Datos del estudiante

**Nombre y apellidos** Nombre y apellidos del estudiante

**Fecha de entrega** 01/01/2022

Análisis de ventas en Odoo: de los datos brutos al *dashboard* visual con Python

**Objetivos de la actividad**

A través de esta actividad, vas a conseguir:

* Comprender la estructura básica de datos en Odoo relacionada con las ventas.
* Extraer y preparar datos de un entorno simulado de Odoo para su análisis.
* Utilizar las librerías Pandas y Matplotlib de Python para realizar un análisis exploratorio de datos (EDA) y crear visualizaciones impactantes.
* Identificar patrones y tendencias en los datos de ventas que puedan informar decisiones empresariales.
* Desarrollar habilidades en la integración y manipulación de datos provenientes de sistemas ERP como Odoo.

**Pautas de elaboración**

Esta actividad consiste en simular la extracción de datos de ventas de un entorno Odoo, limpiarlos, transformarlos y, finalmente, visualizarlos para extraer conclusiones.

**Pasos por seguir:**

* **Simulación de datos de Odoo (archivo CSV):** no tendrás acceso directo a una instancia de Odoo para esta actividad. En su lugar, se te proporcionará un archivo CSV (ventas\_odoo\_simulado.csv) que simula datos de ventas exportados de un módulo de Odoo. Este archivo contendrá columnas como ID\_Pedido, Fecha\_Pedido, Cliente, Producto, Cantidad, Precio\_Unitario, Total\_Linea, Estado\_Pedido, Comercial. El primer paso es descargar y guardar el archivo ventas\_odoo\_simulado.csv en la misma carpeta donde crearás tu *script* de Python.
* **Carga de datos con Pandas:** utiliza la librería Pandas para cargar el archivo CSV en un DataFrame. Inspecciona las primeras filas del DataFrame, los tipos de datos de cada columna y busca valores nulos.
* **Limpieza y preprocesamiento de datos:** asegúrate de que la columna Fecha\_Pedido esté en formato de fecha (datetime). Verifica que Cantidad, Precio\_Unitario y Total\_Linea sean tipos numéricos. Maneja cualquier valor nulo si los encuentras (por ejemplo, rellenando con un valor predeterminado o eliminando la fila, según el contexto). Crea una nueva columna Mes\_Pedido extrayendo el mes de Fecha\_Pedido. Crea una nueva columna Año\_Pedido extrayendo el año de Fecha\_Pedido.
* Análisis exploratorio de datos (EDA) con Pandas:
  + **Ventas totales por mes/año:** calcula las ventas totales por mes y por año.
  + **Top cinco productos más vendidos (por cantidad y por valor):** identifica qué productos generan más ingresos y cuáles se venden en mayor volumen.
  + **Top cinco clientes con mayor gasto:** determina qué clientes son los más valiosos en términos de ingresos.
  + **Ventas por comercial:** analiza el rendimiento de ventas de cada comercial.
  + **Distribución de estados de pedido:** cuenta la frecuencia de cada estado de pedido (Borrado, Confirmado, Cancelado, etc.).
* **Visualización de datos con Matplotlib:**
  + **Gráfico de líneas de ventas mensuales/anuales:** visualiza la tendencia de ventas a lo largo del tiempo.
  + **Gráfico de barras de top cinco productos (por valor):** muestra los productos que más contribuyen a los ingresos.
  + **Gráfico de barras de top cinco clientes:** ilustra los clientes con mayor volumen de compra.
  + **Gráfico de barras de ventas por comercial:** compara el rendimiento de los comerciales.
  + **Gráfico de pastel *(pie chart)* de distribución de estados de pedido:** muestra la proporción de cada estado de pedido. Asegúrate de que tus gráficos tengan títulos claros, etiquetas en los ejes y, si es necesario, leyendas. Guarda las visualizaciones como archivos de imagen (por ejemplo, .png).
* **Interpretación y conclusiones:** redacta un breve informe (en un archivo Markdown o como comentarios en tu *script* Python) con las principales conclusiones obtenidas de tu análisis: ¿qué patrones observaste?, ¿qué información es relevante para la toma de decisiones empresariales?

**Extensión y formato**

* **Extensión:** el *script* de Python debe tener una extensión máxima de entre 50 a 70 líneas de código (excluyendo comentarios y la sección de conclusiones si la haces en un archivo aparte). El informe de conclusiones no debe exceder las 300 palabras.
* **Formato:**
  + **Código Python:** archivo.py. Se recomienda utilizar comentarios para explicar las partes clave del código.
  + **Conclusiones:** puede ser un archivo.md (Markdown) o comentarios dentro del *script* Python.
  + **Gráficos:** archivos .png generados por el script.

**Criterios de evaluación**

|  |  |  |
| --- | --- | --- |
| Análisis de ventas en Odoo | Descripción | Peso  % |
| Carga y preprocesamiento de datos | El código carga correctamente el archivo CSV, identifica y maneja los tipos de datos apropiados y realiza las transformaciones necesarias (fechas, creación de nuevas columnas). Se evidencia un manejo adecuado de valores nulos si están presentes. | 20 % |
| Análisis exploratorio con Pandas | Se realizan cálculos y agregaciones relevantes para cada uno de los puntos solicitados (ventas totales, top productos, top clientes, ventas por comercial, etc.) utilizando las funciones de Pandas de forma eficiente. | 20 % |
| Visualización de datos con Matplotlib | Se generan al menos cinco gráficos diferentes que representan las visualizaciones solicitadas. Los gráficos son claros, tienen títulos, etiquetas y leyendas adecuadas, y reflejan correctamente los datos analizados. | 20 % |
| Interpretación y conclusiones | Se presenta un informe claro y conciso con las principales conclusiones derivadas del análisis. Las observaciones son pertinentes, se basan en los datos visualizados y ofrecen información valiosa para la toma de decisiones empresariales. | 20 % |
| Calidad y claridad del código | El código Python es legible, está bien estructurado, utiliza nombres de variables descriptivos y, si aplica, incluye comentarios pertinentes. Sigue buenas prácticas de programación y es fácil de entender y ejecutar. | 20 % |
|  |  | **100 %** |