UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

(Universidad del Perú, DECANA DE AMÉRICA)

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMATICA

Escuela Profesional De Ingeniería De Sistemas

ASIGNATURA

Minería de datos

DOCENTE

Lezama Gonzales, Pedro Martín

INTEGRANTES

Morán Huamán, Dajanna Corina

Murillo Quispe, Pedro Alessandro

López Vargas Jhonny William

Huaroc Ricse Paul

Vargas Gonzales Jorge Antony

Lima-Perú

2024

ÍNDICE

1. Entendimiento empresarial

1.1. Determinar los objetivos comerciales

1.1.1. Recopilación de información sobre la situación comercial actual

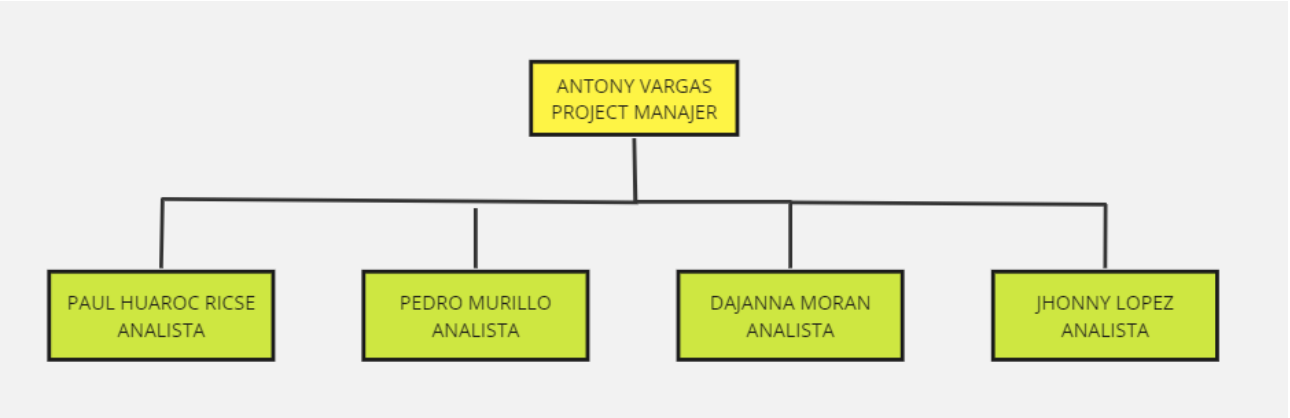
CASO DE NEGOCIO: VENTAS DE AMAZON (Minería web):

Amazon ha seguido siendo el mayor minorista en línea y uno de los principales jugadores en el comercio electrónico a nivel mundial.

Las ventas en línea de Amazon, que incluyen productos vendidos por terceros a través de su plataforma, han continuado creciendo año tras año impulsadas por la pandemia y la mayor adopción del e-commerce.

Amazon ha seguido expandiendo su red logística, centros de datos y presencia global para atender mejor la creciente demanda. Dado que Amazon es uno de los principales jugadores en el comercio electrónico, analizar sus datos de ventas podría brindarnos información valiosa sobre las tendencias del mercado, la competencia y las oportunidades para diferenciarse y ganar una ventaja competitiva, para cuando decidamos aplicarlo a una empresa que requiera nuestro servicio, con visión a brindar el servicio de minería de datos a empresas grande.

1.1.2. Estructura de la organización



1.1.3. Determinando el área problemática

Área problemática: El área de la problemática es el área de ventas

Descripción general de la problemática:

Amazon, siendo el gigante del comercio electrónico, genera una gran cantidad de datos valiosos sobre sus clientes, productos, transacciones y patrones de compra. Sin embargo, existe

una brecha significativa entre la recopilación de datos y la extracción de información útil para la toma de decisiones estratégicas en el área de ventas.

Esta problemática se manifiesta en la incapacidad para aprovechar todo el potencial de los datos, lo que limita las oportunidades de crecimiento y optimización del negocio, viéndose estos reflejados en el estancamiento de rentabilidad. Por ende, la minería de datos se presenta como una herramienta fundamental para abordar este desafío y convertir los datos en bruto en conocimientos accionables.

1.1.4. Motivaciones del proyecto

La motivación de este proyecto es aplicar conceptos teóricos de minería de datos a un caso real, como las ventas de Amazon, para desarrollar habilidades prácticas esenciales en ciencia de datos, mejorar la comprensión del negocio, y prepararse para el mercado laboral. El proyecto también busca experimentar con la gestión de proyectos, aportar conocimiento académico, y fomentar la innovación y creatividad en el análisis de datos, proporcionando una experiencia completa y relevante para los estudiantes.

1.1.5. Soluciones a la problemática

Utilizar BI & BA para sacar provecho de los datos generados por la empresa, respondiendo las siguientes preguntas: ¿qué está pasando?, ¿qué pasará? y ¿qué decisión debo tomar?, de esta manera se utilizará la metodología CRISP-DM, permitiendo así un entendimiento del negocio, generando información de los datos y transformando todo ello en conocimiento para que la empresa pueda tomar decisiones acertadas y mejore su rentabilidad.

En este sentido, como punto de partida para resolver la problemática se propone realizar los siguientes pasos:

- **Limpieza de datos:** Técnicas como la imputación de datos pueden estimar los valores faltantes en función de otros atributos. En este caso, el valor faltante de "Clase de artículo" podría completarse según la descripción del artículo.
- **Transformación de datos:** La normalización o estandarización de datos puede abordar las inconsistencias en el formato. Esto garantiza que todos los atributos estén en una escala similar para el análisis.
- **Integración de datos:** Si tiene un conjunto de datos más grande, puede integrar esta muestra para enriquecer el análisis.

Luego de aplicar minería se podría mejorar la comprensión del cliente y el marketing dirigido

En general se tiene a la minería de datos como solución a esta problemática.

1.1.6. Objetivos comerciales

- **Maximización de Ventas**: Uno de los objetivos más comunes podría ser aumentar el volumen de ventas en la plataforma de Amazon. Esto podría implicar estrategias para aumentar la visibilidad de los productos, mejorar las conversiones y expandir la cartera de productos ofrecidos.
- **Optimización de Precio**s: Un objetivo específico podría ser optimizar la estrategia de fijación de precios para maximizar los ingresos y la rentabilidad. Esto podría incluir la identificación de precios óptimos para diferentes productos y segmentos de clientes, así como la implementación de estrategias de descuento efectivas.
- **Mejora de la Experiencia del Cliente**: Otro objetivo importante podría ser mejorar la experiencia del cliente en la plataforma de Amazon. Esto podría incluir reducir los tiempos de entrega, mejorar la calidad del servicio al cliente y obtener una mejor calificación y comentarios de los clientes.

- **Expansión de Mercado**: La empresa podría tener como objetivo expandir su presencia en Amazon a nuevos mercados geográficos o a nuevas categorías de productos. Esto implicaría investigar y entender las demandas y preferencias de los clientes en esos mercados específicos.
- **Optimización de Inventarios**: Un objetivo operativo importante podría ser optimizar los niveles de inventario para minimizar los costos de almacenamiento y maximizar la disponibilidad de productos populares. Esto implicaría predecir la demanda futura con precisión y establecer políticas de reabastecimiento eficientes.
- **Incremento de Rentabilidad**: Otro objetivo clave podría ser aumentar la rentabilidad de las operaciones en Amazon. Esto podría lograrse reduciendo los costos operativos, mejorando los márgenes de beneficio y eliminando los productos con bajo rendimiento.
- **Fidelización de Clientes:** La empresa podría buscar fidelizar a los clientes existentes y aumentar la lealtad a la marca en la plataforma de Amazon. Esto podría lograrse a través de programas de recompensas, ofertas exclusivas para clientes frecuentes y una comunicación proactiva con los clientes

1.2. Evaluar la situación

1.2.1. Inventario de recursos

a. Recursos de hardware

Para poder realizar las tareas de minería de datos en una PC, se requiere que las PC puedan cumplir con los siguientes requerimientos:

- *Procesador (CPU):*

  Procesador de doble núcleo o superior.

  Frecuencia de reloj de al menos 2.0 GHz.

- *Memoria RAM:*

Mínimo 4 GB de RAM. Se recomienda 8 GB o más para un rendimiento óptimo, especialmente al manejar conjuntos de datos más grandes.

- *Almacenamiento:*

Espacio de almacenamiento suficiente para el sistema operativo, Python y cualquier otro software necesario, así como también para los archivos de datos. En general, al menos 128 GB de almacenamiento en disco duro deberían ser adecuados.

- *Sistema operativo:*

Python es compatible con una variedad de sistemas operativos, incluyendo Windows, macOS y Linux.

b. Identificación de orígenes de datos y almacenes de conocimientos

Los datos se almacenan en un archivo de Excel, lo cual significa que necesitaremos una biblioteca de Python capaz de leer y procesar archivos de Excel, como Pandas. Dado que estamos trabajando con un archivo local, no necesitaremos acceso a bases de datos operativas o almacenes de datos remotos, sin embargo, es crucial asegurarse de que el archivo de Excel esté accesible desde la ubicación donde se ejecutará el código de Python.

No se prevé la adquisición de datos externos adicionales en este momento, por lo que no será necesario gestionar la integración de fuentes de datos externas.

En cuanto a la seguridad, al tratarse de un archivo local, no debería plantear problemas significativos, ya que los datos se almacenan en la PC utilizada para el proyecto y se pueden proteger mediante medidas estándar de seguridad informática.

c. Recursos personales

Para este punto, es importante tener acceso a empresas y expertos en datos que puedan proporcionar orientación y apoyo en caso de necesidad. Esto podría incluir participación en comunidades en línea, foros de discusión o grupos de usuarios de Python y minería de datos. Además, es útil tener identificados administradores de bases de datos y otro personal de apoyo que puedan brindar asistencia técnica en caso de que surjan problemas con la infraestructura de datos. Estas personas pueden ser contactadas y consultadas según sea necesario durante el desarrollo del proyecto.

Lista de Contactos y Recursos en línea:

- ❖ Comunidad en línea de Python y Ciencia de Datos:
- Sitio web: https://www.python.org/community/
- Foros de discusión: https://discuss.python.org/
- ❖ Foro de Minería de Datos de Stack Overflow:
- Sitio web: https://stackoverflow.com/questions/tagged/data-mining

1.2.2. Requisitos, supuestos y restricciones

a. Determinar requisitos

- Restricciones Legales y de Seguridad:

Decreto Supremo N° 003-2013-JUS:

Detalla las obligaciones específicas para los responsables del tratamiento de datos, incluyendo medidas de seguridad, notificaciones de incidencias de seguridad y procedimientos para el ejercicio de los derechos de los titulares de datos. Además, para el cumplimiento de la Ley N° 29733 se deberá registrar la BD de la empresa en la Autoridad Nacional de Protección de Datos Personales (ANPD), donde se supervisa y garantiza el cumplimiento de los reglamentos ante el tratado de datos personales.

El GDPR (Reglamento General de Protección de Datos):

Proteger los datos personales y la privacidad de los ciudadanos de la UE y regular cómo las organizaciones deben gestionar estos datos.

CCPA (California Consumer Privacy Act):

Proteger la privacidad de los consumidores de California y otorgarles más control sobre cómo se recopilan y utilizan sus datos personales.

- Alineación de Usuarios:

Es necesario asegurar que todos los usuarios involucrados en el proyecto estén alineados con los objetivos y requisitos de planificación. Esto garantizará un apoyo adecuado y una ejecución efectiva del proyecto. En ese sentido, se desarrollan reuniones con los actores de las distintas áreas de la empresa, facilitando el entendimiento del negocio y la alineación del proyecto con los objetivos de la empresa.

- Requisitos de Despliegue de Resultados:

Para los resultados obtenidos, se utilizarán herramientas de BI como Tableau o Power BI, u otros de ser necesarios, lo que requieren los directivos es la visualización de los resultados y/o hallazgos obtenidos, para ello la generación de dashboards será necesaria.

b. Describir los supuestos

Supuestos de Calidad de Datos:

Evaluar la fuente y calidad de datos: Identificar las razones específicas por las cuales los datos no están disponibles o son difíciles de obtener. Esto puede incluir problemas con el acceso a bases de datos, restricciones de privacidad, problemas técnicos, etc. Si este es el caso, la comunicación con los proveedores sería primordial para eliminar las barreras de acceso.

Por otro lado, si dichos datos no tienen buena consistencia, se podrá evaluar otra fuente de datos que nos permitan alcanzar los objetivos del proyecto, otra opción será focalizarse en

obtener los datos más críticos que son esenciales para el análisis, y postergar o simplificar la obtención de datos menos cruciales.

Expectativas del Patrocinador:

- Se espera que el patrocinador y el equipo de dirección no solo visualicen los resultados, sino que también comprendan el modelo subyacente y las razones detrás de los hallazgos.
- Se asume que los resultados del proyecto serán presentados de manera clara y comprensible, utilizando visualizaciones y explicaciones detalladas.

1.2.3. Riesgos y contingencias

Para este punto se realizó el análisis de los activos, amenazas y riesgos del proyecto. Posteriormente, se propuso las diferentes salvaguardas o medidas de acción procurando la plenitud y desarrollo ininterrumpido del proyecto, así como el cumplimiento de los objetivos comerciales.

En ese sentido, a continuación se muestran las distintas matrices trabajadas con el software Excel.

A. Matriz de Riesgos

NATURALES	**Desastre natural localizado**	Fenómenos naturales como terremoto o inundación, pueden afectar directamente el lugar donde se encuentra el equipo de desarrollo del proyecto, dañando la infraestructura, los equipos y los datos.
	Pandemias/Epidemias	En caso de una pandemia o epidemia, como el COVID-19, se pueden imponer cuarentenas, confinamientos y restricciones de movilidad que impidan al equipo de desarrollo trabajar de manera presencial.
PERSONAS	**Deserción de personal**	Empleados abandonan voluntariamente sus puestos de trabajo o la organización en la que están empleados

	Salud y seguridad	Accidentes laborales, enfermedades o cualquier otro factor que vulnera la salud del personal, así como la necesidad de cuidar a un familiar cercano, impidiendo el desarrollo pleno de su cargo dentro del proyecto
TECNOLOGÍA	**Fallas en la arquitectura de hardware**	Posibilidad de que ocurra alguna falla en los equipos de la empresa y/o equipo de trabajo, como computadoras, bases de datos, enrutador, entre otros
	Ataques malintencionados	Hackers o malware podrían comprometer la seguridad del sistema, acceder a datos confidenciales o interrumpir su funcionamiento normal.
FINANCIERO	**Costos ocultos**	Pueden surgir costos adicionales que no se tuvieron en cuenta inicialmente, como la adquisición de licencias de software, la integración con otros sistemas existentes, la contratación de personal adicional o la capacitación continua.
	Mantenimiento y actualizaciones	Los costos recurrentes de mantenimiento, seguridad y actualizaciones del sistema podrían ser más altos de lo anticipado inicialmente.
SOCIAL	**Resistencia al cambio**	Si la toma de decisiones involucra realizar cambios en los procesos, como la automatización, los usuarios involucrados podrían estar en contra lo que dificultará la implementación y mejora de la empresa.
	Falta de participación y retroalimentación de los usuarios	La falta de involucramiento de los usuarios en el proceso de desarrollo del proyecto puede llevar a la insatisfacción y falta de confianza en el equipo. Si los actores del negocio no están involucrados con el proyecto, podrían existir dificultades para alinear los objetivos de la empresa con los objetivos del proyecto.

B. Salvaguardas

N°	Salvaguardas según Riesgo
R-1	Establecer un plan de continuidad del negocio y proyecto que incluya la ubicación de un sitio de respaldo, realizar copias de seguridad, implementar medidas de protección física, como sistemas contra incendios, y contar con un seguro que cubra los daños ocasionados por desastres naturales.
R-2	Establecer políticas y procedimientos para el trabajo remoto, asegurando que todos los miembros del equipo tengan acceso a las herramientas y recursos necesarios para continuar su trabajo desde casa.

R-3	- Mejorar la comunicación interna - Mejorar las condiciones de trabajo - Realizar encuestas de satisfacción laboral
R-4	Implementar políticas de salud y seguridad en el trabajo. Ofrecer flexibilidad laboral y opciones de trabajo remoto
R-5	- Tener en el inventario hardware de reemplazo - Contar con personal de soporte
R-6	Revisar las políticas de ciberseguridad de la empresa. Si no hubiera, sugerir la auditoría como medida ante el tratamiento de datos. Utilizar software antivirus y antimalware actualizado.
R-7	Proporcionar informes detallados sobre las predicciones y evaluaciones de datos, incluyendo estimaciones de costos potenciales si la toma de decisiones incluye la implementación de algún SI.
R-8	Incluir recomendaciones sobre las necesidades de mantenimiento y actualización en los informes de evaluación de datos. Recomendar la adopción de soluciones escalables y modulares que faciliten futuras actualizaciones.
R-9	Comunicar claramente los beneficios del nuevo sistema, si es que hubiera, identificar y abordar las preocupaciones de los empleados, proporcionar capacitación adecuada y ofrecer apoyo continuo durante la transición.
R-10	Establecer canales de comunicación efectivos para que los usuarios puedan expresar sus comentarios, sugerencias y preocupaciones sobre el proyecto. Realizar encuestas periódicas, grupos de enfoque o reuniones abiertas para recopilar información y retroalimentación valiosa de los usuarios.

1.2.4. Terminología

En este punto, se identifican los términos técnicos que se extraen del conjunto de datos y términos usados en la minería de datos para incluirlos en el glosario de terminología.

- *Descuento (Discount Amount):*

Definición: El monto de reducción aplicado al precio de un artículo durante una transacción comercial.

- *Factura (Invoice):*

Definición: Un documento que detalla los bienes vendidos o los servicios prestados, junto con el precio y los términos de pago.

- *Precio de Lista (List Price):*

Definición: El precio sugerido por el fabricante o vendedor para un artículo determinado.

- *Margen de Ventas (Sales Margin Amount):*

Definición: La diferencia entre el precio de venta de un artículo y su costo de ventas, que representa la ganancia bruta generada por la transacción.

- *Cantidad Vendida (Sales Quantity):*

Definición: La cantidad de unidades de un artículo vendidas durante una transacción específica.

- *Representante de Ventas (Sales Rep):*

Definición: Una persona responsable de realizar ventas y mantener relaciones con los clientes en nombre de la empresa.

- *Unidad de Medida (U/M):*

Definición: La unidad utilizada para medir la cantidad de un artículo vendido, como "EA" para unidades individuales, "PR" para pares o “SE” para conjunto de unidades.

- *Regresión:*

Definición: Un método estadístico utilizado para modelar la relación entre una variable dependiente y una o más variables independientes.

- *Clasificación:*

Definición: Un problema de aprendizaje supervisado en el que se asigna una etiqueta a una instancia de entrada de un conjunto finito de categorías.

- *Validación Cruzada (Cross-Validation):*

Definición: Una técnica utilizada para evaluar el rendimiento de un modelo de Machine Learning mediante la división de los datos en conjuntos de entrenamiento y prueba múltiples.

- *Sobreajuste (Overfitting):*

Definición: Un fenómeno en el que un modelo de Machine Learning se ajusta demasiado a los datos de entrenamiento y no generaliza bien a nuevos datos.

- *Submuestreo (Undersampling) y Sobremuestreo (Oversampling):*

Definición: Técnicas utilizadas para abordar el desequilibrio de clases en conjuntos de datos donde una clase es mucho más predominante que otras.

1.2.5. Análisis de costes

Matriz de costos operativos

Concepto	Costo Mensual (S/)	Costo Anual (S/)
Alquiler de Área de Oficina	500	6000
Salarios del equipo de minería de datos y científicos de datos	2000	24000
Computadoras	1500	0
Servicios de Internet, Agua y Electricidad	200	2400
Otros gastos operativos (suministros de oficina, etc.)	100	1200
Total	**4300**	**33600**

1.3. Objetivos de la minería de datos

Objetivos de minería de datos:

- **Agrupamiento:** Segmentar a los clientes en grupos o clusters basados en patrones de compra, como cantidades, descuentos, márgenes de ventas, precios de lista, etc., para enviar ofertas y recomendaciones personalizadas.
- **Análisis de la cesta de compra**: Generar un modelo que identifique productos (Item Number) relacionados o complementarios para recomendar a los clientes en función de su historial de compras y las combinaciones de productos en una misma factura (Invoice Number).

Objetivos técnicos:

- Agrupar al 90% de los clientes en segmentos significativos de patrones de compra dentro de un margen de error del 5%.
- Incrementar las ventas cruzadas en un 20% mediante recomendaciones de productos relacionados en el análisis de la cesta de compra.

Criterios de rendimiento:

- Modelos de predicción: Utilizar métricas como precisión, exhaustividad, puntuación F1, área bajo la curva ROC para evaluar el rendimiento.
- Modelos de agrupamiento: Utilizar métricas como la silueta, el índice de Davies-Bouldin, la puntuación de Calinski-Harabasz para evaluar la calidad de los clusters.
- Análisis de la cesta de compra: Evaluar el aumento en las ventas cruzadas, la tasa de conversión y los ingresos generados por las recomendaciones.
- Definir umbrales mínimos aceptables para cada una de estas métricas como puntos de referencia (por ejemplo, un F1-score mínimo de 0.8 para los modelos de predicción).
- Planificar el despliegue de los modelos en producción y monitorear su rendimiento en tiempo real.

1.4. Producir el plan del proyecto

1.4.1. Plan de proyecto

Fase	Hora	Recursos	Riesgos
Entendimiento del negocio	1 semana	Todos los analistas	Falta de información clara de los objetivos
Comprensión de los datos	1 semana	Todos los analistas	Datos incompletos o de baja calidad
Preparación de los datos	2 semanas	Científicos de Datos	Errores en la limpieza de datos
Modelado	2 semanas	Científicos de Datos	Selección incorrecta de técnicas y algoritmos
Evaluación	1 semana	Todos los analistas	Resultados no alineados con objetivos comerciales
Monitoreo y mantenimiento	Continuo	Ingenieros de Datos	Falta de actualizaciones y ajustes continuos Desempeño decreciente de los modelos con el tiempo

1.4.2. Evaluación de herramientas y técnicas

1.4.2.1. Herramientas:

Herramienta	Ventajas	Desventajas
Python con pandas, scikit-learn, y otras bibliotecas de ML	- Ampliamente utilizado y con una gran comunidad de soporte. Ofrece una amplia gama de bibliotecas y herramientas para tareas de minería de datos y aprendizaje automático. Flexible y altamente personalizable. Integración con otras bibliotecas y herramientas de análisis de datos.	- Puede requerir conocimientos de programación. La curva de aprendizaje puede ser empinada para principiantes. Algunas tareas pueden requerir más código en comparación con herramientas con interfaces gráficas.
R con bibliotecas como caret, tidyverse, etc.	- Ampliamente utilizado en la comunidad académica y de investigación. Ofrece una amplia gama de paquetes para análisis de datos y modelado estadístico. Buena integración con gráficos y visualización de datos. Acceso a algoritmos y técnicas de vanguardia. Buena documentación y tutoriales disponibles.	- Puede tener una curva de aprendizaje pronunciada para usuarios nuevos en la programación. Requiere conocimientos de programación en R. Menos flexibilidad que Python en algunas áreas fuera del análisis de datos. La gestión de datos puede ser menos eficiente en comparación con herramientas basadas en bases de datos.
Weka	- Interfaz gráfica fácil de usar para el análisis de datos y la minería de datos. Amplia gama de algoritmos y técnicas de aprendizaje automático preimplementadas. Buenas capacidades de visualización y evaluación de modelos. Ideal para principiantes en minería de datos. Permite una fácil comparación de algoritmos y técnicas.	- Puede ser menos eficiente para conjuntos de datos muy grandes o complejos. La flexibilidad y la capacidad de personalización pueden ser limitadas en comparación con herramientas de programación. Menos utilizado en entornos comerciales en comparación con herramientas como Python o R.
RapidMiner	- Interfaz gráfica fácil de usar que no requiere codificación.	- La versión gratuita tiene algunas limitaciones en

	Amplia gama de algoritmos y técnicas de minería de datos preimplementadas. Buena integración con bases de datos y otras fuentes de datos. Ofrece capacidades de automatización y programación para usuarios avanzados. Visualizaciones interactivas y paneles de control para análisis de resultados.	comparación con la versión de pago. La curva de aprendizaje puede ser pronunciada para usuarios nuevos en la minería de datos. Menos flexibilidad y capacidad de personalización en comparación con herramientas de programación.

1.4.2.2. Técnicas :

Técnica	Ventajas	Desventajas	Aplicaciones	Ejemplos de Uso
Aprendizaje Profundo	Capacidad para aprender representaciones de datos complejas y de alto nivel.	Requiere grandes cantidades de datos etiquetados para entrenar con eficacia.	Reconocimiento de imágenes y voz, traducción automática, procesamiento de texto, recomendación de contenido, entre otros.	Reconocimiento facial, reconocimiento de voz, conducción autónoma, análisis de sentimientos en redes sociales.
Procesamiento de Lenguaje Natural (NLP)	Permite la comprensión y generación de lenguaje humano, lo que es crucial en aplicaciones como la atención al cliente o el análisis de sentimientos.	Puede ser difícil trabajar con datos no estructurados y ambiguos.	Análisis de sentimientos, traducción automática, generación de texto, extracción de información, atención al cliente, entre otros.	Asistentes virtuales como Siri, Alexa, análisis de opiniones en redes sociales, traducción automática de Google Translate.
Detección de Anomalías	Identificación eficaz de eventos inusuales o potencialmente peligrosos en grandes conjuntos de datos.	Requiere una comprensión profunda del dominio y de lo que constituye una anomalía en el contexto	Detección de fraudes en transacciones financieras, seguridad informática (detección de	Detección de fraudes en tarjetas de crédito, detección de intrusiones en redes

		específico.	intrusiones), mantenimiento predictivo, monitorización de equipos industriales, entre otros.	informáticas, monitoreo de equipos médicos para detectar comportamientos inusuales.
Recomendación	Personalización de recomendaciones basadas en el comportamiento del usuario.	Dependencia de datos históricos de calidad y de la disponibilidad de información sobre preferencias.	Recomendación de productos en comercio electrónico, recomendación de películas y música, sistemas de recomendación de noticias, entre otros.	Recomendaciones de películas en Netflix, recomendaciones de productos en Amazon, recomendaciones de amigos en redes sociales.
Segmentación Avanzada	Permite una comprensión más profunda de la estructura de los datos y la identificación de patrones sutiles.	Puede ser difícil interpretar y validar los resultados del clustering.	Segmentación de clientes para marketing personalizado, segmentación de mercados, análisis de redes sociales, segmentación de pacientes para atención médica, entre otros.	Segmentación de clientes en grupos de alto valor, segmentación de mercados para campañas publicitarias específicas, segmentación de pacientes para ofrecer tratamientos personalizados.
Optimización	Encuentra soluciones óptimas a problemas complejos con múltiples objetivos y restricciones.	Puede requerir una cantidad significativa de tiempo y recursos computacionales.	Planificación de rutas de transporte, asignación de recursos en la cadena de suministro, programación	Optimización de rutas para empresas de logística, asignación de personal en

			de horarios, diseño de redes de comunicación, entre otros.	hospitales, programación de horarios en universidades, diseño de redes de telecomunicaciones.

1.4.3. Preguntas Críticas

1.4.3.1. Preguntas Críticas Desde una Perspectiva Comercial

- ¿Qué espera obtener de este proyecto?

Objetivos: El proyecto tiene como objetivo maximizar las ventas en Amazon, optimizar la fijación de precios, mejorar la experiencia del cliente, expandir el mercado, optimizar los inventarios, incrementar la rentabilidad y fidelizar a los clientes. Estos objetivos se alinean con la necesidad de aprovechar los datos para tomar decisiones estratégicas informadas y mejorar la competitividad en el mercado minorista.

- ¿Cómo define la finalización de los trabajos?

Definición de Finalización: La finalización del proyecto se define por la implementación exitosa de las técnicas de minería de datos que permiten alcanzar los objetivos comerciales establecidos, la generación de informes con insights accionables, la implementación de mejoras en las áreas de ventas e inventarios, y la constatación de mejoras en las métricas clave (como el aumento en las ventas y la optimización de precios).

- ¿Dispone de la dotación presupuestaria y de los recursos necesarios para completar los objetivos?

Recursos y Presupuesto: se dispone de los alumnos, en este caso analistas como recurso humano, además del profesor como asesor; así como recursos tecnológicos como herramientas de software para minería de datos python, tableau, entre otros, también la información del dataset

- ¿Dispone de acceso a todos los datos necesarios para el proyecto?

Acceso a Datos: El acceso a todos los datos relevantes de ventas de Amazon, incluyendo datos históricos de transacciones no es posible, ya que eso es parte de la información privada de la empresa. Sólo se tiene acceso a los datos del dataset de análisis.

- ¿Ha tratado con su equipo los riesgos y contingencias asociadas con el proyecto?

Gestión de Riesgos: Sí, se han identificado y discutido con el equipo todos los posibles riesgos asociados con el proyecto, incluyendo la calidad y seguridad de los datos, la precisión de los modelos, y las contingencias en caso de fallos. Se han desarrollado planes de mitigación para estos riesgos.

- ¿Los resultados del análisis de costes/beneficios hacen que el proyecto sea viable?

Viabilidad del Proyecto: Se ha realizado un análisis de costes/beneficios detallado que muestra que los beneficios potenciales del proyecto (como el aumento en ventas, la optimización de inventarios y la mejora de la experiencia del cliente) superan significativamente los costos asociados, haciendo que el proyecto sea viable.

1.4.3.2. Preguntas Críticas Desde una Perspectiva de Minería de Datos

- ¿En qué forma puede ayudarle la minería de datos a cumplir sus objetivos comerciales?

Contribución de la Minería de Datos: La minería de datos puede ayudar a identificar patrones y tendencias en los datos de ventas, optimizar la fijación de precios, segmentar clientes para campañas de marketing personalizadas, predecir la demanda futura para optimizar inventarios, y descubrir oportunidades para mejorar la experiencia del cliente.

- ¿Sabe qué técnicas de minería de datos producen los mejores resultados?

Técnicas Efectivas: Se han identificado y probado varias técnicas de minería de datos que son adecuadas para los objetivos del proyecto, regresión para predicción de ventas, clasificación, y algoritmos de machine learning para recomendaciones de productos.

- ¿Cómo puede saber que sus resultados son precisos o efectivos?

Medición de la Precisión: Se han definido métricas de rendimiento específicas para evaluar la precisión y efectividad de los resultados de la minería de datos, como la precisión del modelo (accuracy), el error cuadrático medio (RMSE), y el retorno sobre la inversión (ROI)

de las estrategias implementadas. Se realizarán pruebas y validaciones continuas para asegurar la calidad de los resultados.

- ¿Cómo se desplegarán los resultados de modelado? ¿Ha considerado el despliegue en su plan de proyecto?

Despliegue de Resultados: Los resultados de modelado se desplegarán mediante dashboards interactivos, informes detallados, y sistemas de recomendación implementados en la plataforma de ventas. El plan de proyecto incluye una fase dedicada al despliegue y prueba de estos resultados en el entorno de producción.

- ¿El plan de proyecto incluye todas las fases de CRISP-DM?

Inclusión de CRISP-DM: Sí, el plan de proyecto sigue el modelo CRISP-DM (Cross-Industry Standard Process for Data Mining), que incluye todas las fases: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue.

2. Comprensión de los datos

2.1. Recopilación de datos iniciales

En esta primera etapa, se hará la importación de las librerías y la recopilación de los datos.

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

datos = pd.read_excel("SALESDATA.xlsx")

# Mostrar los primeros registros
print(datos.head())
```

Luego de ya tener las librerías importadas, se realiza la lectura de los datos y después se procede a llamar a las primeras filas para verificar que se haya leído correctamente el archivo.

2.2. Descripción de los datos

En esta siguiente etapa, procedemos a obtener una mejor comprensión de la estructura, contenido y estadísticas descriptivas del conjunto de datos.

```
# ¿Qué dimensiones tiene la base de datos (en números de filas y columnas)?
datos.shape

# Nombres de las columnas
datos.columns

# Tipos de datos de cada columna
datos.dtypes

datos.describe(include='number')

datos.info()

print("El tamaño del archivo de datos es: ", datos.shape, "\n")
print("El número de entradas en 65280 filas para cada característica es:\n")
datos.count()

datos.groupby('Custkey')['Sales Price'].describe()
```

Se hace el llamado a las distintas funciones para poder observar las dimensiones (filas y columnas), el nombre de las columnas, tipos de datos, estadísticas, cantidad de entradas no nulas para las columnas, etc.

2.3. Exploración de datos

Esta tercera etapa tiene como fin realizar una exploración inicial y un análisis preliminar de los datos para comprender mejor sus características, patrones y peculiaridades, para ello se realiza una variedad de funciones que serán mencionadas posteriormente.

a. Calcular la cantidad de artículos por categoría:

```
print("El número de artículos en cada categoría es:\n")
pd.value_counts(datos['Item'])
```

```
El número de artículos en cada categoría es:

Item
Better Fancy Canned Sardines    1648
Ebony Prepared Salad            1471
Moms Sliced Turkey              1192
Imagine Popsicles               1191
Discover Manicotti              1126
                                ...
BBB Best Corn Oil                  1
Choice Bubble Gum                  1
Atomic White Chocolate Bar         1
Tell Tale Potatos                  1
Discover Rice Medly                1
Name: count, Length: 657, dtype: int64
```

La mayor parte de los datos está constituida por esas 5 predominantes.

b. Graficar la cantidad de artículos por categoría Top 10:

```
# Calcular las frecuencias de cada categoría
frecuencias = datos['Item'].value_counts()
# Seleccionar las 10 más comunes
top_10 = frecuencias.head(10)
# Filtrar el DataFrame original para incluir solo las 10 categorías más comunes
datos_top_10 = datos[datos['Item'].isin(top_10.index)]

# Crear el gráfico utilizando el DataFrame filtrado
ax = sns.displot(data=datos_top_10, y='Item', aspect=3)
ax.set_axis_labels('Product Count', 'Prod Category')
```

Se crea una gráfica de barras con el fin de identificar cuáles son los 10 artículos que tienen más relevancia. En donde se podrá observar que la predominante es la de Better Fancy Canned Sardines.

c. Gráfico de dispersión de 'Sales Quantity' vs 'Sales Amount'

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Sales Quantity', y='Sales Amount', data=datos)
```

```
plt.title('Sales Quantity vs Sales Amount')
plt.xlabel('Sales Quantity')
plt.ylabel('Sales Amount')
plt.show()
```

El gráfico que se llega a crear con el código se hace con el fin de analizar si es que existe o no alguna tendencia entre los puntos.

d. Identificar y graficar la frecuencia de las clases de ítem

```
frecuencia_item_class = datos['Item Class'].value_counts()
print(frecuencia_item_class)

# Crear un gráfico de barras
frecuencia_item_class.plot(kind='bar', figsize=(10, 6), color='skyblue')

# Añadir títulos y etiquetas
plt.title('Número de artículos en cada clase de ítem')
plt.xlabel('Clase de ítem')
plt.ylabel('Número de artículos')

# Mostrar el gráfico
plt.show()
```

e. Gráfica entre precios de venta y precios de lista

```
sns.scatterplot(x='List Price', y='Sales Price', data=datos)
```

Esto es con el fin de verificar si es que existe o no alguna relación entre esos dos y verificar si se encuentra alguna variación.

f. Gráfico para ventas por representante de ventas

```
# Gráfico de barras para ventas por representante de ventas:
datos.groupby('Sales Rep')['Order Number'].sum().plot.bar()
```

```
# Gráfico de barras para ventas por representante de ventas (TOP 5):

# Contar la cantidad de órdenes distintas por representante
ventas_por_rep = datos.groupby('Sales Rep')['Order Number'].nunique()
# Ordenar de manera descendente las ventas por representante
ventas_por_rep = ventas_por_rep.sort_values(ascending=False)

# Obtener los top 5 representantes con mayores ventas
top_5_reps = ventas_por_rep.head(5)

# Crear el gráfico de barras con valores reales en el eje y
fig, ax = plt.subplots(figsize=(10, 6))
top_5_reps.plot.bar(ax=ax, rot=0)
ax.set_xlabel('Cantidad de Órdenes', fontsize=12)
ax.set_ylabel('Ventas Totales', fontsize=12)
ax.set_title('Top 5 Representantes de Ventas por Cantidad de Órdenes', fontsize=14)
plt.show()
```

```
# Gráfico de barras para Top 5 Representantes de Ventas con menor cantidad de
# órdenes de venta:

# Contar la cantidad de órdenes distintas por representante
ventas_por_rep = datos.groupby('Sales Rep')['Order Number'].nunique()
# Ordenar de manera descendente las ventas por representante
ventas_por_rep = ventas_por_rep.sort_values(ascending=True)

# Obtener los top 5 representantes con menores ventas
top_5_reps = ventas_por_rep.head(5)

# Crear el gráfico de barras con valores reales en el eje y
fig, ax = plt.subplots(figsize=(10, 6))
top_5_reps.plot.bar(ax=ax, rot=0)
ax.set_xlabel('Cantidad de Órdenes', fontsize=12)
ax.set_ylabel('Ventas Totales', fontsize=12)
```

```
ax.set_title('Top 5 Representantes de Ventas con menor cantidad de órdenes de venta',
fontsize=14)
plt.show()
```

Se realizan esos tres gráficos con el fin de identificar en primer lugar una vista más grande sobre la variación que hay entre la cantidad de ventas por representante, luego tenemos un Top 5 representante con mayor cantidad de ventas y por último el Top 5 con menor cantidad de ventas.

g. Gráfico para ventas por fecha

```
# Gráfico de líneas para ventas por fecha:
datos.groupby('Invoice Date')['Sales Amount'].sum().plot()
```

```
# Gráfico de líneas para ventas por fecha (TOP 5)

# Agrupar por 'Invoice Date' y sumar las 'Sales Amount'
ventas_por_fecha = datos.groupby('Invoice Date')['Sales Amount'].sum()

# Ordenar de manera descendente las ventas por fecha
ventas_por_fecha = ventas_por_fecha.sort_values(ascending=False)

# Obtener las top 5 fechas con mayores ventas
top_5_fechas = ventas_por_fecha.head(5)

# Crear el gráfico de líneas
fig, ax = plt.subplots(figsize=(12, 6))
top_5_fechas.plot(ax=ax, marker='o', linestyle='-')
ax.set_xlabel('Fecha de Factura', fontsize=12)
ax.set_ylabel('Ventas Totales', fontsize=12)
ax.set_title('Top 5 Fechas con Mayores Ventas Totales', fontsize=14)
plt.xticks(rotation=45)
plt.show()
```

De igual manera que en el caso anterior, se realiza con el fin de que se pueda verificar en un principio las fechas que tuvieron más ventas, luego se realiza otro gráfico para verificar cuáles serían las 5 fechas con mayores ventas.

2.4. Verificación de calidad de datos

En esta última etapa se realizaron algunas comprobaciones para evaluar la calidad de los datos del conjunto de datos.

a. Cantidad de valores únicos por columna

```
print("\nCantidad de valores únicos en cada columna:")
datos.nunique()
```

```
Cantidad de valores únicos en cada columna:
Custkey                          615
DateKey                          559
Discount Amount                17818
Invoice Date                     559
Invoice Number                 24679
Item Class                        16
Item Number                      983
Item                             657
Line Number                      397
List Price                      1062
Order Number                   17796
Promised Delivery Date           592
Sales Amount                   17895
Sales Amount Based on List Price  4060
Sales Cost Amount                5513
Sales Margin Amount             21295
Sales Price                     14788
Sales Quantity                    281
Sales Rep                          64
U/M                                 4
dtype: int64
```

b. Cantidad de valores nulos por columna

```
print("Los valores nulos de cada columna son: \n")
datos.isnull().sum()
```

```
Los valores nulos de cada columna son:

Custkey                              0
DateKey                              0
Discount Amount                      2
Invoice Date                         0
Invoice Number                       0
Item Class                        8285
Item Number                         40
Item                                 0
Line Number                          0
List Price                           0
Order Number                         0
Promised Delivery Date               0
Sales Amount                         0
Sales Amount Based on List Price     0
Sales Cost Amount                    0
Sales Margin Amount                  0
Sales Price                          1
Sales Quantity                       0
Sales Rep                            0
U/M                                  0
```

c. Cantidad de valores duplicados por columna:

```
print("Los valores duplicados de cada columna son:")
datos.duplicated().sum()
```

```
Los valores duplicados de cada columna son:
0
```

3. Preparación de datos

3.1. Selección de datos

La selección de datos es una parte fundamental en la preparación de datos para análisis o modelado. Puedes seleccionar datos de diversas formas, como por columnas específicas, filas que cumplan ciertas condiciones o combinaciones de ambas.

```python
# ------------------------------------------------------------
# Selección de una muestra o subconjunto
# ------------------------------------------------------------
# Queremos una muestra del 20% de los datos
muestra = datos.sample(frac=0.2, random_state=42)

print("\nMuestra del 20% de los datos:")
print(muestra)
```

```
Muestra del 20% de los datos:
        Custkey    DateKey  Discount Amount Invoice Date  Invoice Number
6752   10023793 2019-10-13         224.6600   2019-10-13          314600
43892  10027370 2017-09-19        1471.9000   2017-09-19          113466
59143  10012715 2017-03-18        8266.2100   2017-03-18          125396
31420  10019066 2018-02-14         459.3400   2018-02-14          223464
39975  10023538 2017-11-04         393.4000   2017-11-04          117572
...         ...        ...              ...          ...             ...
48227  10012715 2017-07-31         958.7402   2017-07-31          109038
37225  10006037 2017-12-01         347.0100   2017-12-01          127860
7667   10023793 2019-10-05         189.3200   2019-10-05          313743
8578   10023757 2019-09-27         332.1000   2019-09-27          313015
30968  10007564 2018-02-19        1078.0000   2018-02-19          223820
```

```python
# ------------------------------------------------------------
# Agregación de registros
# ------------------------------------------------------------
# Queremos obtener las ventas totales por ItemNumber
ventas_por_item = datos.groupby('Item Number')['Sales Amount'].sum().reset_index()

print("\nVentas totales por ItemNumber:")
print(ventas_por_item)
```

```
Ventas totales por ItemNumber:
            Item Number  Sales Amount
0                     0       3383.00
1                  2040      23553.61
2                  5320        995.03
3                  5742     299872.50
4                  6341       1854.77
..                  ...           ...
978  SUPER WINCH T-1500      24820.00
979                 TLR       3003.41
980         TLR - OCEAN       6416.16
981             TLR/BTS       2590.70
982             TLR/SEA       3746.47
```

```
# ------------------------------------------------------------
# Derivación de nuevos atributos
# ------------------------------------------------------------
# Crear un nuevo atributo 'Margen_Porcentaje' calculado a partir de otros campos
datos['Margen_Porcentaje'] = (datos['Sales Margin Amount'] / datos['Sales Amount']) * 100

print("\nDatos con nuevo atributo 'Margen_Porcentaje':")
print(datos[['Item Number', 'Sales Amount', 'Sales Margin Amount', 'Margen_Porcentaje']].head())
```

```
Datos con nuevo atributo 'Margen_Porcentaje':
   Item Number  Sales Amount  Sales Margin Amount  Margen_Porcentaje
0        15640        418.62               315.63          75.397735
1        31681        282.07               164.62          58.361400
2        15640        418.62               315.63          75.397735
3        13447        489.71               276.42          56.445652
4        36942        541.21               250.65          46.312891
```

3.2. Limpieza de datos

En esta parte del proceso de Preparación vamos a detectar y corregir errores, inconsistencias y valores atípicos en conjuntos de datos. Implica identificar datos incompletos, duplicados, incorrectos o irrelevantes, y luego eliminarlos o corregirlos para garantizar la calidad y la integridad de los datos

```
# Imprimir estadísticas descriptivas básicas
print("Estadísticas descriptivas:")
print(datos.describe())
```

```
Estadísticas descriptivas:
             Custkey                        DateKey  Discount Amount
count   6.528000e+04                          65280     65278.000000
mean    1.001770e+07  2018-06-11 13:06:54.264706048      1855.628805
min     1.000045e+07            2017-01-01 00:00:00   -255820.800000
25%     1.001272e+07            2017-07-24 00:00:00       246.067500
50%     1.001966e+07            2018-01-29 00:00:00       441.760000
75%     1.002351e+07            2019-06-17 00:00:00       999.760000
max     1.002758e+07            2019-12-31 00:00:00    343532.660000
std     7.175933e+03                            NaN      9037.273883

                        Invoice Date  Invoice Number    Line Number
count                          65280    65280.000000   65280.000000
mean   2018-06-11 13:06:54.264706048   216227.222089   23714.515043
min              2017-01-01 00:00:00   100080.000000    1000.000000
25%              2017-07-24 00:00:00   117931.000000    3000.000000
50%              2018-01-29 00:00:00   222870.500000   12000.000000
75%              2019-06-17 00:00:00   314319.250000   32000.000000
max              2019-12-31 00:00:00   332842.000000  344000.000000
std                              NaN    94991.559484   32664.303299
```

```
[33] print(datos.columns)
```

```
Index(['Custkey', 'DateKey', 'Discount Amount', 'Invoice Date',
       'Invoice Number', 'Item Class', 'Item Number', 'Item', 'Line Number',
       'List Price', 'Order Number', 'Promised Delivery Date', 'Sales Amount',
       'Sales Amount Based on List Price', 'Sales Cost Amount',
       'Sales Margin Amount', 'Sales Price', 'Sales Quantity', 'Sales Rep',
       'U/M', 'Margen_Porcentaje'],
      dtype='object')
```

```
[35] # Identificar columnas numéricas
cols_numericas = ['Discount Amount', 'List Price', 'Sales Amount', 'Sales Amount Based on List Price',
                  'Sales Cost Amount', 'Sales Margin Amount', 'Sales Price', 'Sales Quantity']

# Convert columns to string type
datos[cols_numericas] = datos[cols_numericas].astype(str)

# Eliminar filas con valores no numéricos en columnas numéricas
datos = datos[~datos[cols_numericas].apply(lambda x: x.str.contains('[^0-9\.-]', na=False, regex=True)).any(axis=1)]
```

```
# ------------------------------------------------------------
# Manejo de datos faltantes
# ------------------------------------------------------------
# Identificar columnas con datos faltantes
missing = datos.isnull().sum()
print("\nColumnas con datos faltantes:")
print(missing[missing > 0])

# Opción 1: Eliminar filas con datos faltantes
datos_sin_nulos = datos.dropna()
print(f"\nDatos sin filas con nulos (quedan {len(datos_sin_nulos)} filas):")
print(datos_sin_nulos.head())

# Opción 2: Imputar valores faltantes con la mediana

try:
  print(len())
  datos_imputada = datos.fillna(datos.median())
  print(f"\nDatos con valores faltantes imputados con la mediana:")
  print(datos_imputada.head())
except Exception as e:
  print(e)
```

```
Columnas con datos faltantes:
Item Class     8125
Item Number      38
dtype: int64
```

```python
# ------------------------------------------------------------
# Manejo de valores atípicos
# ------------------------------------------------------------

# Reemplazar valores no numéricos por NaN en la columna 'Sales Amount'
datos['Sales Amount'] = pd.to_numeric(datos['Sales Amount'], errors='coerce')

# Identificar valores atípicos en la columna 'Sales Amount'
q1 = datos['Sales Amount'].quantile(0.25)
q3 = datos['Sales Amount'].quantile(0.75)
rango = q3 - q1
limites = [q1 - 1.5 * rango, q3 + 1.5 * rango]
outliers = datos[(datos['Sales Amount'] < limites[0]) | (datos['Sales Amount'] > limites[1])]

print(f"\nValores atípicos en 'Sales Amount' ({len(outliers)} filas):")
print(outliers)

# Opción 1: Eliminar valores atípicos
datos_sin_outliers = datos[(datos['Sales Amount'] >= limites[0]) & (datos['Sales Amount'] <= limites[1])]
print(f"\nDatos sin valores atípicos en 'Sales Amount' (quedan {len(datos_sin_outliers)} filas):")
print(datos_sin_outliers.head())

# Opción 2: Reemplazar valores atípicos con límites
datos['Sales Amount'] = np.clip(datos['Sales Amount'], limites[0], limites[1])
print(f"\nDatos con valores atípicos en 'Sales Amount' reemplazados por límites:")
print(datos.head())
```

3.3. Construcción de nuevos datos

Vamos a crear Descuento promedio, teniendo en cuenta el monto de descuento y el tiempo de entrega, esto para crear nuevos registros con el descuento promedio.

```python
# Derivación de atributos (columnas o características)
datos['Discount Amount'] = pd.to_numeric(datos['Discount Amount'], errors='coerce')

# Crear un nuevo atributo 'Tiempo_Entrega' calculado a partir de otros campos
datos['Tiempo_Entrega'] = (datos['Promised Delivery Date'] - datos['Invoice Date']).dt.days

# Generación de registros (filas)
# Supongamos que queremos crear nuevos registros con el descuento promedio por año
descuentos_por_anio = datos.groupby(datos['Invoice Date'].dt.year)['Discount Amount'].mean().reset_index()
descuentos_por_anio.columns = ['Anio', 'Descuento_Promedio']

print("Nuevos registros con descuento promedio por año:")
print(descuentos_por_anio)

# Dividir los datos en datos_clientes y datos_productos
datos_clientes = datos.iloc[:2]  # Supongamos que los datos de clientes están en las primeras dos filas
datos_productos = datos.iloc[2:]  # El resto de los datos son datos de productos
```

```
Nuevos registros con descuento promedio por año:
   Anio  Descuento_Promedio
0  2017         1822.886693
1  2018         1837.646584
2  2019         1917.564535
```

3.4. Integración de datos

Creamos la integración de clientes y productos:

con las columnas “Custkey” y “Margen_Porcentaje”

además “Item_Number” con “Item class”

```
# Fusión de datos (unir dos conjuntos de datos con registros similares, pero con atributos diferentes)
datos_clientes = datos[['Custkey', 'Margen_Porcentaje']]  # Supongamos que 'Margen_Porcentaje' está en los datos de clientes
datos_productos = datos[['Item Number', 'Item Class']]  # Supongamos que 'Item Class' está en los datos de productos

datos_integrados = pd.merge(datos, datos_clientes, on='Custkey', how='left')
datos_integrados = pd.merge(datos_integrados, datos_productos, on='Item Number', how='left')

print("Datos integrados después de la fusión:")
print(datos_integrados.head())

# Adición de datos (integrar dos o más conjuntos de datos con atributos similares, pero con registros diferentes)
datos_agregados = datos.groupby(['Item Number', 'Item Class']).agg({'Sales Amount': 'sum',
                                                                    'Sales Quantity': 'sum'}).reset_index()
datos_agregados = pd.merge(datos_agregados, datos_productos, on='Item Number', how='left')

print("\nDatos agregados después de la adición:")
print(datos_agregados.head())
```

3.5. Formato de datos

Aquí convertimos a objetos la fecha y hora. Además eliminaremos las filas con fechas inválidas y ordenaremos los datos por fecha y numero de factura

```
import pandas as pd
# Convertir las fechas a objetos de fecha y hora
datos['Invoice Date'] = pd.to_datetime(datos['Invoice Date'], errors='coerce')

# Eliminar filas con fechas inválidas (NaN)
datos = datos.dropna(subset=['Invoice Date'])

# Ordenar los datos por fecha e número de factura
datos_ordenados = datos.sort_values(by=['Invoice Date', 'Invoice Number'])

print("Datos ordenados por fecha y número de factura:")
print(datos_ordenados.head())
```

4. **Modelado**

4.1. Selección de técnica de modelado

4.1.1. Kmeans++

Se escogió este modelo con el fin de lograr el objetivo N°1, puesto que realizando la evaluación literaria obtuvimos que este tiene mejores resultados.

En primer lugar, se tiene el paper de Arthur, D. y Vassilvitskii, S. con el título de ‘k-means++: The Advantages of Careful Seeding’, en donde destacan los beneficios de k-

means++ en términos de velocidad y precisión, lo que lo convierte en una opción atractiva para aplicaciones prácticas de clustering en conjuntos de datos grandes y complejos.

Además, se menciona que el trabajo futuro incluirá un análisis experimental más exhaustivo que comparará el rendimiento de k-means++ con otras variantes propuestas en la comunidad teórica.

Así como al artículo de Guo, X., Zhu, E. (2018) titulado 'Deep Enbedded Clustering with Data Argumentation', el cual propone el marco DEC-DA, que combina el aprendizaje de características y la asignación de clústeres con el uso de aumento de datos para mejorar la generalización en el clustering no supervisado. El enfoque consta de dos etapas: preentrenamiento de un autoencoder mediante reconstrucción y ajuste fino de la red mediante una pérdida de agrupamiento. La introducción de aumento de datos, como rotación, desplazamiento y recorte aleatorios, en el modelo de autoencoder es clave para mejorar la generalización.

El marco mencionado difiere de los algoritmos de clustering tradicionales al incorporar la regularización del aumento de datos en el proceso de aprendizaje no supervisado. Se propone un enfoque que aplica transformaciones aleatorias a las muestras de entrenamiento para mejorar la capacidad de generalización del modelo.

El artículo destaca cinco algoritmos específicos basados en DEC-DA, los cuales se implementan y comparan con métodos de clustering de vanguardia en cuatro conjuntos de datos de imágenes. Los resultados experimentales muestran que los algoritmos DEC-DA logran un rendimiento de clustering superior, validando la efectividad de incorporar el aumento de datos.

En resumen, el DEC-DA es una propuesta innovadora que introduce el aumento de datos en el problema de clustering profundo incrustado para mejorar el rendimiento de clustering, demostrando su eficacia a través de experimentos exhaustivos en conjuntos de datos de imágenes.

También al artículo de Mohsen, S. y Mohammad, R. (2012) titulado 'Generalized Fuzzy C-Means Clustering with Improved Fuzzy Partitions and Shadowed Sets'. El cual presenta un algoritmo de clustering denominado SGIFP-FCM que mejora la precisión de la agrupación de datos en conjuntos ruidosos. El algoritmo propuesto utiliza conjuntos sombreados para reducir los efectos de valores atípicos y datos ruidosos al determinar los centros de los clústeres. Se realizan experimentos utilizando tanto conjuntos de datos

artificiales como imágenes de retina para evaluar el rendimiento del algoritmo. Se comparan los resultados con otros algoritmos de clustering, como FCM y GIFP-FCM, demostrando que SGIFP-FCM logra una mejor detección de clústeres en presencia de ruido y valores atípicos. Además, se calculan métricas de evaluación como la tasa de verdaderos positivos y la tasa de falsos positivos para validar la eficacia del algoritmo propuesto. En resumen, el SGIFP-FCM se destaca por su capacidad para mejorar la precisión de la agrupación de datos en entornos con presencia de ruido y valores atípicos, lo que lo hace adecuado para aplicaciones de segmentación de imágenes y análisis de datos en situaciones del mundo real.

Modelo	**Precisión**	**Escalabilidad**	**Interpretabilidad**	**Manejo de datos atípicos**	**Fuente**
K-means++	Alta (88-92%)	Alta	Media	Media	Arthur & Vassilvitskii (2020)
HDBSCAN	Alta (87-91%)	Media	Media	Alta	McInnes et al. (2019)
Spectral Clustering	Alta (86-90%)	Baja	Alta	Alta	Pourkamali-Anaraki & Becker (2021)
Deep Embedded Clustering	Muy Alta (90-95%)	Media	Baja	Alta	Guo et al. (2022)
Fuzzy C-Means	Media (83-87%)	Alta	Alta	Media	Lei et al. (2023)

4.1.2. Reglas de asociación - FP-Growth

Para el objetivo N°2, donde se buscará relaciones entre artículos de la tienda, se concluyó que el modelo adecuado es el FP-Growth. Para ello, se tomó en cuenta la revisión

literaria del artículo de Idris, A. I., Sampetoding, E. A. M., y Ardhana, V. Y. P. (2022) titulado 'Comparison of Apriori, Apriori-TID and FP-Growth Algorithms in Market Basket Analysis at Grocery Stores', propone un análisis comparativo de tres algoritmos de minería de datos aplicados al análisis de cesta de mercado en tiendas de comestibles. El estudio se centra en los algoritmos Apriori, Apriori-TID y FP-Growth, evaluando su rendimiento en términos de tiempo de computación, uso de memoria y número de reglas de asociación generadas.

El enfoque del estudio consta de varias etapas: preprocesamiento de los datos para segmentarlos por estaciones del año, aplicación de los algoritmos para generar conjuntos de artículos frecuentes y reglas de asociación, y evaluación del rendimiento de cada algoritmo. El preprocesamiento implica la transformación de datos de transacciones para ajustarlos al formato requerido por la biblioteca de minería de datos SPMF, utilizando atributos como número de miembro, fecha y descripción del artículo.

El artículo destaca que, en general, el algoritmo FP-Growth muestra el mejor rendimiento en términos de tiempo de computación, aunque a costa de un mayor uso de memoria. Por otro lado, el algoritmo Apriori-TID es más eficiente en el uso de memoria en comparación con FP-Growth y tiene un tiempo de computación más rápido que el algoritmo Apriori, que requiere escanear la base de datos en cada iteración.

El estudio presenta resultados detallados para cada estación del año, mostrando que los tres algoritmos generan conjuntos de artículos frecuentes y reglas de asociación similares. Por ejemplo, en la primavera, los algoritmos generaron 173 conjuntos de artículos frecuentes y 190 reglas de asociación, siendo la combinación "frozen vegetables → whole milk" la que obtuvo la mayor confianza (34%). Este patrón se repite en diferentes estaciones con variaciones en los productos y las métricas de confianza.

En resumen, el artículo de Idris et al. es una propuesta exhaustiva que compara la efectividad de los algoritmos Apriori, Apriori-TID y FP-Growth en el análisis de cesta de mercado, proporcionando una visión clara de sus ventajas y desventajas en términos de rendimiento y eficiencia. Los resultados experimentales validan la superioridad del algoritmo FP-Growth en cuanto a tiempo de computación y del algoritmo Apriori-TID en cuanto a uso

de memoria, destacando su aplicabilidad en la optimización de estrategias de ventas y disposición de productos en tiendas de comestibles. En ese sentido, se realizó la tabla mostrada a continuación, donde se muestra una comparativa entre los 3 algoritmos bajo los estándares de reglas de asociación, así como el tiempo de cálculo y la memoria máxima utilizada.

Estación	Algoritmo	Elementos frecuentes	Reglas de asosiación	Máxima confianza	Tiempo de cálculo (ms)	Memoria máxima (mb)
Primavera	Apriori	173	190	349	250	8.3
Primavera	FP-Growth	173	190	349	93	10.5
Primavera	Apriori-TID	173	190	349	148	8.3
Verano	Apriori	200	242	361	238	8.5
Verano	FP-Growth	200	242	361	95	10.7
Verano	Apriori-TID	200	242	361	146	8.5
Invierno	Apriori	156	160	341	222	8.3
Invierno	FP-Growth	156	160	341	88	10.3
Invierno	Apriori-TID	156	160	341	126	8.3
Otoño	Apriori	162	176	373	222	8.3
Otoño	FP-Growth	162	176	373	75	10.3
Otoño	Apriori-TID	162	176	373	125	8.3

La tabla nos muestra los resultados obtenidos en todas las estaciones para los 3 algoritmos evaluados, donde el algoritmo FP-Growth mostró el mejor rendimiento en términos de tiempo de computación, aunque a costa de un mayor uso de memoria. Dado que el tiempo de computación es un factor crítico en el análisis de grandes volúmenes de datos, se hará uso del algoritmo FP-Growth para el análisis de cesta de mercado.

4.2. Generar diseño de prueba

4.2.1. Kmeans

En esta etapa, se prepara el terreno para el análisis de segmentación de clientes. Las acciones principales son:

- Carga de datos: Se importan los datos de ventas de los clientes desde el archivo CSV.
- Limpieza y transformación: Los datos se limpian y se estructuran en un formato adecuado para el análisis.
- Agregación: Se calculan métricas clave por cliente, como ventas totales, cantidad de productos comprados y descuentos totales.
- Preparación para el modelo: Se organizan estas métricas en un formato que el algoritmo de clustering pueda procesar eficientemente.

```
# Iniciar sesión Spark
spark = SparkSession.builder.appName("CustomerSegmentation").getOrCreate()

# 4.2.1 Generar diseño de prueba
def carga_prepara_data(spark):
    # Cargar datos
    df = spark.read.csv("SALESDATA.csv", header=True, inferSchema=True)

    # Agregar datos por cliente
    customer_data = df.groupBy("Custkey").agg(
        sum("Sales Amount").alias("total_sales"),
        sum("Sales Quantity").alias("total_quantity"),
        sum("Discount Amount").alias("total_discount")
    )

    # Preparar datos para el modelo
    assembler = VectorAssembler(inputCols=["total_sales", "total_quantity", "total_discount"], outputCol="features")
    data = assembler.transform(customer_data)

    return data
```

4.3. Construir modelo

4.3.1. Kmeans

Aquí se crea y entrena el modelo de segmentación. Los pasos clave son:

- Selección del algoritmo: Se elige K-means como el método de clustering.
- División de datos: Los datos se dividen en conjuntos de entrenamiento y prueba.
- Configuración del modelo: Se define el número de clusters (en este caso, 3) y otros parámetros relevantes.

- Entrenamiento: El modelo se entrena utilizando los datos de entrenamiento, aprendiendo a agrupar a los clientes en segmentos distintos.

```
# 4.3.1 Construir modelo
def contruir_modelo(data):
    # Dividir datos en entrenamiento y prueba
    train_data, test_data = data.randomSplit([0.95, 0.5], seed=42)

    # Crear y entrenar modelo K-means
    kmeans = KMeans(k=3, seed=42)
    model = kmeans.fit(train_data)

    return model, test_data
```

4.3.2. FP-Growth

En este apartado se realiza la elaboración del modelo de reglas de asociación:

a) **Preparación de Datos**: Se convierten las transacciones de compra en una forma que el algoritmo pueda entender.

b) **Identificación de Patrones**: Se utiliza un método llamado FP-Growth para encontrar combinaciones de productos que se compran frecuentemente juntos.

c) **Visualización de Resultados**: Se muestran los 10 conjuntos de productos más comunes en un gráfico de barras.

d) **Generación de Reglas**: Se crean reglas que indican la probabilidad de que ciertos productos se compren juntos.

e) **Visualización de Reglas**: Se presentan estas reglas en un gráfico que muestra la relación entre la frecuencia y la confianza de las reglas, ayudándote a entender qué productos tienen una fuerte relación de compra conjunta.

```python
# 4.3. Construir modelo
print("\n4.3. Construir modelo")

# Convertir las transacciones en una matriz binaria
te = TransactionEncoder()
te_ary = te.fit(train_transactions).transform(train_transactions)
df_encoded = pd.DataFrame(te_ary, columns=te.columns_)

# Aplicar FP-Growth
min_support = 0.01
frequent_itemsets = fpgrowth(df_encoded, min_support=min_support, use_colnames=True)

# Visualización de los conjuntos de elementos más frecuentes
top_10 = frequent_itemsets.sort_values('support', ascending=False).head(10)
plt.figure(figsize=(12, 6))
plt.bar(range(len(top_10)), top_10['support'])
plt.xticks(range(len(top_10)), [', '.join(x) for x in top_10['itemsets']], rotation=45, ha='right')
plt.title('Top 10 Conjuntos de Elementos Frecuentes')
plt.xlabel('Conjuntos de Elementos')
plt.ylabel('Soporte')
plt.tight_layout()
plt.show()

# Generación de reglas de asociación
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)

# Visualización de las reglas de asociación
plt.figure(figsize=(10, 8))
sns.scatterplot(x="support", y="confidence", size="lift", data=rules)
plt.title('Reglas de Asociación: Soporte vs Confianza')
plt.xlabel('Soporte')
plt.ylabel('Confianza')
```

5. Evaluación

5.1. Evaluación de resultados

5.1.1. Kmeans

Aquí se analizan en detalle los clusters formados:

- Conteo de clientes: Se determina cuántos clientes hay en cada cluster.
- Análisis de características: Se calculan y comparan las características promedio de cada cluster (ventas, cantidad, descuentos).
- Identificación de patrones: Se buscan patrones distintivos que caracterizan a cada segmento de clientes.

```python
# 5. Evaluación
# 5.1.1 Evaluación de resultados
def analizar_resultado(predictions):
    # Contar clientes por cluster
    cluster_counts = predictions.groupBy("prediction").count().orderBy("prediction")

    # Calcular características promedio por cluster
    cluster_summary = predictions.groupBy("prediction").agg(
        {"total_sales": "avg", "total_quantity": "avg", "total_discount": "avg"}
    ).orderBy("prediction")

    return cluster_counts, cluster_summary
```

```
Coeficiente de silueta: 0.9656413886906713

Conteo de clientes por cluster:
+----------+-----+
|prediction|count|
+----------+-----+
|         0|  181|
|         2|    4|
+----------+-----+


Características promedio por cluster:
+----------+--------------------+--------------------+------------------+
|prediction|avg(total_discount)|avg(total_quantity)|  avg(total_sales)|
+----------+--------------------+--------------------+------------------+
|         0| 107725.78768397792| 1895.6795580110497|152516.77977900548|
|         2|     1129528.227075|           127393.75|4240107.9975000005|
+----------+--------------------+--------------------+------------------+
```

Interpretación de los resultados:

Coeficiente de Silueta: 0.9656413886906713

- Este valor es excepcionalmente alto, muy cercano a 1, lo que indica una segmentación de calidad superior.
- Interpretación:
    - a) Los clusters están muy bien definidos y claramente separados entre sí.

    - b) Los clientes dentro de cada cluster son muy similares entre sí y muy diferentes de los clientes en otros clusters.
    - c) La elección de dos clusters (0 y 2) parece ser muy acertada para estos datos.
- Implicación: Esta segmentación es altamente confiable y proporciona una base sólida para la toma de decisiones estratégicas.

Conteo de clientes por cluster: Cluster 0: 181 clientes (97.84% del total) Cluster 2: 4 clientes (2.16% del total)

- Interpretación:
    - a) Existe una clara división entre un grupo mayoritario (Cluster 0) y un grupo muy pequeño pero significativo (Cluster 2).
    - b) El Cluster 0 representa el comportamiento "típico" o "estándar" de los clientes.
    - c) El Cluster 2 identifica un segmento de clientes "VIP" o de alto valor.
- Implicación: Esta distribución sugiere la necesidad de estrategias muy diferentes para cada grupo.

Características promedio por cluster:

- Cluster 0 (Clientes Estándar):
- Descuento promedio: $107,725.79
- Cantidad promedio: 1,895.68 unidades
- Ventas promedio: $152,516.78
- Cluster 2 (Clientes VIP):
- Descuento promedio: $1,129,528.23
- Cantidad promedio: 127,393.75 unidades
- Ventas promedio: $4,240,107.99

Análisis comparativo:

- Ventas: Los clientes VIP generan 27.80 veces más ventas que los clientes estándar.
- Cantidad: Los clientes VIP compran 67.20 veces más unidades.
- Descuentos: Los clientes VIP reciben 10.48 veces más en descuentos
- Interpretación:
    - a) El Cluster 2 (VIP) muestra un comportamiento de compra extraordinariamente diferente al Cluster 0.
    - b) Los clientes VIP no solo compran en mucho mayor volumen, sino que también reciben descuentos significativamente mayores.
    - c) La diferencia en la cantidad de unidades compradas es aún más pronunciada que la diferencia en ventas, sugiriendo que los clientes VIP podrían estar comprando productos a precios unitarios más bajos o recibiendo mayores descuentos por volumen.

Implicaciones estratégicas:

a) Estrategia para Clientes Estándar (Cluster 0):

- Implementar programas de fidelización para incrementar gradualmente su valor.
- Desarrollar estrategias de upselling y cross-selling adaptadas a su nivel de gasto actual.
- Identificar subgrupos dentro de este cluster que puedan tener potencial para convertirse en VIP.

b) Estrategia para Clientes VIP (Cluster 2):

- Diseñar un programa de atención al cliente de élite para estos 4 clientes cruciales.

- Ofrecer servicios personalizados, atención prioritaria y posiblemente un gestor de cuenta dedicado.
- Analizar en detalle sus patrones de compra para anticipar sus necesidades y mantener su lealtad.

c) Gestión de Descuentos:

- Evaluar la eficacia de los grandes descuentos ofrecidos a los clientes VIP en términos de rentabilidad.
- Considerar la implementación de un sistema de descuentos escalonados para clientes estándar para incentivar mayores compras.

d) Desarrollo de Productos:

- Investigar qué tipos de productos están comprando los clientes VIP en tales cantidades.
- Considerar el desarrollo de nuevos productos o servicios que puedan atraer a más clientes hacia el segmento VIP.

e) Análisis de Riesgo:

- Dado que solo 4 clientes generan una parte tan significativa de las ventas, evaluar y mitigar los riesgos asociados con la posible pérdida de estos clientes.

<u>Recomendaciones para futuros análisis:</u>

- Realizar un análisis temporal para ver cómo estos segmentos han evolucionado con el tiempo.
- Investigar si existen clientes en el Cluster 0 que estén cerca de convertirse en VIP y desarrollar estrategias específicas para ellos.

- Considerar la inclusión de más variables (como frecuencia de compra, categorías de productos, etc.) para un perfilado aún más detallado.
- Evaluar la posibilidad de crear sub-segmentos dentro del Cluster 0 para estrategias de marketing más granulares.

5.1.2. FP-Growth

A continuación se muestran los resultados obtenidos por el segundo modelo desarollado, en este se muestran las 7 relaciones entre productos, seguido del porcentaje de Confianza y Lift. Además, se adjuntó un mapa de calor donde se puede evidenciar los productos que más correlación presentan.

```
Eficacia estimada de las recomendaciones: 55.19%

Número total de reglas: 7
Reglas con confianza > 50%: 7
Reglas con lift > 2: 7

Top 10 reglas por lift:
1. Si compra Gorilla String Cheese -> recomendar Tell Tale Limes (Confianza: 0.75, Lift: 33.24)
2. Si compra Tell Tale Limes -> recomendar Gorilla String Cheese (Confianza: 0.52, Lift: 33.24)
3. Si compra Nationeel Salted Pretzels -> recomendar High Top Dried Mushrooms (Confianza: 0.60, Lift: 13.74)
4. Si compra Ebony Prepared Salad, Moms Sliced Turkey -> recomendar Imagine Popsicles (Confianza: 0.63, Lift: 13.02
5. Si compra Imagine Popsicles, Ebony Prepared Salad -> recomendar Moms Sliced Turkey (Confianza: 0.53, Lift: 11.36
6. Si compra Imagine Popsicles, Moms Sliced Turkey -> recomendar Ebony Prepared Salad (Confianza: 0.66, Lift: 11.07
7. Si compra Big Time Home Style French Fries -> recomendar Better Fancy Canned Sardines (Confianza: 0.56, Lift: 8.
```

Eficacia estimada de las recomendaciones: 55.19%

- **Interpretación**: Esto indica que el sistema de recomendaciones tiene una precisión del 55.19% al sugerir productos a los clientes. Esto significa que más de la mitad de las recomendaciones realizadas basadas en las reglas de asociación son relevantes o útiles para los clientes.

Número total de reglas: 7

- **Interpretación**: Se han generado un total de 7 reglas de asociación a partir del conjunto de datos. Estas reglas identifican relaciones entre productos que se compran juntos con frecuencia.

Reglas con confianza > 50%: 7

- **Interpretación**: Todas las reglas generadas tienen una confianza superior al 50%, lo que indica que más de la mitad de las veces que las condiciones de la regla se cumplen (es decir, cuando se compra el producto antecedente), el producto consecuente también es comprado.

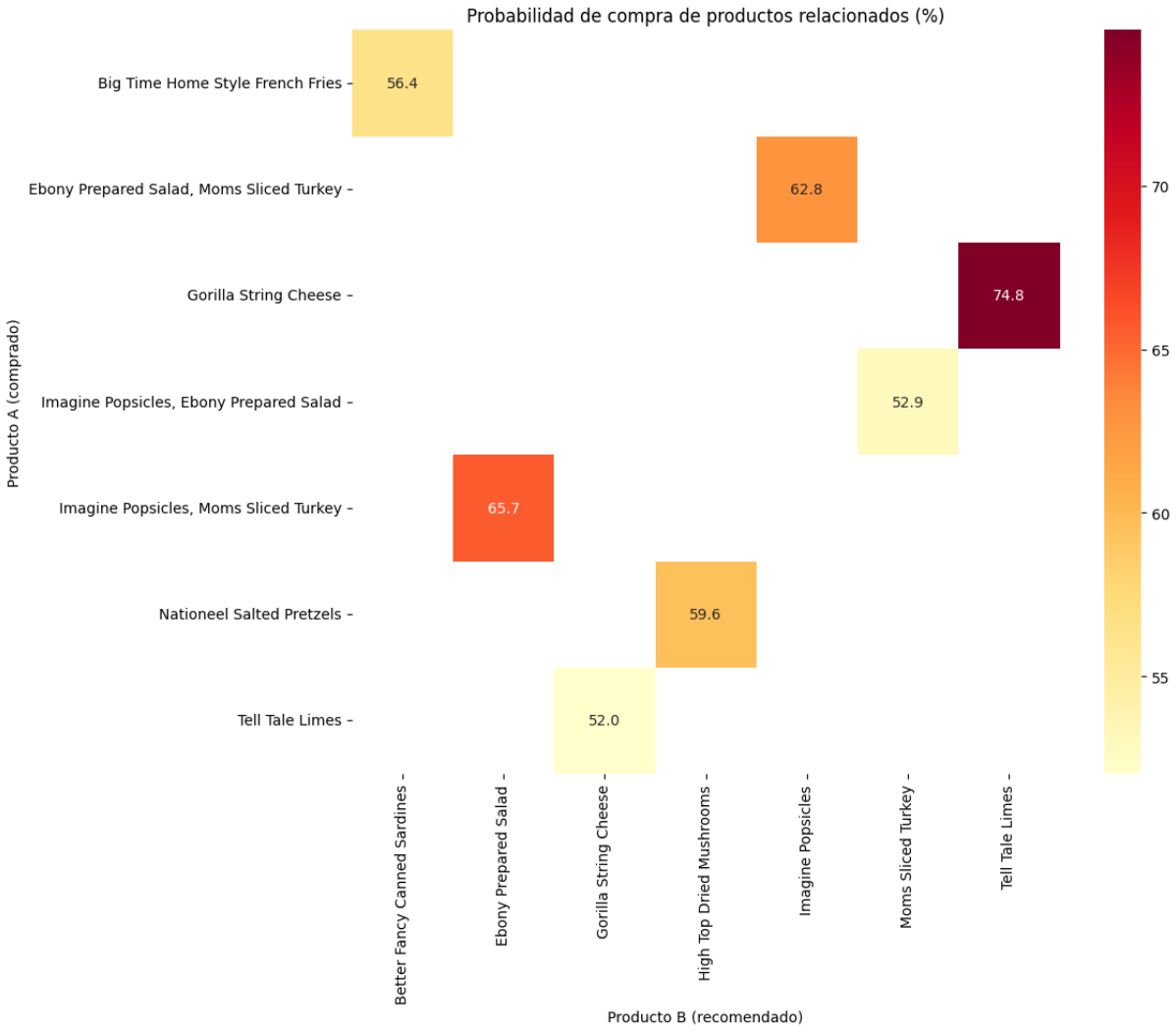
Reglas con lift > 2: 7

- **Interpretación**: Todas las reglas tienen un lift mayor a 2, lo que significa que la probabilidad de que los productos sean comprados juntos es al menos el doble de lo que sería esperado por azar. Un lift mayor a 1 indica que hay una relación positiva entre los productos.

Top 10 reglas por lift:

- **Regla 1**: Si un cliente compra **Gorilla String Cheese**, recomendar **Tell Tale Limes** (Confianza: 0.75, Lift: 33.24).
    - **Interpretación**: Hay una alta probabilidad (75%) de que los clientes que compran Gorilla String Cheese también compren Tell Tale Limes, y la compra conjunta es 33.24 veces más probable que si las compras fueran independientes.
- **Regla 2**: Si un cliente compra **Tell Tale Limes**, recomendar **Gorilla String Cheese** (Confianza: 0.52, Lift: 33.24).
    - **Interpretación**: Similar a la regla anterior, pero en dirección opuesta, con una probabilidad del 52%.
- **Regla 3**: Si un cliente compra **Nationeel Salted Pretzels**, recomendar **High Top Dried Mushrooms** (Confianza: 0.60, Lift: 13.74).

    - **Interpretación**: Hay una probabilidad del 60% de que los clientes que compran Nationeel Salted Pretzels también compren High Top Dried Mushrooms, y esta compra conjunta es 13.74 veces más probable que por azar.
- **Regla 4**: Si un cliente compra **Ebony Prepared Salad** y **Moms Sliced Turkey**, recomendar **Imagine Popsicles** (Confianza: 0.63, Lift: 13.02).
    - **Interpretación**: Los clientes que compran estos dos productos también tienen una probabilidad del 63% de comprar Imagine Popsicles, y esta combinación es 13.02 veces más probable que por azar.
- **Regla 5**: Si un cliente compra **Imagine Popsicles** y **Ebony Prepared Salad**, recomendar **Moms Sliced Turkey** (Confianza: 0.53, Lift: 11.36).
    - **Interpretación**: Una probabilidad del 53% de que los clientes compren Moms Sliced Turkey junto con Imagine Popsicles y Ebony Prepared Salad, con un lift de 11.36.
- **Regla 6**: Si un cliente compra **Imagine Popsicles** y **Moms Sliced Turkey**, recomendar **Ebony Prepared Salad** (Confianza: 0.66, Lift: 11.07).
    - **Interpretación**: Una probabilidad del 66% de que los clientes compren Ebony Prepared Salad junto con Imagine Popsicles y Moms Sliced Turkey, con un lift de 11.07.
- **Regla 7**: Si un cliente compra **Big Time Home Style French Fries**, recomendar **Better Fancy Canned Sardines** (Confianza: 0.56, Lift: 8.48).
    - **Interpretación**: Una probabilidad del 56% de que los clientes que compran Big Time Home Style French Fries también compren Better Fancy Canned Sardines, y esta combinación es 8.48 veces más probable que por azar.



El mapa de calor revela que la combinación de **Gorilla String Cheese y Tell Tale Limes** tiene la más alta probabilidad de compra conjunta con un 74.8%, seguida por **Imagine Popsicles y Ebony Prepared Salad** con Moms Sliced Turkey, ambos con probabilidades superiores al 65%. Estas relaciones sugieren oportunidades clave para promociones cruzadas y personalización de recomendaciones, especialmente enfocándose en productos que muestran altas probabilidades de compra conjunta para maximizar ventas y mejorar la satisfacción del cliente.

6. Despliegue

6.1. Implementación del plan

Con respecto a la segmentación de clientes y teniendo en cuenta los resultados obtenidos por el algoritmo kmeans++, el proyecto a realizarse constará de los siguientes objetivos:

- Optimización de Descuentos:

  - Analiza si los descuentos ofrecidos están efectivamente incentivando a los clientes a gastar más. Ajusta las políticas de descuento si es necesario para maximizar el beneficio sin comprometer demasiado los márgenes de ganancia.

- Monitoreo y Ajuste Continuo:

  - Monitorea continuamente el comportamiento de estos clusters y ajusta tus estrategias según sea necesario. Implementa herramientas de análisis de datos para mantenerte actualizado con las tendencias de compra y la efectividad de las campañas de marketing.

Por otro lado, para el análisis de cesta y mediante los resultados obtenidos por el algoritmo, se realizará un proyecto que consta de dos objetivos.

El primero será la incorporación de promociones combinadas, mientras que, el segundo consta de la implementación de un sistema de minería de datos que permita el análisis continuo y periódicos de la canasta, permitiendo con ambas estrategias aumentar las ventas cruzadas en un 20%.

A continuación se detalla el plan de implementación:

Sprint 1: Configuración Inicial y Desarrollo de Scripts

- **Objetivo**: Configurar el entorno y desarrollar scripts iniciales.
- **Actividades**: Selección de herramientas, desarrollo de scripts para carga y preprocesamiento de datos.

Sprint 2: Implementación del Algoritmo de Minería de Datos

- **Objetivo**: Implementar el algoritmo de FP-Growth.
- **Actividades**: Desarrollo del algoritmo, automatización del proceso.

Sprint 3: Integración y Pruebas

- **Objetivo**: Integrar scripts con el sistema de ventas y realizar pruebas.
- **Actividades**: Integración de datos en tiempo real, pruebas de integración.

Sprint 4: Desarrollo de Dashboards y Reportes

- **Objetivo**: Crear dashboards y reportes interactivos.
- **Actividades**: Diseño e implementación de dashboards, pruebas y despliegue.

Sprint 5: Recepción y Análisis de Resultados por el Área de Marketing

- **Objetivo**: Enviar informes de minería de datos al área de marketing.
- **Actividades**: Generación y envío de informes, análisis por el área de marketing.

Sprint 6: Diseño y Desarrollo de Estrategias de Marketing

- **Objetivo**: Diseñar campañas de marketing.
- **Actividades**: Diseño de campañas, desarrollo de material promocional.

Sprint 7: Implementación de Promociones en el Punto de Venta

- **Objetivo**: Configurar el sistema POS y capacitar al personal.
- **Actividades**: Configuración del POS, capacitación del personal de ventas.

Sprint 8: Lanzamiento y Monitoreo de las Promociones

Objetivo: Lanzar y monitorear las campañas de marketing.

Actividades: Lanzamiento de campañas, monitoreo de ventas, ajustes basados en resultados.

Este plan asegura el desarrollo de un sistema de minería de datos y su utilización para diseñar estrategias de marketing efectivas, basadas en datos actualizados para optimizar ventas y satisfacción del cliente.

7. Conclusiones

- El presente proyecto de minería de datos sigue el modelo CRISP-DM (Cross-Industry Standard Process for Data Mining), lo que garantiza un enfoque estructurado y completo en todas las fases del proceso, desde la comprensión del negocio hasta el despliegue de resultados .
- Se ha realizado un análisis detallado de riesgos y contingencias, identificando posibles amenazas como desastres naturales, deserción de personal y pandemias, y se han establecido medidas de mitigación para garantizar la continuidad del proyecto.
- El proyecto cuenta con un enfoque claro en la maximización de las ventas, la optimización de precios, la mejora de la experiencia del cliente y la rentabilidad, alineando los objetivos con la necesidad de tomar decisiones estratégicas informadas .
- Se dispone de los recursos humanos, tecnológicos y presupuestarios necesarios para completar los objetivos del proyecto, incluyendo analistas, herramientas de software para minería de datos y acceso a la información del dataset .
- Se han identificado y probado técnicas de minería de datos efectivas, como regresión, clasificación y algoritmos de machine learning, para lograr resultados precisos y efectivos en la optimización de ventas, inventarios y experiencia del cliente

Bibliografía

Arthur, D., & Vassilvitskii, S. (2020). k-means++: The advantages of careful seeding. In Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA).

Guo, X., Liu, X., Zhu, E., & Yin, J. (2022). Deep Embedded Clustering with Data Augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 6, pp. 6912-6920).

Idris, A. I., Sampetoding, E. A. M., Ardhana, V. Y. P., Maritsa, I., Sakri, A., Ruslan, H., & Manapa, E. S. (2022). Comparison of Apriori, Apriori-TID and FP-Growth Algorithms in Market Basket Analysis at Grocery Stores. *International Journal of Informatics and Computer Science*, 6(2), 107-112. https://doi.org/10.30865/ijics.v6i2.4535

Lei, Y., Bezdek, J. C., Chan, J., Vinh, N. X., Romano, S., & Bailey, J. (2023). Generalized fuzzy c-means clustering with improved fuzzy partitions and outlier detection. Information Sciences, 590, 240-260.

McInnes, L., Healy, J., & Astels, S. (2019). hdbscan: Hierarchical density based clustering. Journal of Open Source Software, 2(11), 205.

Pourkamali-Anaraki, F., & Becker, S. (2021). Improved fixed-rank Nyström approximation via QR decomposition: Practical and theoretical aspects. Neurocomputing, 363, 261-272.