del código de la Figura 15.

Figura 16

Grafica de Tendencia de Ventas Mensuales



Vemos que el mes de octubre es el mes donde la empresa tiene mas ventas y el mes de junio donde las ventas son más bajas.

Hoja ‘Plan De Embarque’

Luego de haber analizado la información más relevante de la hoja 'Ventas', procedemos a analizar los datos de la hoja 'Plan De Embarque', aunque hay algunos campos que contienen la misma información, en la hoja 'Plan De Embarque' se discriminan las entregas para el cumplimiento de los contratos macro, para desarrollar el análisis vamos a desarrollar un proceso similar al anterior.

Carga y Comprensión de los Datos

Al igual que en el paso anterior lo primero que hacemos es cargar las librerías necesarias en el jupyter notebook, tal como se muestra en la Figura 17.

Figura 17

Librerías De Python Para Procesar Los Datos Hoja 'Plan De Embarque'

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Una vez que hemos importado las librerías necesarias procedemos a cargar el conjunto de datos de datos tal como se muestra en la Figura 18, para facilitar el análisis hemos cambiado el nombre original del conjunto de datos por el nombre ‘InformacionEmpresa.xlsx’.

Figura 18

Carga del Conjunto de Datos Hoja 'Plan de Embarque'

```
df1=pd.read_excel('informacionEmpresa.xlsx',sheet_name='PLAN DE EMBARQUE', header=0)
✓ 0.3s
```

Ahora procedemos a usar el método info() de la biblioteca pandas para obtener un resumen de la información del dataframe tal como se muestra en la Figura 19.

Figura 19

Resumen de la Información del Dataframe

```
df1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134 entries, 0 to 133
Data columns (total 49 columns):
```

Vemos que el mismo tiene 134 registros y 49 columnas, la siguiente parte del análisis será realizar un proceso de depuración de las columnas irrelevantes o vacías.

Depuración de Datos

En la Tabla 8, podemos observar que la columna ‘CONTRATOS PROVEEDOR’, tiene todos sus valores nulos salvo, la columna ‘PROVEEDOR’ tiene solamente un valor no nulo, procederemos a eliminar esa columna y a eliminar aquellas columnas que no aportan mucho valor al análisis o que tienen que ver con información reservada de la empresa, como son 'Our Ref’, y 'Reference'.

Tabla 8

Columnas a Eliminar Hoja 'Plan de Embarque'

Our Ref	134 non-null
Reference	134 non-null
Contratos Proveedor	0 non-null
Proveedor	1 non-null

Nota. Columnas que serán eliminadas de la hoja ‘Plan de Embarque’

El proceso de eliminación de las columnas mencionadas en el párrafo anterior se muestra en la Figura 20, donde se hace una lista de las columnas y luego se aplica el método drop(), de la biblioteca pandas.

Figura 20

Eliminación de las Columnas Innecesarias para el Análisis

```
# Eliminar columnas específicas del DataFrame
columnas_a_eliminar = ['OUR REF','REFERENCE','CONTRATOS PROVEEDOR','PROVEEDOR']
df1= df1.drop(columns=columnas_a_eliminar)
```

A continuación, procedemos a eliminar los espacios en blanco de los nombres de las columnas tal como se muestra en la Figura 21, para mejorar la legibilidad y la eficiencia del código.

Figura 21

Eliminación de los Espacios en Blanco de los Nombres de las Columnas

```
df1.columns=df1.columns.str.strip()
```

No se eliminaron más columnas porque, a pesar de algunas tienen pocos datos, los datos que tienen pueden ser relevantes.

Transformación de los Datos

Con el propósito de analizar las tendencias estacionales a lo largo de los meses, extraemos de la columna denominada 'ETS', el mes y creamos una nueva columna, llamada 'MES', tal como se muestra en la Figura 22, con el propósito de observar posibles tendencias

mensuales, dado de que la información solo abarca el año 2022 y el año 2023, no vemos necesario crear una nueva columna para el año.

Figura 22

Creación de una Columna Para Observar Tendencias Mensuales

```
df1['ETS'] = pd.to_datetime(df1['ETS'], errors='coerce')
df1['MES'] = df1['ETS'].dt.month
```

✓ 0.0s

Análisis de los Datos

Una vez hechos los procesos de limpieza y transformación procedemos a analizar el conjunto de datos, y extraer conclusiones relevantes, para ello lo primero que hacemos es obtener el resumen estadístico de las variables numéricas, para ello usamos el código que se muestra en la Figura 23.

Figura 23

Código Para Obtener El Resumen Estadístico De La Variable Numérica Relevante

```
df1.select_dtypes(include=['number']).describe().T
```

El resultado la ejecución del código de la Figura 23 se muestra en la Tabla 9.

Tabla 9

Resumen Estadístico Variables Numéricas Hoja 'Plan de Embarque'

	Count	Mean	Std	Min	25%	50%	75%	Max
Bags	134.0	255.776.119	99.920.092	9.0	275.0	275.0	285.0	550.0

Nota. Resumen estadístico de las variables numéricas del conjunto de datos de la hoja llamada 'Plan de Embarque'.

Vemos que la mayoría de los despachos de están entre 275 y 285 con valores atípicos el más pequeño de 9 y el más grande de 550, esto se explica en el caso del más pequeño al negocio de la empresa con cafeterías y a los más grandes con pedidos especiales.

Entre los hallazgos relevantes tenemos, cuáles son los principales productores de las marcas que compran las empresas, para ello ejecutamos el código mostrado en la Figura 24.

Figura 24

Código Para Obtener los Productores de las Marcas de Café Vendidas

```
# Agrupar por BUYER, MARKS y PRODUCER y contar los registros
agrupado_buyer_marks_producer = (
    df1.groupby(["BUYER", "MARKS", "PRODUCER"])
    .size()
    .reset_index(name="TOTAL_REGISTROS")
    .sort_values("TOTAL_REGISTROS", ascending=False)
)

# Mostrar los 10 primeros resultados
agrupado_buyer_marks_producer.head(10)
```

El resultado de ejecutar el código mostrado en la Figura 24 se muestra en la Tabla 10.

Tabla 10

Productores de Las Marcas de Café Vendidas

Buyer	Marks	Producer	Total Registros
Sucafina NA Inc	Colombia Excelso FTO	Asoprokia	14
Expocafe	Colombia Excelso sin Certificación	Team Tap	10
Sucafina NA Inc	Colombia Excelso Orgánico	Tayronaca	7
Expocafe	Colombia Excelso FTO	Asoprokia	7
Efico Nv	Colombia excelso sin certifcacion	Team Tap	5
Montana Coffee Importers	Colombia excelso sin certifcacion	Team Tap	4
Armenia Coffee Corporation	Colombia excelso organico	Tayronaca	4
Expocafe	Colombia excelso RFA	Team Tap	4
Volcafe USA	Colombia excelso RFA	Asoagrotol Galilea	4
Armenia Coffee Corporation	Colombia excelso FTO	Asoprokia	3

Nota. Empresas productoras de las marcas de café vendidas por la empresa

Esto nos permite conocer cuantas marcas de café producen las empresas, y su volumen de clientes.

Otro hallazgo significativo, es conocer cuáles son los puertos de partida desde los cuales las empresas realizan sus envíos y cuáles son sus puertos de destino, para lograrlo ejecutamos el código que se muestra en la Figura 25.

Figura 25

Código Para Encontrar los Puertos de Partida y Llegada

```
# Agrupar por BUYER, PORT y DEST y contar los registros
agrupado_buyer_port_dest = (
    df1.groupby(["BUYER", "PORT", "DEST"])
    .size()
    .reset_index(name="TOTAL_REGISTROS")
    .sort_values("TOTAL_REGISTROS", ascending=False)
)

agrupado_buyer_port_dest.head(10)
```

El resultado de ejecutar dicho código se muestra en la Tabla 11.

Tabla 11

Puertos de Partida y Llegada Para los Envíos de las Empresas

Buyer	Port	Dest	Total registros
Sucafina NA Inc	FOB-NBA	New York	13
Armenia Coffee Corporation	FOB-NBA	New York	9
Sucafina NA Inc	FOB-NBA	Toronto	7
Britt Brands Colombia S.A.S	DAP	Quindío	4
GoodSam Foods	FOB-SMTA	New York	4

Buyer	Port	Dest	Total registros
Montana Coffee Importers	FOB-NBA	Oakland	4
Volcafe USA	FOB-BUN	Houston, TX	4
Sucafina NA Inc	FOB-BUN	Los Angeles	4
MCP Alliance Corp	FOB-BUN	Yokohama	3
Expocafe	EXW	Europa	3

Vemos que el principal puerto de salida es Barranquilla y su principal puerto de llegada es New York, algo que no sorprende dado que Estados Unidos es el mayor comprador por país.

Otro hallazgo significativo, cuáles son las principales empresas transportadoras de carga local que llevan, la carga a los puertos, en la Figura 26, mostramos el código y los resultados que nos mostró.

Figura 26

Código Para Encontrar Transportadoras que Llevan la Carga a los Puertos

```
# Agrupar por TRANSPORADORA y PORT y contar los registros
agrupado_transportadora_port = (
    df1.groupby(["TRANSPORADORA", "PORT"])
    .size()
    .reset_index(name="TOTAL_REGISTROS")
    .sort_values("TOTAL_REGISTROS", ascending=False)
)

# Mostrar los primeros resultados
agrupado_transportadora_port.head(10)
```

El resultado de ejecutar dicho código se muestra en la Tabla 12.

Tabla 12

Empresas Transportadoras que Llevan la Carga a los Puertos

Transportadora	PORT	Total Registros
Condetrans	FOB-NBA	56
Condetrans	FOB-BUN	10
Cootranscarga	FOB-SMTA	5
Condetrans	FOB-CTGENA	4
Cootranscarga	FOB-BUN	3
Condetrans	DAP	3
Coldetrans	FOB-CTGENA	2
Partiular	DAP	2
Condetrans	0	2
Barbarita	0	1

Vemos que la principal empresa transportadora es CODETRANS y el puerto al que más lleva carga es el de Barranquilla.

Otro hallazgo significativo son las navieras que llevan la carga desde los puertos nacionales a puertos extranjeros, el código para visualizar dicha información se muestra en la Figura 27.

Figura 27

Código Para Encontrar Principales Navieras Puerto Origen-Puerto Destino

```
# Agrupar por SHIPPING LINE, PORT y DEST y contar los registros
agrupado_shipping_port_dest = (
    df1.groupby(["SHIPPING LINE", "PORT", "DEST"])
    .size()
    .reset_index(name="TOTAL_REGISTROS")
    .sort_values("TOTAL_REGISTROS", ascending=False)
)
agrupado_shipping_port_dest.head(10)
```

El resultado de ejecutar dicho código se muestra en la Tabla 13.

Tabla 13

Principales Líneas Navieras Puerto Origen-Puerto Destino

Shipping Line	PORT	DEST	Total Registros
MSC	FOB-NBA	New York	18
HAPAG	FOB-NBA	Toronto	8
CMA CGM	FOB-NBA	New York	5
CMA CGM	FOB-BUN	Houston TX	4
CMA CGM	FOB-BUN	Los Angeles	4
EVERGREEN	FOB-SMTA	New York	4
MEARK	FOB-NBA	Antwerp	4
CMA CGM	FOB-NBA	Oakland	4
CMA CGM	FOB-NBA	Antwerp	3
CMA CGM	FOB-CTGENA	Antwerp	3

Vemos que la principal naviera internacional es MSC que parte desde Barranquilla y llega al puerto de Nueva York.

Relacionado con los hallazgos anteriores, tenemos cómo se comportan según el mes los despachos de los puertos colombianos a los puertos extranjeros, para conocer esa información ejecutamos el código que se muestra en la Figura 28.

Figura 28

Código Para Conocer los Despachos de los Puertos Según el Mes

```
agrupado_shipping_port_dest = (
    df1.groupby(["MES", "PORT", "DEST"])
    .size()
    .reset_index(name="TOTAL_REGISTROS")
    .sort_values("TOTAL_REGISTROS", ascending=False)
)
agrupado_shipping_port_dest.head(10)
```

El resultado de la ejecución del código anterior se muestra en la Tabla 14.

Tabla 14

Despachos de los Puertos Según el Mes

Mes	PORT	DEST	Total Registros
5.0	FOB-NBA	New York	10
2.0	FOB-NBA	New York	5
5.0	FOB-NBA	Oakland	4
3.0	FOB-NBA	New York	3
5.0	FOB-NBA	Toronto	3
12.0	FOB-BUN	Houston TX	3

Mes	PORT	DEST	Total Registros
9.0	FOB-NBA	Antwerp	3
8.0	FOB-NBA	Toronto	3
8.0	FOB-NBA	Antwerp	2
8.0	FOB-NBA	Antwerp	2

Tenemos también, cuáles son las principales trilladoras, el código para encontrar dicha información se muestra en la Figura 29.

Figura 29

Código Para Encontrar Las Principales Trilladoras

```
df1['MILL'].value_counts()
```

El resultado de la ejecución de dicho código se muestra en la Tabla 15.

Tabla 15

Principales Trilladoras

Trilladora	Total Registros
Real Cafetera	54
Manizales	24
Santa Tirsa	20
Coffenar	16
Barbarita	10
Manizales	5

Trilladora	Total Registros
Green Hills Coffee	4
Fnc	1

Y para concluir este análisis vamos a mostrar la cantidad de sacos despachados por tipo de café, para hacerlo nos vamos a valer de una gráfica el código para obtener dicha grafica lo mostramos en la Figura 30.

Figura 30

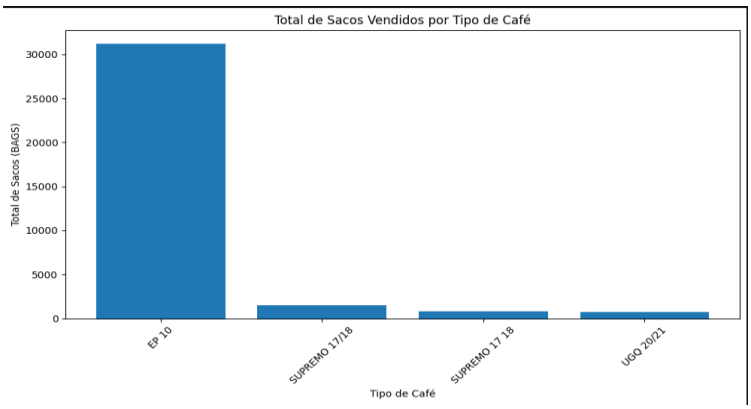
Código Para Graficar los Sacos Despachados por Tipo de Café

```
# Agrupar por tipo de café y sumar los sacos
bags_by_type = df1.groupby('TYPE')['BAGS'].sum().sort_values(ascending=False)
# Crear gráfico de barras
plt.figure(figsize=(10, 6))
plt.bar(bags_by_type.index, bags_by_type.values)
plt.xlabel('Tipo de Café')
plt.ylabel('Total de Sacos (BAGS)')
plt.title('Total de Sacos Vendidos por Tipo de Café')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

El resultado de ejecutar dicho código se muestra en la Figura 31.

Figura 31

Sacos Despachados por Tipo de Café



El análisis realizado sobre este conjunto de datos puede brindarle a la empresa un mayor conocimiento de su operación, permitiéndole optimizar rutas logísticas, identificar oportunidades comerciales y atraer nuevos clientes. Asimismo, este trabajo orienta la exploración analítica, al señalar qué tendencias y patrones deben ser buscados dentro de los datos disponibles.

Base de Datos

Luego de haber realizado, el proceso de depuración y transformación de los datos, procedemos, a crear una base de datos relacional, que será una parte muy importante del sistema de información que crearemos, la base de datos la codificamos con el script que se muestra en la Figura 32.

Figura 32

Script Para la Creación de la Base de Datos

```
CREATE DATABASE expoCafe
  CHARACTER SET utf8mb4
  COLLATE utf8mb4_unicode_ci;
```

A continuación, en la Figura 33 vemos el listado de bases de datos en el servidor local de MySQL, en el que podemos observar la base de datos ‘expocafe’, la cual se creo con el script que se mostro en la Figura 32.

Figura 33

Listado del Catálogo de Bases de Datos en Servidor Local de MySQL

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| agencia            |
| citas              |
| classicmodels      |
| expocafe           |
```

Posteriormente a eso, vamos a usar la herramienta jupyter notebook, para cargar el dataframe directamente en una tabla de la base de datos, creada anteriormente, para ello usamos el código que se muestra en la Figura 34.

Figura 34

Código Para Cargar el Dataframe de la Hoja 'Ventas' Como una Tabla en MySQL

```
import pandas as pd
from sqlalchemy import create_engine
df = pd.read_excel("Plan de Embarque 2022-2023.xlsx", sheet_name="Ventas",header=0)
df.columns=df.columns.str.strip()
# Conexión a MySQL (ajusta los valores)
usuario = 'root'
contraseña = '12345'
host = 'localhost'
puerto = '3306'
base_datos = 'expoCafe'
# Crear motor SQLAlchemy
engine = create_engine(f'mysql+pymysql://{usuario}:{contraseña}@{host}:{puerto}/{base_datos}')
# Convertir DataFrame a tabla (crear automáticamente si no existe)
df.to_sql(name='ventas', con=engine, if_exists='replace', index=False)
```

En la Figura 35 vemos en el lado izquierdo a la tabla ‘ventas’ como una de las tablas de la base de datos ‘expocafe’ y en el lado derecho el resultado de la ejecución de una consulta para conocer la cantidad de registros de la tabla.

Figura 35

Muestra la Tabla 'ventas' y la Cantidad de Registros de la Tabla

```
mysql> show tables;
+--------------------+
| Tables_in_expocafe |
+--------------------+
| plan_de_embarque   |
| ventas             |
+--------------------+
2 rows in set (0.03 sec)
```

```
mysql> SELECT COUNT(*) FROM ventas;
+----------+
| COUNT(*) |
+----------+
|      214 |
+----------+
1 row in set (0.01 sec)
```

El resultado de dicha consulta nos muestra que logramos cargar todas las filas del dataframe como registros en la tabla de la base de datos.

En la Figura 35 vemos una descripción parcial de los campos que contiene la tabla ‘ventas’ de la base de datos ‘expocafe’.

Figura 36

Descripción Parcial de la Tabla 'Ventas'

Field	Type	Null	Key	Default	Extra
ESTADO DE CONTRATO	text	YES		NULL	
REFERENCIA INTERNA	text	YES		NULL	
CONTRATO DE VENTA	text	YES		NULL	
FECHA DE ORDEN DE VENTA	text	YES		NULL	
IMPORTADOR / EXPORTADOR	text	YES		NULL	
FLO ID IMPORTADOR	double	YES		NULL	
PAIS	text	YES		NULL	
TIPO DE PRODUCTO	text	YES		NULL	
SACOS	double	YES		NULL	
UNIDAD DE EMPAQUE (Kg)	double	YES		NULL	
TIPO DE EMPAQUE	text	YES		NULL	
No DE EMPAQUE	double	YES		NULL	

En la tabla 37 vemos la ejecución de una consulta sobre la tabla ‘ventas’ de la base de datos ‘expocafe’.

Figura 37

Ejecución de una Consulta sobre la Tabla 'Ventas'

```
mysql> select `REFERENCIA INTERNA`,`CONTRATO DE VENTA`,`IMPORTADOR / EXPORTADOR`,`PAIS`
    -> from Ventas limit 20;
```

REFERENCIA INTERNA	CONTRATO DE VENTA	IMPORTADOR / EXPORTADOR	PAIS
2021-22-0021-TAP	75085-1	Volcafe USA	United States
2021-22-0022-TAP	75086	Volcafe USA	United States
2021-22-0023-TAP	75087-1	Volcafe USA	United States
2021-22-0028-TAP	CLB-8723	Meo-Fichaux	France
2021-22-0054-TAP	NPCO-18107-CLB_8782- MPEX-7340	Sucafina NA Inc	United States
2022-23-0001-TAP	CO220631-1	MCP Alliance Corp	Panama
2022-23-0001-TAP	CO220631-2	MCP Alliance Corp	Panama
2022-23-0001-TAP	CO220631-3	MCP Alliance Corp	Panama
2022-23-0002-TAP	P10228	Mitsui & Co	Colombia

Luego cargamos la información del conjunto de datos ‘Plan De Embarque’, en la tabla, ‘plan_de_embarque’ el código para ello se muestra en la Figura 38.

Figura 38

Código Para Cargar el Dataframe 'Plan De Embarque' como una Tabla en MySQL

```
import pandas as pd
from sqlalchemy import create_engine
df=pd.read_excel("Plan de Embarque 2022-2023.xlsx", sheet_name="PLAN DE EMBARQUE",header=0)
df.columns=df.columns.str.strip()
# Conexión a MySQL (ajusta los valores)
usuario = 'root'
contraseña = '12345'
host = 'localhost'
puerto = '3306'
base_datos = 'expoCafe'
# Crear motor SQLAlchemy
engine = create_engine(f'mysql+pymysql://{usuario}:{contraseña}@{host}:{puerto}/{base_datos}')
# Convertir DataFrame a tabla (crear automáticamente si no existe)
df.to_sql(name='plan_de_embarque', con=engine, if_exists='replace', index=False)
```

Posteriormente en la Figura 39 mostramos a la tabla ‘plan de embarque’ junto con la tabla ‘ventas’ que son las tablas de la base de datos ‘expocafe’

Figura 39

Muestra de la Tabla 'Plan de Embarque' en la Base de Datos 'Expocafe'

```
mysql> show tables;
+-------------------+
| Tables_in_expocafe |
+-------------------+
| plan_de_embarque   |
| ventas             |
+-------------------+
2 rows in set (0.03 sec)
```

En la Figura 40 mostramos una descripción parcial de los campos de la tabla ‘plan de embarque’, donde podemos ver el tipo de dato de algunos campos y su valor por defecto.

Figura 40

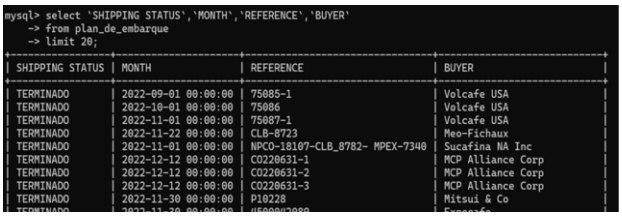
Descripción Parcial del Conjunto de Campos de la Tabla 'Plan de Embarque'

```
mysql> show columns from plan_de_embarque;
+-------------------------+-----------+------+-----+---------+-------+
| Field                   | Type      | Null | Key | Default | Extra |
+-------------------------+-----------+------+-----+---------+-------+
| MATERIA PRIMA  ASIGNADA | text      | YES  |     | NULL    |       |
| SHIPPING STATUS         | text      | YES  |     | NULL    |       |
| BROKER                  | text      | YES  |     | NULL    |       |
| MONTH                   | datetime  | YES  |     | NULL    |       |
| OUR REF                 | text      | YES  |     | NULL    |       |
| REFERENCE               | text      | YES  |     | NULL    |       |
| BUYER                   | text      | YES  |     | NULL    |       |
| BAGS                    | bigint    | YES  |     | NULL    |       |
| TYPE                    | text      | YES  |     | NULL    |       |
```

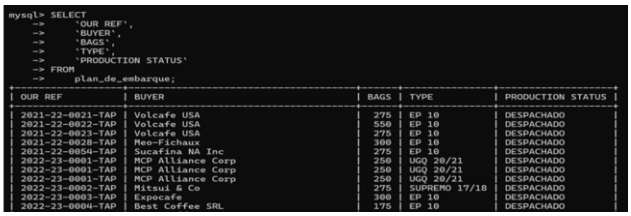
En la Figura 41 podemos observar la ejecución de algunas consultas en la tabla ‘plan de embarque para mostrar la funcionalidad de esta.

Figura 41

Muestra de la Ejecución de Consultas en la Tabla 'Plan de Embarque'



```
mysql> select `SHIPPING STATUS`,`MONTH`,`REFERENCE`,`BUYER`
    -> from plan_de_embarque
    -> limit 20;
+-----------------+---------------------+-----------------------------------+-------------------------+
| SHIPPING STATUS | MONTH               | REFERENCE                         | BUYER                   |
+-----------------+---------------------+-----------------------------------+-------------------------+
| TERMINADO       | 2022-09-01 00:00:00 | 75085-1                           | Volcafe USA             |
| TERMINADO       | 2022-10-01 00:00:00 | 75086                             | Volcafe USA             |
| TERMINADO       | 2022-11-01 00:00:00 | 75087-1                           | Volcafe USA             |
| TERMINADO       | 2022-11-22 00:00:00 | CLB-8723                          | Meo-Fichaux             |
| TERMINADO       | 2022-11-01 00:00:00 | NPCO-18107-CLB_8782- MPEX-7340    | Sucafina NA Inc         |
| TERMINADO       | 2022-12-12 00:00:00 | C0220631-1                        | MCP Alliance Corp       |
| TERMINADO       | 2022-12-12 00:00:00 | C0220631-2                        | MCP Alliance Corp       |
| TERMINADO       | 2022-12-12 00:00:00 | C0220631-3                        | MCP Alliance Corp       |
| TERMINADO       | 2022-11-30 00:00:00 | P10228                            | Mitsui & Co             |
```



```
mysql> SELECT
    ->     `OUR REF`,
    ->     `BUYER`,
    ->     `BAGS`,
    ->     `TYPE`,
    ->     `PRODUCTION STATUS`
    -> FROM
    ->     plan_de_embarque;
+-----------------+-------------------+------+-----------+-------------------+
| OUR REF         | BUYER             | BAGS | TYPE      | PRODUCTION STATUS |
+-----------------+-------------------+------+-----------+-------------------+
| 2021-22-0021-TAP | Volcafe USA       |  275 | EP 10     | DESPACHADO        |
| 2021-22-0022-TAP | Volcafe USA       |  550 | EP 10     | DESPACHADO        |
| 2021-22-0023-TAP | Volcafe USA       |  275 | EP 10     | DESPACHADO        |
| 2021-22-0028-TAP | Meo-Fichaux       |  300 | EP 10     | DESPACHADO        |
| 2021-22-0054-TAP | Sucafina NA Inc   |  275 | EP 10     | DESPACHADO        |
| 2022-23-0001-TAP | MCP Alliance Corp |  250 | UGQ 20/21 | DESPACHADO        |
| 2022-23-0001-TAP | MCP Alliance Corp |  250 | UGQ 20/21 | DESPACHADO        |
| 2022-23-0001-TAP | MCP Alliance Corp |  250 | UGQ 20/21 | DESPACHADO        |
| 2022-23-0002-TAP | Mitsui & Co       |  275 | SUPREMO 17/18 | DESPACHADO    |
| 2022-23-0003-TAP | Expocafe          |  300 | EP 10     | DESPACHADO        |
| 2022-23-0004-TAP | Best Coffee SRL   |  175 | EP 10     | DESPACHADO        |
```

Finalmente, para completar el desarrollo de la base de datos debemos crear claves primarias para las tablas de la base de datos, tal como se muestra en la Figura 42.

Figura 42

Creación de Claves Primarias Para las Tablas de la Base de Datos

```sql
ALTER TABLE ventas
  ADD COLUMN id INT NOT NULL AUTO_INCREMENT PRIMARY KEY FIRST;

ALTER TABLE plan_de_embarque
  ADD COLUMN id INT NOT NULL AUTO_INCREMENT PRIMARY KEY FIRST;
```

Aplicación de Algoritmos de Aprendizaje Supervisado

Después de llevar a cabo el proceso de depuración y transformación de los datos, así como un análisis exploratorio, se procede a aplicar algoritmos de *machine learning* sobre los conjuntos de datos. En el caso de la hoja ‘Ventas’, el objetivo es predecir si la cantidad de kilos en un contrato superará los 10.000 kg (la cantidad de kilos en la hoja ventas se llama ‘Kg Estándar’) mientras que en la hoja ‘Plan De Embarque’, se busca estimar el número de sacos por envío.

Hoja ‘Ventas’

Con el fin de garantizar que los algoritmos de aprendizaje automático generen resultados fiables y precisos, es necesario preparar el conjunto de datos, por ello a continuación preparamos los datos de la hoja ‘Ventas’ para aplicar algoritmos de Machine Learning, para ello, primero convertimos las fechas a formato datetime y extraemos el mes como una variable numérica; luego, creamos una variable binaria "Alto Volumen" para clasificar los envíos grandes; después, seleccionamos solo las columnas relevantes para reducir el ruido, codificamos las variables categóricas con get_dummies para que los modelos puedan procesar texto, y finalmente, eliminamos todos los valores nulos para asegurar que los datos estuvieran limpios y listos. De esta forma, nos aseguramos de que los datos se encuentren en un formato estructurado y limpio con el fin de mejorar la calidad del entrenamiento de tal manera que el algoritmo encuentre patrones reales y no ruido. A continuación, en la Figura 43, mostramos el algoritmo de preparación que fue descrito anteriormente.

Figura 43

Preparación de los Datos Para la Aplicación de Algoritmos de Machine Learning

```
# Convertir fechas y extraer el mes
df['MES EMBARQUE'] = pd.to_datetime(df['MES DE EMBARQUE'], errors='coerce')
df['MES'] = df['MES EMBARQUE'].dt.month
# Crear variable binaria de clasificación: alto volumen
df['ALTO VOLUMEN'] = df['KG ESTÁNDAR'] > 10000
# Filtrar columnas útiles
features = [
    'PAIS', 'INCOTERMS', 'TIPO DE EMPAQUE', 'SACOS', 'UNIDAD DE EMPAQUE (Kg)',
    'CERT', 'CALIDAD / PUNTAJE', 'MES', 'FACTOR CPS'
]
# Codificar variables categóricas
df_model = pd.get_dummies(df[features + ['KG ESTÁNDAR', 'ALTO VOLUMEN']], drop_first=True)
# Eliminar NaN si hay
df_model = df_model.dropna()
# Dividir features y targets
X_reg = df_model.drop(['KG ESTÁNDAR', 'ALTO VOLUMEN'], axis=1)
y_reg = df_model['KG ESTÁNDAR']
X_clf = X_reg.copy()
y_clf = df_model['ALTO VOLUMEN']
```

Algoritmo de Regresión lineal

Posteriormente a la preparación del conjunto de datos, el primer algoritmo que aplicamos es un algoritmo de regresión lineal para intentar predecir la cantidad de café exportado en kilogramos (KG Estándar), en función de variables como país, incoterm, tipo de empaque, número de sacos, unidad de empaque, certificación, puntaje de calidad, mes de embarque y el factor CPS. El objetivo es estimar el volumen exportado dado un conjunto de características logísticas y comerciales de cada embarque. A continuación, en la Figura 44 mostramos el algoritmo.

Figura 44

Algoritmo de Regresión Lineal Para Predecir la Cantidad de Café Exportado

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

# Dividir datos
X_train, X_test, y_train, y_test = train_test_split(X_reg, y_reg, test_size=0.2, random_state=42)

# Entrenar modelo
reg = LinearRegression()
reg.fit(X_train, y_train)

# Evaluar
y_pred = reg.predict(X_test)
print("MSE:", mean_squared_error(y_test, y_pred))
print("R² Score:", r2_score(y_test, y_pred))
```

En la Tabla 16 procedemos a mostrar los resultados del modelo de regresión lineal aplicado a la hoja ‘ventas’.

Tabla 16

Resultado de Evaluación del Modelo de Regresión Aplicado a la Hoja ‘Ventas’

Métrica	Valor
MSE	196862,51
R^2 Score	0,993871758

Dado que el modelo de regresión lineal busca ajustar una recta que represente la relación entre las variables predictoras y el valor objetivo, es posible obtener una aproximación visual de su precisión. Para ello usamos el código que se muestra en la Figura 45.

Figura 45

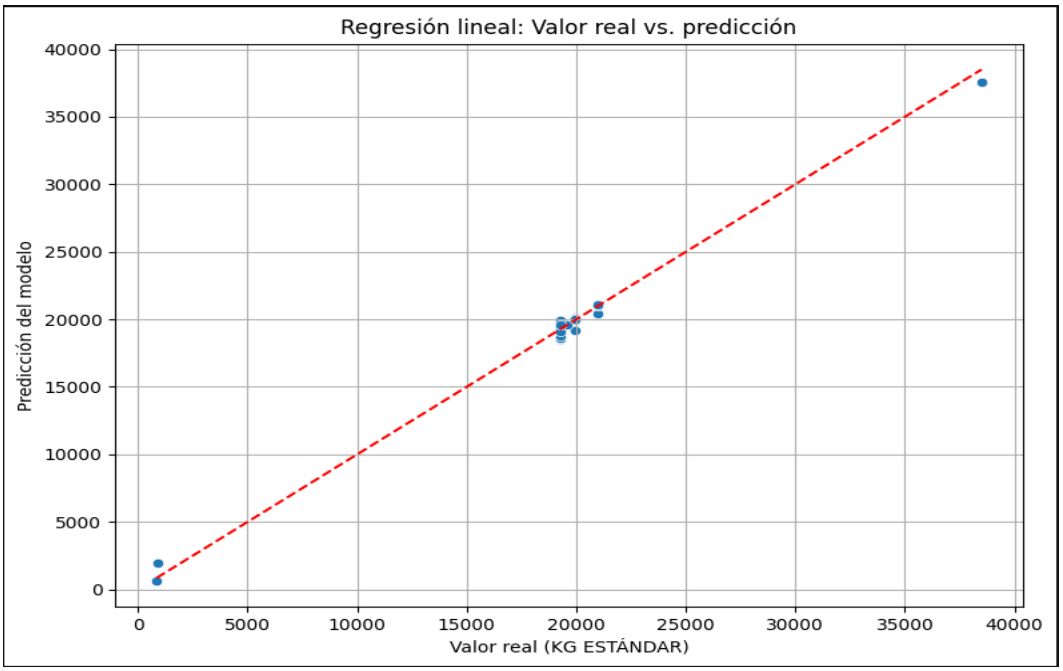
Código Para Graficar el Valor Real vs la Predicción que Hace la Regresión Lineal

```
plt.figure(figsize=(8,6))
sns.scatterplot(x=y_test, y=y_pred)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color='red', linestyle='--')
plt.xlabel('Valor real (KG ESTÁNDAR)')
plt.ylabel('Predicción del modelo')
plt.title('Regresión lineal: Valor real vs. predicción')
plt.grid(True)
plt.tight_layout()
plt.show()
```

En la Figura 46 se observa cómo las predicciones del modelo se alinean con los valores reales, lo que permite evaluar gráficamente qué tan bien la recta ajustada representa el comportamiento del conjunto de datos. Esta visualización refuerza la idea de que el modelo logra capturar adecuadamente la tendencia general de los datos.

Figura 46

Graficar del Valor Real vs la Predicción que Hace la Regresión Lineal



Como se observa en la Tabla 16 y en la Figura 46, el modelo presenta un desempeño destacado. Predice con alta precisión el peso (‘KG Estándar’) del embarque a partir de variables logísticas y de calidad. El error absoluto es bajo en comparación con los valores típicos del conjunto de datos, lo que indica una buena capacidad predictiva. Estos resultados sugieren que las variables seleccionadas son relevantes y que el modelo logra capturar de manera efectiva los patrones presentes en los datos.

Algoritmo Árbol de Decisión

Tras analizar el comportamiento del peso de los embarques mediante un modelo de regresión lineal, el cual mostró un desempeño sobresaliente al predecir con alta precisión el valor de la variable ‘Kg Estándar’, se procedió a abordar este mismo fenómeno desde una perspectiva distinta: la clasificación binaria.

A continuación, se aplicó un algoritmo de árbol de decisión, el cual se muestra en la Figura 47, con el objetivo de predecir la variable categórica ‘Alto Volumen’, la cual indica si un embarque supera los 10.000 kg. Este modelo permite identificar de forma clara qué factores (como el país de destino, el tipo de empaque o el mes de embarque) influyen en que un envío sea considerado de alto volumen, lo que facilita la toma de decisiones logísticas y comerciales para la empresa.

Figura 47

Algoritmo de Árbol de Decisión Aplicado a la Hoja 'Ventas'

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix

X_train_clf, X_test_clf, y_train_clf, y_test_clf = train_test_split(X_clf, y_clf, test_size=0.2, random_state=42)

clf = DecisionTreeClassifier(max_depth=4)
clf.fit(X_train_clf, y_train_clf)

y_pred_clf = clf.predict(X_test_clf)

print(confusion_matrix(y_test_clf, y_pred_clf))
print(classification_report(y_test_clf, y_pred_clf))
```

En la tabla 17 mostramos la matriz de confusión del modelo de árbol de decisión aplicado a la hoja ‘ventas’.

Tabla 17

Matriz de Confusión del Modelo de Árbol de Decisión Aplicado a la Hoja ‘Ventas’

	Predicción: False	Predicción: True
Real: False	2	0
Real: True	0	31

En la Tabla 18 mostramos las métricas de evaluación del modelo de Árbol de Decisión.

Tabla 18

Métricas de Evaluación del Modelo de Árbol de Decisión

Clase	Precisión	Recall	F1-Score	Soporte
False	1.00	1.00	1.00	2
True	1.00	1.00	1.00	31
Accuracy			**1.00**	33
Macro Avg	1.00	1.00	1.00	33
Weighted Avg	1.00	1.00	1.00	33

Este resultado indica que el modelo clasificó correctamente todos los embarques del conjunto de prueba como de alto o bajo volumen, alcanzando una precisión y exactitud del 100 %. La matriz de confusión muestra que no se cometieron errores: se identificaron correctamente los 2 casos de bajo volumen y los 31 casos de alto volumen. Esto sugiere un desempeño excelente del modelo. Sin embargo, dado el bajo número de ejemplos en la clase minoritaria (bajo volumen),

sería recomendable validar el modelo con un conjunto de datos más amplio para confirmar su fiabilidad. Por el momento, no se dispone de información adicional por parte de la empresa.

Algoritmo de los k-Vecinos más Cercanos

Después de aplicar un modelo de regresión lineal para predecir el valor continuo de Kg Estándar y de utilizar un árbol de decisión para clasificar los embarques según si superan o no los 10.000 kg (‘Alto Volumen’), se implementó un tercer enfoque: el algoritmo de los *k vecinos más cercanos* (k-NN).

Este modelo de clasificación se fundamenta en la similitud entre observaciones, asignando a cada nuevo registro la clase predominante entre sus *k* vecinos más cercanos en el espacio de variables. En la Figura 48 se presenta la implementación de este algoritmo, utilizando un valor de $k = 3$ y las mismas variables predictoras empleadas en los modelos anteriores. Los resultados, reflejados en la matriz de confusión y el reporte de clasificación, permiten comparar el desempeño del k-NN con el del árbol de decisión, y evaluar su eficacia para identificar correctamente los embarques de alto volumen.

Figura 48

Algoritmo de los K-Vecinos más Cercanos (K-Nearest Neighbors, KNN)

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix

# Dividir el dataset (ya hecho previamente)
X_train_clf, X_test_clf, y_train_clf, y_test_clf = train_test_split(X_clf, y_clf, test_size=0.2, random_state=42)

# Crear modelo KNN
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train_clf, y_train_clf)

# Predicciones
y_pred_knn = knn.predict(X_test_clf)

# Evaluación
print(confusion_matrix(y_test_clf, y_pred_knn))
print(classification_report(y_test_clf, y_pred_knn))
```

En la Tabla 19 mostramos la matriz de confusión del modelo k-vecinos más cercanos (k-NN) aplicado a la hoja 'ventas'.

Tabla 19

Matriz de Confusión del Modelo K-Vecinos Aplicado a la Hoja 'Ventas'

	Predicción: False	Predicción: True
Real: False	2	0
Real: True	0	31

En la Tabla 20 mostramos las métricas de evaluación del modelo k-vecinos más cercanos (k-nn) aplicado a la hoja 'ventas'.

Tabla 20

Métricas de Evaluación del Modelo K-Vecinos Aplicado a la Hoja 'Ventas'

Clase	Precisión	Recall	F1-Score	Soporte
False	1.00	1.00	1.00	2
True	1.00	1.00	1.00	31
Accuracy			**1.00**	33
Macro avg	1.00	1.00	1.00	33
Weighted avg	1.00	1.00	1.00	33

Después de aplicar el algoritmo y analizar los resultados, vemos que el algoritmo de *k-Vecinos más Cercanos* (k-NN), con un valor de $k = 3$, logró clasificar correctamente la totalidad de los embarques del conjunto de prueba, identificando sin errores tanto los casos de alto

volumen como los de bajo volumen. Alcanzó una precisión, recall y F1-score del 100 % en ambas clases, lo que refleja una capacidad sobresaliente para distinguir entre ellas. Este resultado sugiere que las variables utilizadas capturan de forma adecuada los patrones presentes en los datos, y que el algoritmo tiene potencial para convertirse en una herramienta eficaz en el apoyo a decisiones operativas.

A continuación, en la Tabla 21, se presenta una comparación de los resultados obtenidos tras la aplicación de los distintos algoritmos al conjunto de datos.

Tabla 21

Resumen Comparativo Del Desempeño De Los Algoritmos Aplicados

Modelo	Tipo	Variable objetivo	Métrica principal	Valor obtenido	Conclusión
Regresión Lineal	Regresión	KG Estándar	R^2 Score	0.99	Excelente ajuste; el modelo predice con alta precisión el volumen estándar exportado.
Árbol de Decisión	Clasificación	Alto Volumen (True/False)	Accuracy / F1-score	100 %	Clasificó correctamente todos los casos; modelo simple,

Modelo	Tipo	Variable objetivo	Métrica principal	Valor obtenido	Conclusión
					interpretable y efectivo.
K-Nearest Neighbors (k = 3)	Clasificación	Alto Volumen (True/False)	Accuracy / F1-score	100 %	Igual de preciso que el árbol; sin embargo, es más sensible a variaciones en los datos.
Comparación General	—	—	—	—	La regresión ofrece predicción numérica precisa; el árbol es ideal para clasificación robusta y rápida; k-NN es sensible pero útil para validar resultados.

Nota. Esta tabla muestra el resumen comparativo del desempeño de los algoritmos

En función de los objetivos operativos de la empresa, si se busca estimar con precisión el peso exacto de los embarques, se recomienda utilizar el modelo de regresión lineal. Para decisiones rápidas basadas en si el volumen supera un umbral determinado, el árbol de decisión es la mejor opción por su simplicidad, interpretabilidad y precisión. Por su parte, el algoritmo k-NN puede emplearse como modelo complementario o de validación, aunque su sensibilidad a pequeñas variaciones en los datos sugiere precaución en contextos con alta variabilidad.

Hoja ‘Plan De Embarque’

Tras analizar un el conjunto de datos de la hoja 'Ventas', pasaremos a realizar un proceso similar sobre la hoja 'PLAN DE EMBARQUE', este segundo conjunto de datos detalla la información de los embarques de café, que comercializa la empresa.

Algoritmo de Regresión Lineal

El objetivo que nos planteamos para este conjunto de datos es predecir la cantidad de sacos (BAGS) en cada embarque, y para ello se seleccionaron variables que, según nuestro criterio, podrían influir en ese resultado: el mes del contrato (MONTH_NUM), el comprador (BUYER), el tipo de producto (TYPE) y el laboratorio asignado (LABORATORY).

A partir de estas variables, se preparó el conjunto de datos a partir de los siguientes pasos: conversión de fechas a valores numéricos, selección de columnas relevantes, codificación de variables categóricas mediante one-hot encoding, y división del conjunto en entrenamiento y prueba, dicho proceso se detalla en la Figura 49.

Figura 49

Preparación Del Conjunto De Datos De La Hoja 'Plan De Embarque'

```
# 1. Extraer información temporal: convertir fechas en números de mes
df1['MONTH_NUM'] = df1['MONTH'].dt.month             # Mes del contrato
df1['DELIVERY_MONTH'] = df1['DATE DELIVERY'].dt.month  # Mes de entrega (puede usarse más adelante)

# 2. Seleccionar columnas relevantes para el modelo
# Estas variables fueron elegidas porque pueden influir en la cantidad de sacos exportados:
# - MONTH_NUM: refleja estacionalidad o demanda por mes
# - BUYER: cada comprador puede tener distintos volúmenes
# - TYPE: el tipo de café influye en el volumen del pedido
# - LABORATORY: puede relacionarse con certificaciones o estándares
# - BAGS: variable objetivo (número de sacos a predecir)
reg_cols = ['MONTH_NUM', 'BUYER', 'TYPE', 'LABORATORY', 'BAGS']
reg_df1 = df1[reg_cols].dropna()  # Elimina filas con valores nulos para evitar errores en el modelo

# 3. Codificar variables categóricas
# Las variables categóricas ('BUYER', 'TYPE', 'LABORATORY') deben ser convertidas a numéricas
# Se usa one-hot encoding, eliminando la primera categoría de cada variable para evitar multicolinealidad
reg_df1_encoded = pd.get_dummies(reg_df1.drop(columns=['BAGS']), drop_first=True)
# 4. Separar variables independientes y dependiente
X = reg_df1_encoded
y = reg_df1['BAGS']
```

Con el conjunto de datos ya preparado, se aplica un modelo de regresión lineal para predecir la cantidad de sacos (BAGS) en cada embarque, utilizando las variables seleccionadas. Los datos se dividen en entrenamiento y prueba, y se evalúa el modelo mediante el MAE y el R^2. En la Figura 50 se muestra el algoritmo de regresión lineal.

Figura 50

Algoritmo de Regresión Lineal Para Predecir la Cantidad de Sacos Bags

```
# 4. Dividir en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# 5. Entrenar el modelo de regresión lineal
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
# 6. Realizar predicciones
y_pred = lin_reg.predict(X_test)
# 7. Evaluar el modelo
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
# 8. Mostrar resultados
print("📉 Error absoluto medio (MAE):", round(mae, 2))
print(f"📈 Coeficiente de determinación (R²): {r2:.4f}")
# 9. Mostrar coeficientes del modelo
coeficientes = pd.Series(lin_reg.coef_, index=X.columns)
print("\n📊 Coeficientes del modelo de regresión lineal:")
print(coeficientes.sort_values(ascending=False))
# 10. Comparar valores reales vs predichos
resultados = pd.DataFrame({
    'Valor real': y_test,
    'Predicción': y_pred.round(2)
})
print("\n🔍 Muestra de predicciones:")
print(resultados.head(10))
```

En la Tabla 22 se muestran las métricas de desempeño del algoritmo de regresión lineal aplicado a la hoja ‘plan de embarque’.

Tabla 22

Métricas de Desempeño del Algoritmo de Regresión Hoja 'Plan de Embarque'

Métrica	Valor
Error Absoluto Medio (MAE)	30.72
Coeficiente de Determinación (R^2)	0.4985

Vemos que el modelo de regresión lineal obtuvo un error absoluto medio (MAE) de 30.72, lo que indica que, en promedio, se equivoca por aproximadamente 31 sacos al predecir la cantidad exportada por embarque. Dado que los embarques tienen en promedio 256 sacos, este

error representa cerca del 12 %, lo cual puede considerarse razonable.

Además, el coeficiente de determinación (R^2) fue de 0.4985, lo que indica que el modelo logra explicar casi el 50 % de la variabilidad en los datos, lo que podría considerarse un desempeño aceptable para un primer modelo lineal con variables categóricas codificadas.

Algoritmo Árbol de Decisión

A continuación, implementamos un modelo de árbol de decisión para regresión con el fin de predecir la cantidad de sacos (BAGS) por embarque. En este caso, la preparación del conjunto de datos se realiza directamente en el mismo bloque de código, como se muestra en la Figura 51. Utilizamos las variables MONTH_NUM, BUYER, TYPE y LABORATORY como predictores, y limitamos la profundidad del árbol a 5 para evitar sobreajuste. Este enfoque permite modelar relaciones no lineales entre las variables y evaluar si mejora el desempeño frente al modelo de regresión lineal aplicado previamente.

Figura 51

Algoritmo de Árbol de Decisión Aplicado a la Predicción de Sacos

```
# 1. Preparar los datos
reg_cols = ['MONTH_NUM', 'BUYER', 'TYPE', 'LABORATORY', 'BAGS']
reg_df1 = df1[reg_cols].dropna()
# 2. Codificar variables categóricas
reg_df1_encoded = pd.get_dummies(reg_df1.drop(columns=['BAGS']), drop_first=True)
# 3. Separar variables predictoras (X) y objetivo (y)
X = reg_df1_encoded
y = reg_df1['BAGS']
# 4. Dividir en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# 5. Crear y entrenar el modelo de árbol de decisión
tree_reg = DecisionTreeRegressor(max_depth=5, random_state=42)
tree_reg.fit(X_train, y_train)
# 6. Realizar predicciones
y_pred = tree_reg.predict(X_test)
# 7. Evaluar el modelo
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("🌳 Árbol de decisión para regresión:")
print(f"📉 MAE (Error absoluto medio): {mae:.2f}")
print(f"📈 R² (Coeficiente de determinación): {r2:.4f}")
```

En la Tabla 23 mostramos las métricas de evaluación del árbol de decisión para regresión en la hoja ‘plan de embarque’.

Tabla 23

Métricas de Evaluación del Árbol de Decisión Hoja ‘Plan de Embarque’

Métrica	Valor
Error absoluto medio (MAE)	32.11
Coeficiente de Determinación (R^2)	0.4042

El modelo de árbol de decisión logra un desempeño aceptable al predecir la cantidad de sacos exportados, con un error absoluto medio de aproximadamente 32 sacos. Sin embargo, su coeficiente de determinación ($R^2 = 0.4042$) indica que solo logra explicar el 40 % de la variabilidad en los datos, lo que sugiere una capacidad predictiva moderada.

Si lo comparamos con el modelo de regresión lineal, que obtuvo un MAE de 30.72 y un R^2 de 0.4985, podemos observar que:

- La regresión lineal tiene un error ligeramente menor, lo que significa que en promedio se acerca un poco más a los valores reales.
- El R^2 también es superior en el modelo lineal, lo que indica que explica mejor la variación en los datos de entrenamiento.

A pesar de esto, el árbol de decisión tiene la ventaja de capturar relaciones no lineales y ofrecer interpretabilidad a través de su estructura jerárquica.

Para complementar la comparación y reforzar la evaluación de modelos, procederemos a aplicar un tercer algoritmo: K-Vecinos más Cercanos (k-NN), lo cual permitirá contrastar enfoques basados en similitud con los modelos ya analizados.

Algoritmo de los k-Vecinos más Cercanos

Después de evaluar los modelos de regresión lineal y árbol de decisión en la predicción de la cantidad de sacos exportados por embarque, se procede a aplicar un tercer enfoque: el algoritmo de K-Vecinos más Cercanos (k-NN). Este modelo se basa en la idea de que observaciones similares tienden a tener resultados similares; por tanto, predice el valor objetivo a partir del promedio de los k registros más cercanos en el espacio de variables. Su inclusión en el análisis permite contrastar los enfoques lineales y basados en reglas con uno centrado en la similitud, lo cual enriquece la comparación y puede ofrecer nuevas perspectivas sobre el comportamiento de los datos.

A continuación, en la Figura 52 implementamos el algoritmo de K-Vecinos más Cercanos (k-NN).

Figura 52

Algoritmo de K-Vecinos más Cercanos Aplicado a la Hoja 'Plan de Embarque'

```
# Seleccionar columnas relevantes
reg_cols = ['MONTH_NUM', 'BUYER', 'TYPE', 'LABORATORY', 'BAGS']
reg_df1 = df1[reg_cols].dropna()
# Codificar variables categóricas
reg_df1_encoded = pd.get_dummies(reg_df1.drop(columns=['BAGS']), drop_first=True)
# Separar variables predictoras (X) y objetivo (y)
X = reg_df1_encoded
y = reg_df1['BAGS']
# 2. Dividir en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# 3. Crear y entrenar el modelo k-NN para regresión
knn_reg = KNeighborsRegressor(n_neighbors=3)
knn_reg.fit(X_train, y_train)
# 4. Realizar predicciones
y_pred = knn_reg.predict(X_test)
# 5. Evaluar el modelo
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("🔍 Modelo k-NN para regresión:")
print(f"📉 Error absoluto medio (MAE): {mae:.2f}")
print(f"📈 Coeficiente de determinación (R²): {r2:.4f}")
# 6. Comparar valores reales vs. predichos (opcional)
resultados = pd.DataFrame({
    'Valor real': y_test,
    'Predicción': y_pred.round(2)
})
print("\n🟢 Muestra de predicciones:")
print(resultados.head(10))
```

En la Tabla 24 se muestran las métricas de evaluación del algoritmo k-nn para regresión aplicado a la hoja 'plan de embarque'

Tabla 24

Métricas de Evaluación del Algoritmo k-NN Hoja 'Plan de Embarque'

Métrica	Valor
Error absoluto medio (MAE)	34.37
Coeficiente de determinación (R^2)	0.2650

En la tabla 25 se muestran los valores reales del conjunto de datos y la predicción que se logra con el algoritmo.

Tabla 25

Muestra De Valores Reales Y Predicciones Del Modelo

Índice	Valor real	Predicción
58	275	275.00
61	275	275.00
7	250	258.33
69	32	16.67
37	285	281.67
78	285	281.67
87	275	275.00
13	275	275.00
54	275	275.00
121	130	205.00

El modelo de regresión k-NN con k=3 presentó un error absoluto medio (MAE) de 34.37 y un coeficiente de determinación (R^2) de 0.2650, lo que indica una capacidad moderada para explicar la variabilidad de los datos. Aunque el MAE es comparable al de los modelos anteriores, el valor de R^2 es el más bajo de los tres algoritmos evaluados, lo que sugiere que el modelo tiene menor poder explicativo. Además, al observar las predicciones, se nota una tendencia del modelo a aproximar los valores hacia promedios cercanos, lo que limita su capacidad para detectar correctamente valores atípicos o extremos. En comparación con la regresión lineal ($R^2 \approx 0.50$) y el árbol de decisión ($R^2 \approx 0.40$), el k-NN resultó ser menos preciso y explicativo, por lo que no sería la primera opción.

A continuación, en la Tabla 26, se presenta una comparación de los resultados obtenidos al aplicar los algoritmos de Regresión Lineal, Árbol de Decisión y k-Vecinos más Cercanos, para estimar la cantidad de sacos exportados. Esta evaluación considera métricas clave como el error absoluto medio (MAE) y el coeficiente de determinación (R^2), permitiendo identificar el modelo con mejor desempeño predictivo.

Tabla 26

Evaluación De Los Algoritmos Para Estimar La Cantidad De Sacos Exportados

Modelo	MAE (Error Absoluto Medio)	R^2 (Coeficiente de Determinación)	Conclusión
Regresión Lineal	30.72	0.4985	Mejor desempeño general, buen ajuste lineal a los datos, captura bien la relación entre variables.

Modelo	MAE (Error Absoluto Medio)	R^2 (Coeficiente de Determinación)	Conclusión
Árbol de Decisión	32.11	0.4042	Buen rendimiento, ligeramente menor que la regresión lineal. Ventaja: interpreta relaciones no lineales.
K-Nearest Neighbors	34.37	0.2650	Peor desempeño en términos de R^2. Predicciones más planas y menos sensibles a extremos.

Nota. Esta tabla muestra el resumen comparativo del desempeño de los algoritmos

Dado el equilibrio entre precisión (MAE) y capacidad explicativa (R^2), el modelo de regresión lineal es el más adecuado para predecir la cantidad de sacos en un embarque. Su bajo error promedio y mejor ajuste lo convierten en una herramienta confiable para la toma de decisiones operativas basadas en predicción de volumen exportado.

Implementación de Visualizaciones de Datos Relevantes

La implementación de los gráficos relevantes la realizaremos en una aplicación web desarrollada usando el framework Django, a continuación, mostramos la pantalla de inicio de la aplicación de la empresa exportadora, en la Figura 53.

Figura 53

Página de Inicio de la Aplicación en Django

La imagen inferior la Figura 54, nos muestra la página que nos permite acceder a la información de las ventas de la empresa.

Figura 54

Pagina Correspondiente a la Información de las Ventas de la Empresa

Como podemos ver en la imagen superior, la página permite consultar los datos correspondientes, a las ventas de la empresa y algunos gráficos de datos relevantes de la misma, a continuación, en la Figura 55, mostramos lo que sucede cuando hacemos clic en el botón ver datos.

Figura 55

Figura que Muestra la Información de las Ventas de la Empresa

REFERENCIA INTERNA	CONTRATO DE VENTA	IMPORTADOR / EXPORTADOR	PAÍS	TIPO DE PRODUCTO	SACOS	UNIDAD DE EMPAQUE
2021-22-0021-TAP	75085-1	Volcafe USA	United States	Excelso	275.0	70.0
2021-22-0022-TAP	75086	Volcafe USA	United States	Excelso	550.0	70.0
2021-22-0023-TAP	75087-1	Volcafe USA	United States	Excelso	275.0	70.0
2021-22-0028-TAP	CLB-8723	Mec-Eichaux	France	Excelso	300.0	70.0

En la Figura 56, se muestra el botón ‘Editar’ que aparece al final de la ventana al hacer clic en este botón podemos editar el registro en la base de datos.

Figura 56

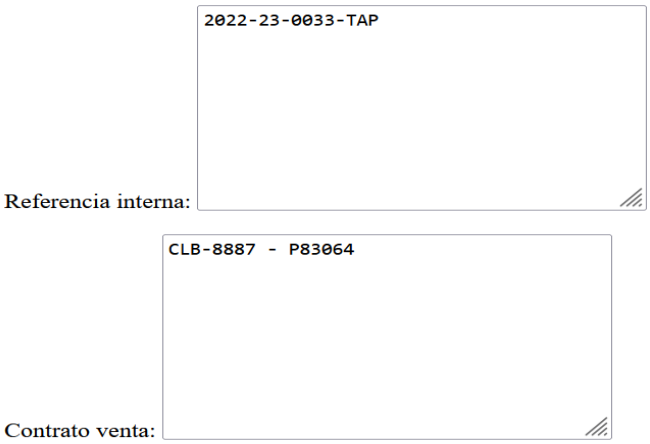
Muestra Como Editar los Registro de la Base de Datos

INCOTERMS	LABORATORIO	CALIDAD / PUNTAJE	FACTOR CPS	TERMINO DE PRECIO	POSICIÓN DE PRECIO	ACCIONES
FOB-BUN	None	None	0.0	None	None	Editar
FOB-BUN	None	None	0.0	None	None	Editar
FOB-BUN	None	None	0.0	ABIERTO	MARCH	Editar

En la Figura 57, se muestra el formulario que se utiliza para modificar el registro, el cual dispone de un botón de guardar que al hacer clic en el modifica el registro en la base de datos.

Figura 57

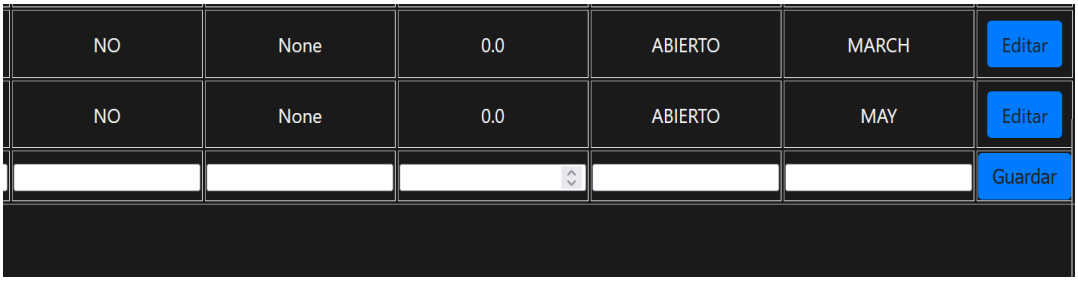
Formulario para Editar el Registro de la Base de Datos



En la figura 49, se muestra la forma en la que se ingresan nuevos registros a la base de datos, se llenan los campos y al final se hace clic en el botón guardar.

Figura 58

Muestra Como Agregar Nuevos Registros en la Base de Datos



A continuación, en la Figura 59 se presentan las pruebas realizadas para verificar el ingreso de registros en el sistema de información, los cuales quedaron almacenados correctamente en la base de datos.

Figura 59

Verificación Del Ingreso Exitoso De Registros En La Base De Datos

2022-23-0012-TAP	CLB-8858 - NPCO-18659	Sucafina NA Inc	Dinamarca	Excelso	275.0
2025-07-VENTA-001	CV-87532-A	Coffee Import Ltd.	Germany	Excelso	320

2025-07-VENTA-001	CV-87532-A	Coffee Import Ltd.	Germany	Excelso	320.0	None
2025			Noruega	Excelso	280.0	None

REF-TEST-001	CV-2025-07	Sucafina NA Inc	Noruega	Excelso	280.0	None
REF-TEST-002					216.0	None
2025-22-0011					270.0	None
TEST001	None	None	None	None	None	None
TEST002	C-12345	Café Global	Colombia	Arábica Premium	275.0	None

REFERENCIA INTERNA	CONTRATO DE VENTA	IMPORTADOR / EXPORTADOR	PAIS	TIPO DE PRODUCTO	SACOS
TEST002	C-12345	Café Global	Colombia	Arábica Premium	275
TEST001	NULL	NULL	NULL	NULL	NULL
2025-22-0011					270
REF-TEST-002					216
REF-TEST-001	CV-2025-07	Sucafina NA Inc	Noruega	Excelso	280
2025			Noruega	Excelso	280

A continuación, en la Figura 60 se muestra, la página que muestra la información correspondiente a los embarques de la empresa tiene un funcionamiento similar a la página anterior.

Figura 60

Muestra la Información Correspondiente a los Datos de Embarque

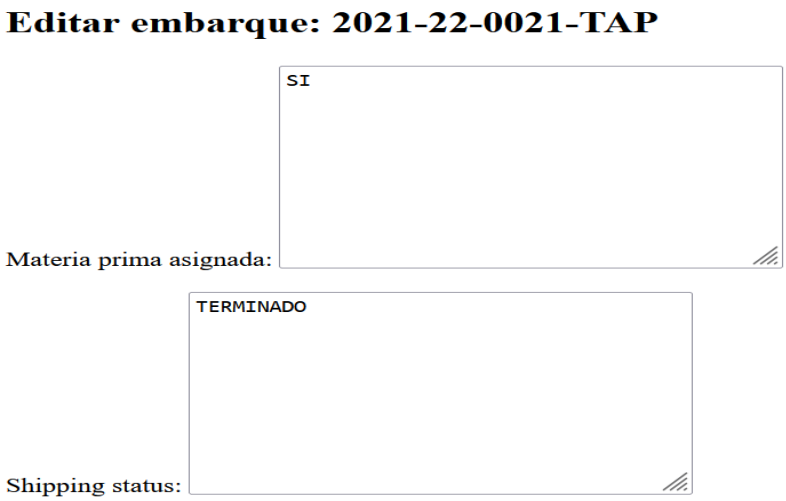
MATERIA PRIMA ASIGNADA	SHIPPING STATUS	BROKER	MONTH	OUR REF	REFERENCE
SI	TERMINADO	None	Sept. 1, 2022, midnight	2021-22-0021-TAP	75085-1
SI	TERMINADO	None	Nov. 30, 2022, midnight	2022-23-0013-TAP	4500042111

FIXATION NY SUPPLIER	TRANSPORADORA	FECHA DE GUIA	CONTRATOS PROVEEDOR	PROVEEDOR	ACCIONES
SI		Sept. 14, 2023, midnight	None	None	Editar
SI		None	None	None	Editar

La Figura 61 muestra el formulario para modificar los registros en la base de datos, una vez se hace clic en guardar.

Figura 61

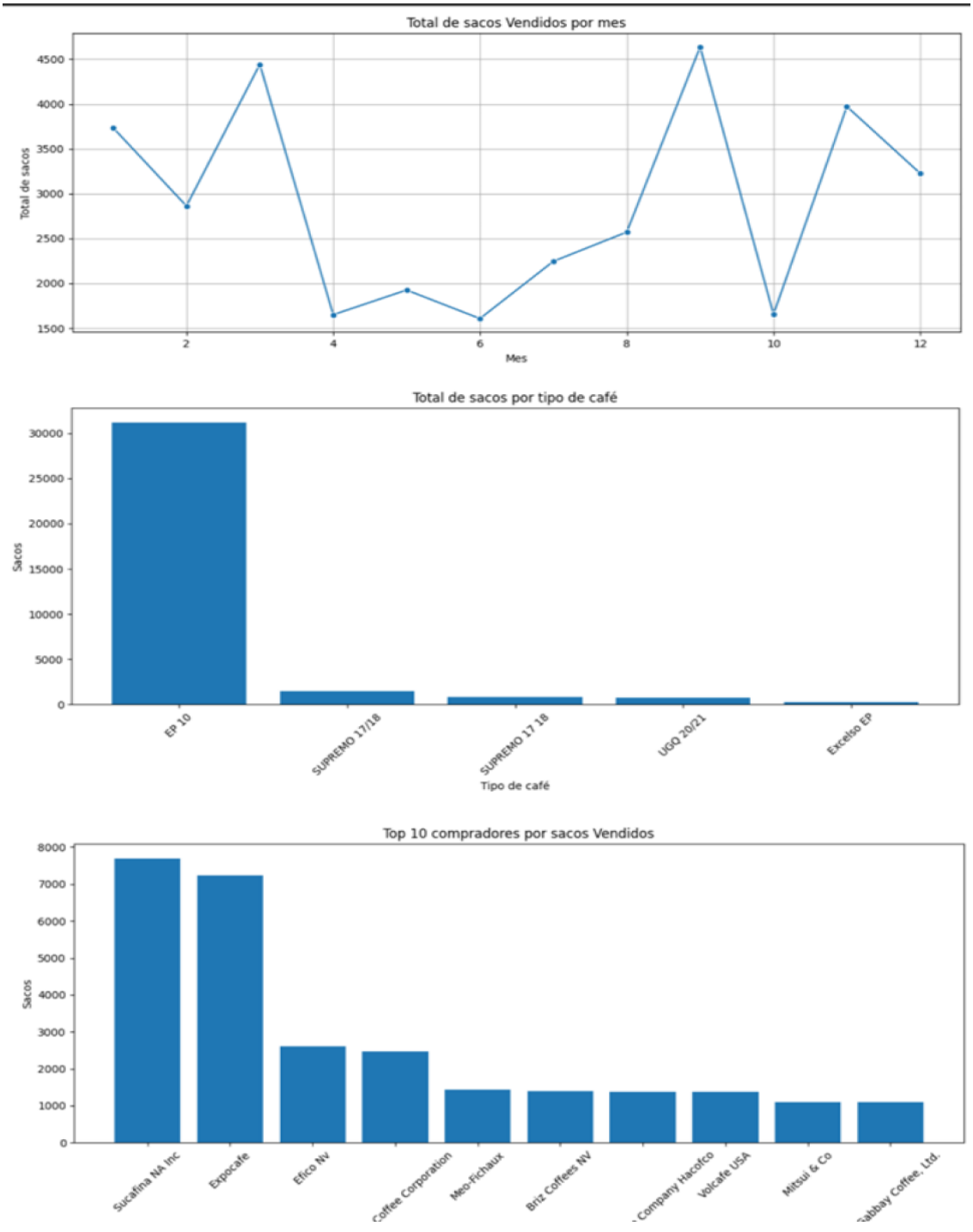
Formulario Para Modificar los Registros Correspondientes a los Embarques



En la Figura 53, se muestran los gráficos relevantes del conjunto de datos almacenados en la base de datos, los cuales se obtienen al hacer clic en el botón ver graficas, en los mismos se ve la tendencia de exportación mensual, el tipo de café que más se vende, y los principales compradores de café, lo cuales son indicadores relevantes para tener en cuenta en las operaciones de la empresa.

Figura 62

Gráficos Relevantes de los Datos de la Empresa Exportadora de Café



Conclusiones

Durante el proceso de depuración y transformación de los datos históricos de la empresa —que incluyó la carga, comprensión y tratamiento de la información— se evidenció el gran potencial de la biblioteca pandas para gestionar conjuntos de datos complejos. Esta herramienta facilitó una exploración profunda de las variables relacionadas con el comercio de café, especialmente en lo referente a las cantidades vendidas en kilos y sacos.

Del análisis realizado, se concluye que el modelo de negocio de la empresa se basa, en gran medida, en la venta a grandes compradores, muchos de ellos ubicados fuera de Colombia. No obstante, también se identificaron clientes nacionales y operaciones con compradores de menor escala, lo cual se refleja en envíos reducidos —por ejemplo, de 9 sacos— empacados en unidades de 20 kg, a diferencia del empaque estándar de 70 kg.

Entre los principales hallazgos, destaca que Estados Unidos es el principal destino internacional de las exportaciones, con una ruta predominante que parte del puerto de Barranquilla y llega al puerto de New York. La mayoría de los embarques se realizan en sacos de fique, certificados bajo los sellos FTO (Fair Trade Organic) y Orgánico, lo que evidencia un enfoque en prácticas sostenibles y comercio justo.

En la hoja ‘PLAN DE EMBARQUE’, que contiene los registros de los despachos asociados a las ventas, se identificó que la empresa transportadora predominante es CODETRANS, cuyo principal destino portuario también es Barranquilla. La línea naviera más utilizada es MSC, y la ruta más frecuente, como era previsible, es Barranquilla–New York. Además, se observó una concentración significativa de envíos durante el mes de mayo.

Toda esta información, adecuadamente procesada y analizada, constituye una herramienta valiosa para que los responsables de la gestión comercial de la empresa comprendan mejor el comportamiento de su operación y puedan tomar decisiones fundamentadas en datos.

En cuanto a la aplicación de algoritmos de machine leargning, al conjunto de datos de la hoja denominada 'Ventas', todos demostraron un alto desempeño tanto en tareas de regresión como de clasificación. La regresión lineal obtuvo un R^2 de 0.99, lo que indica una excelente capacidad para predecir con precisión el volumen kilogramos exportado. En cuanto a la clasificación del volumen como alto o no, tanto el Árbol de Decisión como el modelo K-Nearest Neighbors (k = 3) alcanzaron una precisión del 100%. Sin embargo, el Árbol de Decisión se destaca por su simplicidad, robustez e interpretabilidad, mientras que k-NN, aunque igual de preciso en este caso, es más sensible a cambios en los datos.

En conjunto, se concluye que la regresión lineal es ideal para estimaciones numéricas, el Árbol de Decisión es la opción más adecuada para tareas de clasificación por su estabilidad y claridad, y k-NN puede ser útil como modelo complementario para validar resultados. Estos modelos ofrecen un sólido respaldo para apoyar la toma de decisiones operativas en la empresa.

En cuanto al conjunto de datos de la hoja denominada ‘PLAN DE EMBARQUE’, La regresión lineal mostró el mejor desempeño general, con el menor error absoluto medio (MAE = 30.72) y el mayor coeficiente de determinación (R^2 = 0.4985), lo que indica un buen ajuste a la relación lineal entre las variables. El árbol de decisión presentó un rendimiento cercano, con la ventaja de capturar relaciones no lineales, aunque con un ajuste ligeramente inferior (R^2 = 0.4042). Por su parte, el modelo k-Nearest Neighbors obtuvo el desempeño más bajo (R^2 = 0.2650), mostrando menor sensibilidad a los valores extremos. En conjunto, la regresión lineal se posiciona como la opción más confiable para la predicción numérica en este caso.

El desarrollo del sistema de información demostró que es posible integrar Django con las bibliotecas especializadas de Python para el análisis de datos, logrando una solución funcional y eficiente para la gestión y visualización de la información operativa.

En particular, se evidenció que bibliotecas como Pandas, Matplotlib y Seaborn pueden utilizarse de manera conjunta con Django para procesar, resumir y representar gráficamente grandes volúmenes de datos almacenados en una base de datos MySQL. Esta integración permite que el backend de la aplicación no solo administre la lógica del sistema, sino que también genere análisis dinámicos y visualizaciones útiles directamente desde el navegador, sin necesidad de recurrir a plataformas externas de business intelligence.

Asimismo, el uso de formularios personalizados, consultas ORM y plantillas HTML adaptables permitió construir una interfaz intuitiva para los usuarios, facilitando tanto la entrada como la consulta de los datos. En conjunto, estos resultados confirman que Django, al combinarse con el ecosistema de análisis de datos en Python, ofrece una base robusta para desarrollar sistemas de información a la medida, centrados en la toma de decisiones basada en datos.

Recomendaciones

Se recomienda la optimización de los formularios de captura de datos, con el fin de reducir vacíos de información y prevenir registros incompletos —tal como se evidenció en el análisis de los datos— se recomienda rediseñar los formularios internos de la empresa, incorporando validaciones obligatorias que aseguren el cumplimiento de los campos esenciales y garanticen la integridad mínima de cada registro.

A pesar de que los modelos actuales presentan un buen desempeño, se recomienda gestionar la incorporación de datos adicionales, ya sea a través de la empresa o mediante fuentes abiertas como la Federación Nacional de Cafeteros. Esto permitiría contar con un conjunto de datos lo suficientemente amplio para aplicar técnicas de validación cruzada y ampliar la evaluación de algoritmos. Con ello, se busca aumentar la robustez del análisis, reducir el riesgo de sobreajuste y minimizar posibles sesgos derivados del tamaño y composición de la muestra actual.

Referencias Bibliográficas

Adesina, A. A., Iyelolu, T. V., & Paul, P. O. (2024). Optimizing business processes with advanced analytics: Techniques for efficiency and productivity improvement. *World Journal of Advanced Research and Reviews, 22*(3), 1917–1926.

Aguilar Poma, K. L., & Zamudio Vargas, L. E. (2021). *La Cadena de Logística y su relación con las exportaciones de la SPN 0901.11.90.00-Los demás café sin tostar hacia Alemania durante el periodo 2015–2019*. [Tesis de licenciatura, Universidad Peruana de Ciencias Aplicadas]. https://repositorioacademico.upc.edu.pe/handle/10757/655681

Błaszkiewicz, J., Nowakowska-Bogdan, E., Barabosz, K., et al. (2023). Effect of green and roasted coffee storage conditions on selected characteristic quality parameters. *Scientific Reports, 13*(1), 6447. https://doi.org/10.1038/s41598-023-33755-9

Chavarro Oviedo, J. S., & Fuquen Acosta, K. A. (2022). *Integración de las nuevas tecnologías en el servicio de transporte terrestre de café en Colombia*.

Coronel, C., & Morris, S. (2018). *Database systems: Design, implementation, & management* (13th ed.). Cengage Learning.

Dittert, M., Härting, R. C., Reichstein, C., & Bayer, C. (2018). A data analytics framework for business in small and medium-sized organizations. En *Intelligent Decision Technologies 2017* (pp. 169–181). Springer. https://doi.org/10.1007/978-3-319-59427-9_15

Few, S. (2006). *Information dashboard design: The effective visual communication of data*. O'Reilly Media.

Geron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.

Gois, T. C., Thomé, K. M., & Balogh, J. M. (2023). Behind a cup of coffee: International market structure and competitiveness. *Competitiveness Review: An International Business Journal, 33*(5), 993–1009.

Holovaty, A., & Kaplan-Moss, J. (2009). *The definitive guide to Django: Web development done right* (2nd ed.). Apress.

Inayatulloh, I. (2023). Coffee distribution model with blockchain technology to increase the transparency of local coffee distribution. En *2023 International Conference on Information Management and Technology (ICIMTech)* (pp. 639–643). IEEE. https://doi.org/10.1109/ICIMTech59029.2023.10277770

Infante Aguilar, K. J., & Montoya Apaza, J. A. (2022). *Gestión logística y exportación de café verde de MYPES en Lima hacia el mercado canadiense, 2019–2020*. [Tesis de licenciatura, Universidad Peruana de Ciencias Aplicadas].

Jahin, M. A., Shovon, M. S. H., Shin, J., et al. (2024). Big data—Supply chain management framework for forecasting: Data preprocessing and machine learning techniques. *Archives of Computational Methods in Engineering, 31*, 3619–3645. https://doi.org/10.1007/s11831-023-09984-2

Joshi, A. P., & Patel, B. V. (2020). Data preprocessing: The techniques for preparing clean and quality data for data analytics process. *Oriental Journal of Computer Science and Technology, 13*(2–3), 78–81.

Kittichotsatsawat, Y., Jangkrajarng, V., & Tippayawong, K. Y. (2021). Enhancing coffee supply chain towards sustainable growth with big data and modern agricultural technologies. *Sustainability, 13*(8), 4593. https://doi.org/10.3390/su13084593

Knaflic, C. N. (2015). *Storytelling with data: A data visualization guide for business professionals*. Wiley.

Lapiedra Alcamí, R., Devece Carañana, C., & Guiral Herrando, J. (2011). *Introducción a la gestión de sistemas de información en la empresa*. Publicacions de la Universitat Jaume I.

Le Ngoc, T. N., et al. (2023). Machine learning for agricultural price prediction: A case of coffee commodity in Vietnam market. En *2023 IEEE/ACIS 8th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*. IEEE.

Le, N.-B.-v., Seo, Y.-S., & Huh, J.-H. (2024). Artificial intelligence in finance: Coffee commodity trading big data for informed decision making. *IEEE Access, 12*, 91780–91792. https://doi.org/10.1109/ACCESS.2024.3409762

Pizzaia, J. P. L., Salcides, I. R., Almeida, G. M. D., Contarato, R., & Almeida, R. D. (2018). Arabica coffee samples classification using a multilayer perceptron neural network. En *2018 13th IEEE International Conference on Industry Applications (INDUSCON)* (pp. 80–84). https://doi.org/10.1109/INDUSCON.2018.8627271

Provost, F., & Fawcett, T. (2013). *Data science for business: What you need to know about data mining and data-analytic thinking*. O'Reilly Media.

Quiñones-Ruiz, X. F. (2021). Social brokerage: Encounters between Colombian coffee producers and Austrian buyers – A research-based relational pathway. *Geoforum, 123*, 107–116. https://doi.org/10.1016/j.geoforum.2021.04.005

Rautenbach, S., de Kock, I. H., & Grobler, J. (2022). Data science for small and medium-sized enterprises: A structured literature review. *South African Journal of Industrial Engineering, 33*(3), 83–95. https://doi.org/10.7166/33-3-2801

Singgalen, Y. A. (2024). Design and implementation of coffeeshop management system prototype using rapid application development. *Journal of Information System Research (JOSH), 5*(3), 765–774.

Thai, H.-D., Ko, H.-J., & Huh, J.-H. (2024). Coffee bean defects automatic classification realtime application adopting deep learning. *IEEE Access, 12*, 126503–126517. https://doi.org/10.1109/ACCESS.2024.3419130

Tocto Tripul, M. F. (2019). *Diseño de un plan logístico para la optimización del proceso de exportación de café orgánico en grano Grupo PML Paita 2018*. [Tesis de licenciatura, Universidad César Vallejo]. https://repositorio.ucv.edu.pe/handle/20.500.12692/36931

Utrilla-Catalan, R., Rodríguez-Rivero, R., Narvaez, V., Díaz-Barcos, V., Blanco, M., & Galeano, J. (2022). Growing inequality in the coffee global value chain: A complex network assessment. *Sustainability, 14*(2), 672. https://doi.org/10.3390/su14020672

Valencia-Payan, C., Grass-Ramírez, J. F., Ramírez-González, G., & Corrales, J. C. (2022). A smart contract for coffee transport and storage with data validation. *IEEE Access, 10*, 37857–37869. https://doi.org/10.1109/ACCESS.2022.3165904

Vilcamiza, G., Trelles, N., Vinces, L., & Oliden, J. (2022). A coffee bean classifier system by roast quality using convolutional neural networks and computer vision implemented in an NVIDIA Jetson Nano. En *2022 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI)*. IEEE.

Vizueta Delgado, A. A., & Salinas Guerrero, J. J. (2021). *Costos logísticos y su incidencia en la exportación de café orgánico de la Amazonia ecuatoriana al mercado de Miami–Estados Unidos*. [Tesis de licenciatura, Universidad Laica Vicente Rocafuerte de Guayaquil]. http://repositorio.ulvr.edu.ec/handle/44000/4257