CAJAMAR UNIVERSITYHACK2022 DATATHON

RETO:

ATMIRA PHARMA VISUALIZATION

Entrega Fase Nacional

Equipo:

Web del proyecto:

https://share.streamlit.io/dcruzgz/datathonquiros/Master/mifarma_datathonquiros.py

Repositorio GitHub:

https://github.com/dcruzgz/Datathonquiros

Universitat Oberta de Catalunya (UOC)

Antonio González Rodríguez

Daniel Jesús Cruz Garzón

1. Introducción y objetivos

Para conseguir un óptimo funcionamiento de la empresa, es necesaria una correcta interpretación de los datos que esta genera con su actividad. Un análisis minucioso de esta información resulta de vital importancia para ser capaces de tomar decisiones basadas en los datos (en inglés, *data-driven decision making*), que pueden resultar más eficientes que las decisiones basadas en la intuición o en la simple observación.

Con el presente proyecto, nuestro equipo, ***Datathonquiros***, plantea aportar una herramienta que permita visualizar, de manera sencilla, datos relativos a las ventas producidas por la empresa **Atida Mifarma** durante los años 2017 y 2018. El **objetivo** de dicha herramienta es **ser capaces de observar rápidamente aquellas variables que puedan resultarnos de interés a la hora de tomar decisiones en la empresa**. Tras consultar las tablas de datos que se nos han proporcionado para este fin, hemos considerado los siguientes niveles de análisis como los más relevantes para la plataforma que vamos a desarrollar:

1. **Nivel geográfico.** La ubicación en la que se producen las ventas puede sernos de interés a la hora de plantear ofertas y conocer la situación del negocio. Se nos proporcionan los códigos postales de los compradores, por lo que, a través de los dos primeros dígitos de estos, obtendremos la provincia desde la que se ha realizado cada venta. Además, obtendremos la población de cada una de estas provincias para poder extraer un índice de ganancias relativo a la población de cada provincia para que aquellas con una población mucho mayor que otras no sesguen las decisiones que tomemos al observar estos datos.
2. **Nivel temporal**. El momento en el que se realizan las compras puede darnos información que podemos utilizar para lanzar promociones que pueden beneficiar a la empresa y a los clientes. Por este motivo, hemos obtenido el día, el mes, y el año de cada transacción, centrándonos en estos dos últimos a la hora de la visualización para concentrar un notable número de ventas y obtener la mayor información posible a este respecto.
3. **Nivel de producto**. Es importante comprender cómo pueden influir las categorías de cada producto o la marca de estos en el consumo por parte de los clientes para detectar patrones y preferencias. Además, consideramos también relevante estudiar el efecto del descuento ofertado en los productos para detectar soluciones óptimas que ofrezcan un mejor servicio al cliente y optimicen las decisiones de la empresa.
4. **Nivel de cliente**. No podemos olvidarnos de la importancia de los clientes a la hora de tomar nuestras decisiones, pues son uno de los pilares fundamentales para el correcto funcionamiento del negocio. A través de algoritmos de minería de datos, hemos detectado aquellos productos que los clientes han comprado conjuntamente con mayor probabilidad. Además, también hemos realizado nubes de palabras sobre las descripciones de los productos más vendidos. Con esta información podremos mejorar la experiencia de los clientes y facilitamos que productos que satisfagan sus necesidades lleguen a ellos de manera cómoda y sencilla.

El análisis visual de los datos proporcionados a todos estos niveles puede consultarse de manera sencilla e interactiva en nuestra aplicación, que se puede visitar en este enlace.

Las herramientas que hemos utilizado para lograr dicho objetivo han sido:

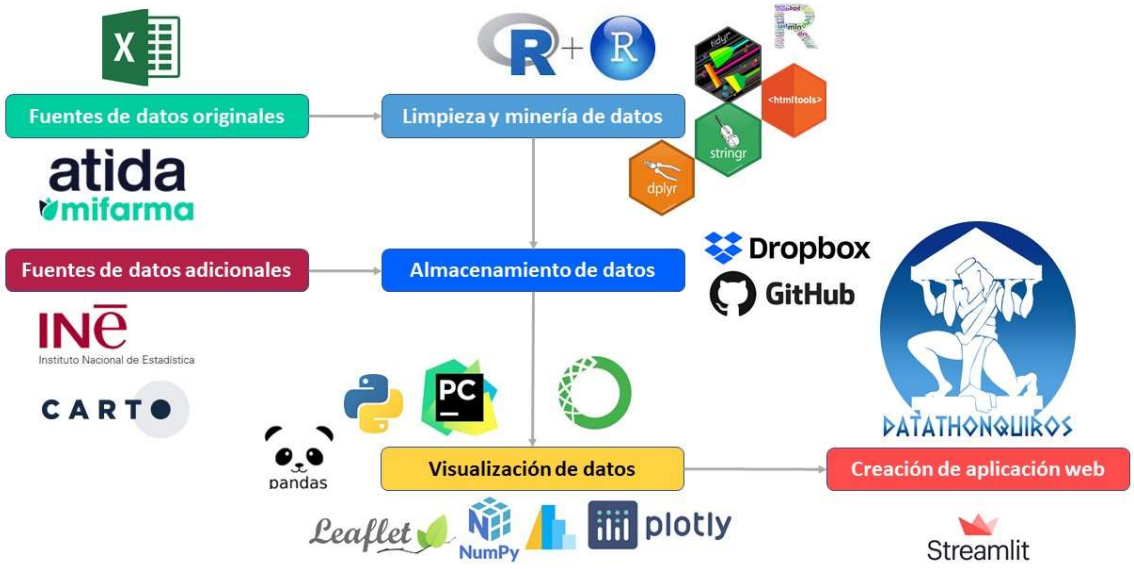
- **Anaconda**. Plataforma desde la que se han obtenido los recursos.
- **R**. Limpieza de datos, minería de datos.
  - Entorno: *Rstudio*. Librerías: *stringr, dplyr, arules, tidyr, tm, SnowballC, wordcloud2, RColorBrewer, RCurl, XML, rsconnect, htmltools*.
- **Python**. Enriquecimiento de datos, visualización de datos.
  - Entorno: *Pycharm*. Librerías: *pandas, numpy, folium* (*leaflet*)*, pyvis, plotly, altair, streamlit*.
- **StreamLit**. Desarrollo de aplicación web.

- **GitHub**. Repositorio con todos los recursos necesarios para la creación de nuestra aplicación.
- **DropBox**. Repositorio con los datasheets de mayor tamaño.

Las fuentes de datos que se han utilizado para desarrollar el presente proyecto han sido:

- **Atida Mifarma**. Tablas proporcionadas para el presente reto.
- **Instituto Nacional de Estadística (INE; enlace, seleccionando la población de 2018)**. Para obtener los códigos postales de las provincias y la población de las mismas.
- **CARTO (enlace)**. Para obtener los datos geográficos (*.geojson*) de las provincias y poder dibujarlas en el mapa correctamente.

El flujo de trabajo que se ha seguido se presenta en el siguiente diagrama:



Y a continuación, pasamos a detallar los pasos que hemos seguido para el desarrollo de nuestra aplicación.

2. Carga, limpieza, y tratamiento de los datos

Los datos de las tablas proporcionadas han sido tratados con ***R*** y han sido depurados de manera que se pueda extraer una gran cantidad de información útil de los mismos. Para una mejor interpretación de los datos, en cada una de las tablas se han realizado las siguientes modificaciones:

- **Tabla "items_ordered_2years.txt":**

1. Antes de cargar el archivo en ***R*** se han eliminado las comillas y se ha pasado el documento a formato .csv. Además, se han eliminado tres filas que ocasionaban errores en los saltos de línea, haciendo que se perdiera información de varios registros. El archivo resultante ha sido llamado "items_ordered_2yearsClean.csv"
2. Se han identificado las variables más relevantes y se ha comprobado si alguna de estas tenía datos nulos.
3. La variable 'city' contiene muchas variaciones de los nombres (tildes, mayúsculas, etc.) de las ciudades, así que no será tenida en cuenta para los análisis. Para mayor homogeneidad, hemos utilizado los dos primeros dígitos del código postal para determinar la provincia desde la que se ha realizado la compra. Estos han sido almacenados en una nueva variable 'zp_sim'.
4. Se han identificado duplicados a través de la variable 'item_id', eliminándose estos registros.

5. De la variable 'created_at' se ha extraído, en columnas separadas, el día ('Day'), el mes ('Month') y el año ('Year') de las compras.
6. Algunos códigos postales contienen una cantidad de dígitos diferente de 5. Hemos limpiado los datos de manera que solo se utilicen aquellos valores que únicamente contienen dígitos en el código postal. A aquellos que tenían 4 dígitos se les ha añadido un 0 delante para obtener el código postal correcto (al empezar por 0 el csv los ha leído como números y se ha eliminado este dígito). Los que tenían 5 dígitos se han dejado tal cual estaban. Una vez hecho esto solo se han seleccionado como válidos aquellos códigos postales contenidos entre el número 01000 y 52999, ya que los que se encuentran fuera de este límite no pertenecen al territorio español, y limitaremos la exploración geográfica a este ámbito.
7. Se ha calculado una nueva variable 'precioreal', resultante de calcular sobre el precio del producto el descuento realizado.
8. Se han eliminado los registros con costes base exageradamente altos respecto a su precio de venta, ya que se interpreta que se trata de un error en la codificación. Además, también se han eliminado los registros cuyo coste base no está registrado (es un valor nulo), ya que sin esta información no puede calcularse el beneficio de cada transacción.
9. Se ha calculado una nueva variable 'Precio_calculado', resultante de restar a la variable antes mencionada 'precioreal', el coste base ('base_cost'). Ambas variables se han multiplicado por la cantidad comprada ('qty_ordered') para saber el beneficio/coste de cada transacción.
10. Se ha calculado una nueva variable 'descuento' que contiene los descuentos por rangos en incrementos de 5%. En otra variable 'descuentolabel' hemos guardado la etiqueta correspondiente a cada nivel.
11. En R, llamaremos a este dataframe "tickets_df2".

- **Tabla "products.csv":**

1. Se han formateado correctamente los nombres de aquellas marcas que contenían tildes o caracteres que no se habían codificado bien (por ejemplo, letra ñ).

- **Tabla "products_categories.csv":**

1. Antes de cargar el archivo en ***R*** se han eliminado las comillas, ya que algunas de estas hacían que no se separaran correctamente los valores de las columnas. Esto se ha guardado en un nuevo archivo "products_categories_sincomillas.csv".
2. Se ha igualado el formato en los valores de las categorías, ya que algunas diferían en las mayúsculas/minúsculas o contenían espacios en blanco al comenzar la categoría, lo cual podía ser problemático al identificarse una misma categoría como varias diferentes.
3. Se ha creado una nueva variable 'cat4' que contiene el último nivel de división de la categoría para cada objeto, que puede ser 'cat1', 'cat2', o 'cat3'. Por ejemplo, "Veterinaria" pertenece a la categoría 1, pero no se subdivide en categoría 2 ni categoría 3, así que este sería su último nivel de división y sería su valor también en 'cat4'.

Por último, hemos generado una nueva tabla de datos que contiene la información de todas las tablas relacionadas, que será la que usaremos para llevar a cabo la visualización de los datos.

Para generar esta nueva tabla de datos modificaremos el dataframe "tickets_df2" que hemos generado en ***R***, al cual añadiremos la siguiente información en cada una de las filas:

- Nombre del producto ('productname').
- Marca del producto ('productmarca').
- Descripción del producto ('productdescr').
- Categoría 1 del producto ('productcat1').
- Categoría 2 del producto ('productcat2').
- Categoría 3 del producto ('productcat3').
- Última categoría del producto ('productcat4').

De esta manera, obtenemos la tabla "tickets_dfbigenglishcleandefinitivo.csv", la cual utilizaremos para todos los siguientes análisis y para nuestra plataforma de visualización. Las columnas no necesarias han sido eliminadas para ahorrar espacio y tiempo de procesamiento, dando como resultado la tabla "tickets_data.csv".

3. Análisis y visualización de los datos

A la hora de visualizar los datos, consideramos de suma importancia la accesibilidad por parte de los usuarios interesados. Es por este motivo que nuestra decisión ha sido utilizar la plataforma ***Streamlit***, una aplicación de tipo *open-source* que es ampliamente usada por multitud de empresas de alto prestigio a nivel internacional, como *IBM*, *Johnson & Johnson*, *Tesla* y *Apple* entre otras. La sencillez que supone para el usuario y el hecho de que dependa del lenguaje de programación ***Python***, que es actualmente el lenguaje más usado en la ciencia de datos y el aprendizaje automático, ha hecho que nos hayamos decantado por esta plataforma para presentar nuestro proyecto. El código para el funcionamiento de la aplicación se ejecuta desde la cuenta de ***GitHub*** de los administradores.

3.1. Nivel geográfico y temporal

Dada la elevada sinergia que pueden presentar el nivel de análisis geográfico y el temporal, hemos decidido incluir esta información dentro del mismo apartado. Los datos de estos apartados se han obtenido a través de cuatro tablas principales:

- Tabla con todos los datos relevantes generada anteriormente ("tickets_data.csv"), sin aquellos registros que carecen de código postal español.
- Tabla con los códigos postales y los nombres de las provincias correspondientes.
- Tabla obtenida de la web del INE con la población de cada provincia en 2018.
- Tabla con las coordenadas del área de cada provincia para su representación gráfica.

Estas tablas han sido relacionadas desde ***Python*** de manera que toda esta información se tiene en cuenta simultáneamente a la hora de realizar la representación gráfica. Dependiendo de los filtros de búsqueda (Categoría y subcategoría, mes, año, y dato mostrado), todas las gráficas se actualizan para contener esta misma información. Las gráficas de la evolución temporal siempre muestran todos los meses y años. Las gráficas mostradas en esta sección son:

1. Mapa del territorio español con ventas de la categoría o subcategoría seleccionada y momentos temporales seleccionados.

2. Gráfico de líneas mostrando la evolución temporal de las ventas en la categoría o subcategoría seleccionada para cada provincia, que pueden filtrarse como se desee.

3. Gráfico de líneas mostrando la evolución temporal de las ventas en la categoría o subcategoría seleccionada para el conjunto del territorio español.

Por defecto, se muestran las ventas relativas (€ por cada 100 mil habitantes) de todos los momentos temporales y todas las categorías, pero también pueden verse las ventas totales modificando el dato mostrado.

3.2. Nivel de producto

En esta sección nos centramos principalmente en las categorías de los productos para mostrar los gráficos que hemos considerado más relevantes. Toda la información es extraída de la tabla "tickets_data.csv", con la excepción de los datos relativos a los descuentos, que ha sido generada utilizando ***R*** (con nombre "Ganancias y pérdidas descuentos y categorías.csv"). En esta sección los gráficos que podemos ver son:

1. Un *treemap* de los balances de las ventas producidas en cada categoría y subcategoría.

2. Las marcas más vendidas, según el porcentaje de ventas totales de la categoría y subcategoría seleccionadas.

3. Las marcas que generan más beneficios de las que tienen mayor porcentaje de ventas.

4. Las marcas que generan menos beneficios en la categoría y subcategoría seleccionadas.

5. El balance de las ventas de la categoría y subcategoría seleccionadas dependiendo del descuento ofertado.

3.3. Nivel de cliente

Se ha usado el paquete *arules* de ***R*** para detectar aquellas categorías o grupos de categorías de productos que se han comprado conjuntamente con mayor probabilidad. Para ello se ha utilizado la variable creada 'productcat4', que indica el último nivel de categoría en el que se divide cada elemento categorizado. Las categorías o conjuntos de categorías que más veces se han comprado conjuntamente con una probabilidad del 10% al 60%, siendo esto decisión del usuario, y que se han comprado en al menos en un 1% de ocasiones, pueden observarse en la aplicación. Utilizando funciones de *network*, presentes en la librería *pyvis* de ***Python***, se ha generado la visualización en forma de red que podemos ver en la aplicación. Cuanto más ancha es la conexión entre dos nodos, mayores son las probabilidades de compra conjunta entre esas dos categorías. Cuanto mayor es el número de conexiones que este tiene con otras categorías, más son las categorías de productos con las que se suele comprar conjuntamente esta categoría.

Para finalizar, hemos utilizado *wordclouds2* en ***R*** para generar nubes de palabras con aquellas que se encuentran más presentes en los productos más comprados. Concretamente, para cada categoría de nivel 1, mostramos nubes de palabras del 10% de productos más vendidos dentro de esa categoría.

4. Toma de decisiones basada en los datos (*data-driven decision making)*

Nuestro apartado "Nivel geográfico y temporal" puede permitirnos observar si existen picos de consumo de una categoría de productos concreta en un territorio específico o en una fecha determinada. Esta información puede ayudarnos a la hora de informar respecto a novedades en productos o lanzar promociones de manera más personalizada.

Respecto al apartado "Nivel de producto", podemos ver las ventas originadas por cada categoría, así como las marcas más vendidas de cada categoría, observando también las que mayor y menor beneficio generan, lo cual puede orientarnos a la hora de tomar decisiones respecto a nuestros proveedores. Además, podemos observar las ganancias y pérdidas generadas mediante los descuentos ofertados en cada tipo de producto. Bien es sabido que el ofrecimiento de promociones atractivas a los clientes aumentará la fidelidad de los mismos, pero ser conscientes del coste que nos suponen estas promociones también puede ser interesante.

Por último, en el apartado "Nivel de cliente", podemos ver qué productos o grupos de productos se compran conjuntamente. Además, podemos detectar preferencias de los clientes a través de las nubes de palabras de cada categoría, ya que esto nos permite ver las palabras que han estado más presentes en las descripciones de los productos más vendidos, pudiendo ayudarnos a entender mejor a nuestros compradores. Con esta información seremos capaces de hacer sugerencias más precisas, que faciliten y mejoren la experiencia del cliente con nuestra empresa.

5. Futuras mejoras

Próximas versiones de la aplicación podrían utilizar técnicas de *web scraping* para extraer información sobre reseñas de los clientes. De esta manera, podríamos conocer los puntos negativos en los comentarios peor valorados, mejorando esos aspectos, y los puntos positivos que agradecen los clientes para seguir trabajando en esa dirección buscando un óptimo desarrollo del negocio.