Instituto Tecnológico y de Estudios Superiores de Monterrey

Inteligencia Artificial Avanzada para la Ciencia de Datos II

Evidencia de Reto con Arca Continental

Edgar González Fernández
Mauricio González Soto

Autores:
Cleber Gerardo Pérez Galícia - A01236390
Juan Pablo Bernal Lafarga - A01742342
Jacobo Hirsch Rodríguez - A00829679
Eryk Elizondo González - A01284899

Noviembre 2024

Introducción

Arca Continental es la segunda embotelladora de Coca-Cola más grande de América Latina y una de las más importantes del mundo. La compañía no solo se especializa en la producción y distribución de bebidas gaseosas, sino que también ofrece una amplia gama de productos, como aguas, jugos y bebidas energéticas. Su compromiso con la sostenibilidad, la innovación y la responsabilidad social la ha posicionado como un líder en la industria de bebidas, contribuyendo al desarrollo económico de las regiones en las que opera.

En el competitivo entorno del mercado de bebidas, el análisis predictivo se ha convertido en una herramienta crucial para el éxito de las empresas. La ciencia de datos desempeña un papel fundamental en este proceso, ya que permite a las organizaciones como Arca Continental analizar grandes volúmenes de datos históricos de ventas, así como datos demográficos y de comportamiento del consumidor con el propósito de realizar predicciones de las ventas alrededor de nuevos y novedosos productos de futuro lanzamiento en el mercado.

Problemática

Arca Continental desea un algoritmo de IA que pueda indicar si un producto nuevo tendrá éxito dentro del mercado con base en el comportamiento de compra de sus clientes y/o sus características demográficas.

Objetivos

General

Proveer a Arca Continental un modelo que prediga el desempeño de nuevos productos entre sus clientes históricos, identificando patrones de adopción y construyendo perfiles de clientes óptimos para futuros lanzamientos.

Específicos

- Identificar características clave de consumo que distinguen a los clientes con alta afinidad hacia nuevos productos.
- Construir perfiles óptimos de clientes basados en patrones de consumo y comportamiento, que permitan predecir la probabilidad de adopción de productos nuevos.
- Recomendar estrategias de segmentación y marketing dirigidas, fundamentadas en los perfiles de clientes identificados, para maximizar la aceptación de nuevos productos.

Antecedentes

La inteligencia artificial ha transformado diversos sectores, ofreciendo herramientas para optimizar procesos, mejorar la toma de decisiones y personalizar productos y servicios. En salud, la IA ha impulsado diagnósticos más precisos y tratamientos personalizados, aunque plantea preocupaciones éticas relacionadas con la privacidad de los datos y el sesgo algorítmico. En el sector financiero, ha mejorado la detección de fraudes y la gestión de riesgos, pero también se discuten temas sobre la seguridad de los datos y la equidad en los algoritmos de crédito. En educación, la personalización de la enseñanza ha crecido, aunque se plantean preocupaciones sobre el uso de datos sensibles de los estudiantes. En todos los sectores, el cumplimiento de normativas como el GDPR y la adherencia a principios éticos, como la transparencia y la equidad, son cruciales.

En el contexto del proyecto con Arca Continental, la inteligencia artificial no solo apoya la predicción del éxito de ventas, sino que también permite una segmentación de mercados más precisa, ayudando a identificar a los clientes con alta probabilidad de adoptar nuevos productos. Al analizar datos demográficos y de consumo históricos, se emplean herramientas avanzadas como el análisis de componentes principales (PCA) para reducir la dimensionalidad y detectar patrones en grandes volúmenes de datos. También se utiliza la similitud de coseno (Cosine similarity) y embeddings para evaluar y comparar perfiles de clientes en función de características específicas, lo cual facilita la identificación de segmentos relevantes de manera más precisa. La normalización de datos asegura que todas las variables tengan la misma escala, mejorando la comparabilidad y la efectividad de los análisis.

Estos métodos avanzados de IA permiten a Arca Continental optimizar su oferta de productos y diseñar estrategias de marketing personalizadas, pero también presentan desafíos éticos y normativos. Es crucial asegurar la privacidad de los clientes y evitar sesgos en los análisis que puedan resultar en segmentaciones injustas o discriminatorias. Además, el cumplimiento de leyes de protección de datos personales, como la Ley Federal de Protección de Datos Personales en México y regulaciones internacionales, es esencial para garantizar prácticas éticas y transparentes. Así, el uso de IA en este proyecto busca no solo mejorar la satisfacción del cliente y la eficiencia operativa, sino también mantener una reputación ética y responsable.

Recursos

Para el desarrollo de la solución propuesta, se eligió el lenguaje Python como herramienta principal, debido a su robustez en el manejo de datos y su amplio ecosis-

tema de bibliotecas para ciencia de datos e inteligencia artificial. La implementación se llevó a cabo de manera local en Visual Studio Code, cumpliendo con los términos establecidos por el acuerdo de confidencialidad (NDA) firmado con Arca Continental.

Las bibliotecas de Python utilizadas para la solución propuesta fueron:

- Pandas: Para la manipulación y análisis de datos estructurados, incluyendo la transformación de archivos CSV en DataFrames y operaciones avanzadas de agrupación y filtrado.
- Numpy: Utilizada en operaciones numéricas avanzadas sobre arreglos multidimensionales.
- Scikit-Learn: Aplicada para la normalización de los datos con MinMaxScaler, la codificación de variables categóricas con One-Hot Encoding, técnicas de reducción de dimensionalidad avanzadas como Análisis de Componentes Principales (PCA) y cálculo de similitud con Cosine Similarity.

Además, se utilizaron técnicas de procesamiento de lenguaje natural como Word2Vec para calcular similitudes semánticas entre columnas categóricas de los datasets y CountVectorizer para tokenizar en la similitud de Jaccard que se explicará más adelante.

Los datos utilizados provienen de dos de tres archivos .CSV proporcionados por Arca Continental, que contienen registros de ventas de 2019 a 2022, descripciones de productos y un diccionario con características demográficas de sus clientes. Siendo los primeros dos los utilizados para concretar el modelo, y que describiremos con más detalle en el apartado de metodología.

Se usaron referencias clave como la guía de uso de bibliotecas mencionadas, más específicamente la biblioteca de Scikit, y documentación de Word2Vec.

Metodología

Extracción de datos

Para la solución de la problemática nos enfocaremos en dos fuentes principales: el historial de ventas y las características de los productos. Para su análisis se convirtieron los archivos en formato CSV a un dataframe de Pandas. Como convencion se estará refiriéndo al daraset extraído de productos como "productos" y al dataset extraído de ventas como "ventas". A continuación se detalla una descripción de las característicass generales que se encontraron para los dos dataframes que vamos a utilizar:

Ventas proporciona un registro histórico de transacciones, donde cada entrada corresponde a la venta de un producto específico a un cliente en un mes dado. Este archivo contiene los siguientes campos relevantes:

- ***CustomerId*** : identificador único del cliente
- ***material*** : código del producto vendido
- ***calmonth*** : fecha en la que se vendió el producto en formato yyyymm y *unibox* : cantidad de unidades vendidas en formato decimal

Productos proporciona información detallada sobre las características de cada producto disponible para la venta. Cada entrada corresponde a un producto específico y contiene los siguientes campos relevantes:

- ***Material:*** Código único que identifica el producto.
- ***Materialdesc***: Descripción detallada del producto, que incluye marca, sabor, tamaño y presentación.
- ***ProductosPorEmpaque***: Cantidad de unidades contenidas en el empaque del producto).
- ***BrandPresRet***: Combinación de la marca y el tipo de presentación (retornable o no retornable) del producto.
- ***ProdKey***: Llave que agrupa los productos en categorías amplias.
- ***Brand***: Marca del producto, que identifica el fabricante o marca comercial.
- ***Presentation***: Tipo de presentación y volumen del producto.
- ***MLSize***: Tamaño en mililitros del producto.
- ***Returnability***: Indicador de retornabilidad del envase.
- ***Pack***: Presentación general del tamaño del producto.
- ***Size***: Tamaño del producto, simplificado en distintas formas de presentación.
- ***Flavor***: Sabor del producto.
- ***Container***: Tipo de envase del producto.
- ***Ncb***: Indicador de negocio para productos no carbonatados o carbonatados.
- ***ProductType***: Tipo de producto.
- ***ProductCategory***: Categoría general del producto.
- ***SegAg***: Segmento agregado del producto, que lo clasifica en categorías de mercado amplias.
- ***SegDet***: Segmento detallado del producto, que permite subcategorizar dentro de SegAg.

- ***GlobalCategory***: Clasificación global del producto, indicando la categoría general a la que pertenece.
- ***GlobalSubcategory***: Subcategoría global del producto, una clasificación más específica dentro de GlobalCategory.
- ***BrandGrouper***: Agrupador de marca que clasifica productos de marcas similares en una categoría.
- ***GlobalFlavor***: Sabor global del producto, que permite estandarizar la clasificación de sabores.

La cantidad de columnas y de filas para los datasets es: ventas tiene 2,347,110 filas y 4 columnas, mientras que productos tiene 793 filas y 22 columnas. La columna material sirve como llave foránea para conectar con ventas, por lo que se comprobó que para cada fila que hay en productos el valor de material es único.

En ventas hay 3 variables de tipo entero, dos de las cuales fungen como llave compuesta siendo CustomerId y material sus respectivas claves. la variable restante es de tipo decimal.

En productos hay 4 columnas de tipo entero y 18 columnas de tipo object, de antemano sabemos que las que son de tipo object pueden ser convertidas a cadenas de texto, proceso que se hará más adelante.

Limpieza de datos

Como la mayoría de las variables son categóricas, y las categorías que se presentan por variable son nominales, se observó que las categorías entre variables eran repetitivas, presentándose en formato de tipo "string". Dicho eso, se formó la hipótesis de que algunas de las variables podrían reducirse si podríamos comprobar la similitud que existe entre columnas.

Utilizamos dos técnicas de procesamiento de lenguaje natural para comprobar dos diferentes tipos de similitud que se pueden encontrar entre "corpus", para la similitud semántica se usaron vector embeddings y para la similitud exacta de palabras se utilizo la similitud de jaccard para el vocabulario entre dos columnas. A continuación se describe de manera detallada el proceso.

0.0.1 Vector Embeddings

El primer paso para implementar este filtro de similitud es seleccionar las columnas que son categóricas, convertimos cada elemento de las columnas seleccionadas a cadenas de texto, llenando los valores faltantes con cadenas de texto vacías, El siguiente paso es juntar cada valor de cada columna para tokenizar el texto que hay en cada fila, la tokenización implementada fue por palabra.

Como paso siguiente es el entrenamiento del modelo word2vec, este modelo es una técnica de aprendizaje automático diseñada para convertir palabras en embeddings, estos embeddings capturan el significado y la relación semántica en un espacio de alta dimensión. Se especificó que el modelo tendrá por vector 100 dimensiones, con una ventana de contexto de 5 palabras, el modelo considera como mínimo una palabra.

Para guardar el resultado necesitamos un espacio donde almacenarlo, por lo que se creó una matriz de ceros de dimensión 18x18, que es la cantidad de columnas categóricas en la tabla de productos.

Para calcular las similitudes entre columnas categóricas iteramos sobre pares de columnas mediante ciclos anidados, evitando comparar una columna consigo misma o duplicar evaluaciones ya realizadas. Los valores de cada columna se transforman en cadenas de texto y, para cada celda, el texto se tokeniza en palabras individuales. Luego, se extrae el vector (embedding) asociado a cada palabra del modelo Word2Vec. Si una celda no contiene palabras válidas se asigna un vector nulo. Posteriormente, se calcula el promedio de los vectores obtenidos por celda, generando un vector representativo para cada fila de la columna. Esto produce una matriz implícita donde cada fila es el vector promedio de una celda, y su tamaño es n×d, siendo n el número de filas de la columna y d la dimensión de los embeddings (en este caso 100).

Una vez generadas estas matrices para el par de columnas en evaluación, se calcula la similitud coseno entre cada par de vectores correspondientes (misma fila) de ambas matrices, formando una matriz de similitudes intermedias de tamaño n×n. La similitud coseno entre los dos vectores se dfine como

$$\text{SimCos}(u, v) = \frac{u \cdot v}{\|\mathbf{u}\|\|\mathbf{v}\|}$$

donde:

- **u**: Es el vector que representa las características del contenido textual de una celda en la primera columna (por ejemplo, *Material_desc*).
- **v**: Es el vector que representa las características del contenido textual de la celda correspondiente (misma fila) en la segunda columna (por ejemplo, *BrandPresRet*).

Esta matriz captura todas las comparaciones posibles entre los valores de las dos columnas en cada fila. De esta matriz, se extraen los valores de la diagonal principal, que representan las similitudes entre celdas alineadas en ambas columnas (misma fila). El promedio de estos valores se calcula y se almacena en una matriz global de similitudes que resume la similitud entre pares de columnas categóricas. Finalmente, esta matriz global se convierte en un DataFrame para facilitar su visualización y análisis.

Para identificar las columnas categóricas con un nivel de similitud significativo, se define un umbral de similitud como referencia, en este caso establecido en 0.95. A partir de la matriz de similitudes previamente calculada,

se recorre cada par de columnas categóricas mediante ciclos anidados, considerando únicamente las combinaciones únicas (sin repetir pares ni incluir comparaciones de una columna consigo misma). Para cada par, se verifica si la similitud calculada excede el umbral establecido.

Cuando la similitud entre dos columnas supera el umbral, se almacena esta relación en un diccionario estructurado. Cada clave del diccionario corresponde al nombre de una columna categórica, y su valor asociado es una lista que contiene tuplas. Cada tupla registra el nombre de la columna similar y el valor de similitud entre ambas. De este modo, el diccionario organiza las relaciones de similitud entre columnas categóricas de forma jerárquica, facilitando la identificación de grupos de columnas con contenido semántico altamente relacionado.

0.0.2 Similitud de Jaccard

Para medir la similitud entre dos columnas categóricas en términos de sus vocabularios, se utiliza el cálculo de la similitud de Jaccard. El proceso comienza uniendo los valores de cada columna en una sola cadena de texto, asegurándose de reemplazar valores faltantes por cadenas vacías para evitar errores. Esto permite tratar el contenido de cada columna como un conjunto único de palabras.

A continuación, se utiliza un CountVectorizer configurado en modo binario para convertir las palabras de cada columna en vectores de presencia o ausencia. Cada posición del vector indica si una palabra específica está presente (1) o ausente (0) en el texto de la columna correspondiente. Esto produce dos vectores binarios, uno para cada columna.

Con estos vectores, se calcula manualmente la similitud de Jaccard, que mide la proporción entre la intersección y la unión de los conjuntos de palabras representados por los vectores.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

donde:

- $|A \cap B|$ es el conjunto de palabras comunes entre ambas columnas.
- $|A \cup B|$ es el conjunto de todas las palabras únicas que aparecen en cualquiera de las dos columnas.

La intersección se calcula como la suma de las coincidencias en ambas columnas (valores mínimos en cada posición del vector), mientras que la unión se calcula como la suma de las posiciones donde al menos una columna tiene un valor (valores máximos en cada posición). Finalmente, la similitud de Jaccard se obtiene dividiendo la intersección entre la unión.

0.0.3 Selección y eliminación de columnas redundantes

Tras realizar el análisis utilizando ambos enfoques de similitud, se procedió a identificar y seleccionar las columnas categóricas que se consideraron redundantes. Para la similitud coseno, se estableció un umbral de 0.95, lo que permitió identificar como redundantes aquellas columnas que presentaban una relación semántica muy alta. En cuanto a la similitud de Jaccard, se aplicaron diferentes umbrales de forma iterativa, comenzando con un valor inicial de 0.8, el cual fue reducido progresivamente hasta 0.5. El umbral inicial de 0.8 ayudó a identificar columnas con vocabularios altamente coincidentes, mientras que los umbrales más bajos, como 0.6 y 0.5, relajaron la restricción de coincidencia exacta en vocabulario, lo que permitió capturar redundancias adicionales entre columnas cuyos valores, aunque no fueran idénticos, presentaban un solapamiento considerable en términos de palabras.

A través de este enfoque combinado, se identificaron las columnas con similitudes significativas según los criterios mencionados. Finalmente, las columnas BrandGrouper, Pack, ProductType, SegAg y SegDet fueron marcadas para eliminación. De tal manera que redujimos la base de datos a la menor cantidad de columnas que nos ayudan a diferenciar todos los productos y que explican la mayor información del mismo.

Transformacion de datos

0.0.4 Base de datos de Productos

Dada la naturaleza categórica de todas las features que definen un producto, las variables con 2 categorías se transformaron en variables binarias manualmente, mientras que a las variables multiclase se les aplicó la función One-Hot Encoding de Scikit Learn, la cual convierte variables categóricas en formato numérico, creando una nueva columna para cada posible categoría y asignando un valor binario (0 o 1) para indicar si una instancia pertenece o no a esa categoría. Con la transformación las variables categóricas de los productos se eliminaron las columnas originales de los productos, y quedamos con 167 columnas de características que definen cada producto. A continuación, se escalaron todas las variables transformadas (excepto Material) a un rango uniforme utilizando la técnica de Min-Max Scaling. Esto asegura que todas las variables, tanto las numéricas como las codificadas con One-Hot Encoding, estén en la misma escala, permitiendo un tratamiento equitativo de todas las características.

Finalmente, reducimos la dimensionalidad del conjunto de datos escalado utilizando Análisis de Componentes Principales (PCA) con 50 componentes principales, reteniendo el 94.45 de la variabilidad que se tenía con las 167 características. Esto simplifica el conjunto de datos para mejorar la eficiencia y el rendimiento del

modelo, preservando la mayor cantidad posible de información relevante. Adicionalmente, se crearon dos nuevos DataFrames con estas componentes principales, uno incluyendo los identificadores de producto basado en la columna Material del conjunto de datos original para mantener la trazabilidad de cada registro y otro sin los identificadores que funciona como una matriz de características en la que cada fila es un vector único e irrepetible que define un producto.

0.0.5 Base de datos de Ventas

Después de preparar los datos del archivo de productos, se llevó a cabo la preparación del archivo de ventas. El proceso comenzó con la transformación de la variable de fecha (Date Time), formateándola en términos de día, mes y año para posteriormente realizar una separación por mes. Además, se realizó un renombramiento de columnas clave para una mejor legibilidad y organización, destacando la columna relacionada con los galones vendidos.

Para resaltar las ventas mensuales de cada cliente sobre los galones comprados de un producto específico, se llevó a cabo una transposición de los meses, convirtiéndolos en columnas. Esto proporcionó un panorama donde cada columna representa un mes y su valor corresponde a la cantidad de galones comprados por un cliente para ese producto específico. Si en algún mes no hubo compra, el valor de esa columna se dejó en 0.

Se creó una nueva columna llamada "proporción" para calcular la proporción de los galones comprados de productos por cliente con respecto al total de meses. Esto ofrece un indicador sobre cómo un producto contribuye en comparación con otras compras realizadas por el mismo cliente. También se registraron las ventas unitarias mensuales de cada producto, tomando en cuenta la proporción en relación con las ventas de otros productos.

Finalmente, obtuvimos un conjunto de datos que detallan de manera profunda las ventas mensuales por cliente y producto, con indicadores clave como la frecuencia de ventas y la proporción de ventas. Facilitando el entendimiento del comportamiento de compra de los clientes y ayuda en el análisis y toma de decisiones estratégicas sobre productos y su rendimiento en diferentes períodos.

Producto Nuevo

Para introducir un nuevo producto, se crea un vector de características igual al de la base de datos de los productos, con todas las características originales y con la única condición de que sea diferente a todos los productos existentes por al menos una característica. Por ejemplo:

$$Existente = [Fanta, Naranja, 600ml, Refresco]$$
$$Nuevo = [Fanta, Limón, 600ml, Refresco]$$

Modelo

0.0.6 Transformación del nuevo producto

Una vez definido un nuevo producto, se prepara para que sea consistente con el formato y las transformaciones aplicadas al conjunto de datos de productos original. Esto incluye la selección de características relevantes, la transformación de presentaciones, la codificación de las variables categóricas, la normalización y la proyección mediante componentes principales.

El resultado final es un conjunto de datos listo para análisis o modelado en el mismo espacio de características reducido, facilitando comparaciones y predicciones con los productos ya procesados.

0.0.7 Similitud de los productos

Se utilizo el indicador de la similitud coseno, el cual es útil en este caso porque permite medir la relación de similitud entre productos en función de sus características numéricas, que han sido transformadas previamente con PCA. Al calcular la similitud entre el nuevo producto y los productos existentes se puede sugerir productos similares a los que un cliente ya ha comprado, organizar productos similares en categorías o grupos basados en sus características y, lo más interesante para el modelo, evaluar qué tan similares son los productos con base en sus características.

Es de esta forma que a través de la similitud coseno entre la matriz de características y el vector de características del nuevo producto se obtiene un vector de similitudes. El cual se agrega a una nueva columna del DataFrame de ventas para identificar productos que, aunque no sean exactamente los mismos, podrían dar un ideal del desempeño esperado del nuevo producto debido a sus características compartidas.

0.0.8 Indiciador final

Para calcular un indicador final que evalúe la compatibilidad entre productos y clientes, se combinaron tres elementos clave: la frecuencia de ventas de los productos, la proporción de ventas y la similitud coseno de los productos con el producto nuevo. Estos tres indicadores son ponderados según su relevancia para el modelo, utilizando pesos específicos: un 20 para la frecuencia de ventas, un 20 para la proporción de ventas y un 60 para la similitud de productos.

Los pasos para calcular el indicador final son los siguientes:

- SaleFrequency (SF): Este indicador refleja cuántos meses en los que el cliente ha comprado un producto.

La mayor frecuencia de compra implica una mayor compatibilidad del cliente con ese producto.

- SaleProportion (SP): Calcula la proporción de galones comprados de un producto específico en relación con las compras totales de ese cliente. Esto ayuda a identificar qué tan relevante es un producto para un cliente en particular, dado su comportamiento de compra en comparación con otros productos.
- ProductSimilarity (PS): Mide qué tan similar es un producto en función de sus características (como el tipo de presentación, sabor, etc.) en comparación con un nuevo producto. Esta medida es crucial para encontrar productos similares que puedan tener un desempeño comparable al nuevo producto.

El cálculo de la compatibilidad final se hace de la siguiente manera:

$$compatability = \alpha \times SF + \beta \times SP + \gamma \times PS$$

Donde:

- $\alpha = 0.2$ (peso de la frecuencia de ventas),
- $\beta = 0.2$ (peso de la proporción de ventas),
- $\gamma = 0.6$ (peso de la similitud de productos).

Este valor de compatibilidad es luego utilizado para ordenar los productos, de modo que los productos más compatibles con cada cliente estén al principio de la lista, facilitando la identificación de potenciales clientes para vender el nuevo producto.

Por último, este indicador final es fundamental para entender cómo un nuevo producto se comportará en el mercado en función de las características de productos similares y los patrones de compra de los clientes.

0.0.9 Evaluación

Para evaluar el desempeño del modelo, se utilizó un enfoque basado en la precisión, que mide qué tan bien identifica a los clientes más compatibles con un producto específico.

Se definió un puntaje que combina la frecuencia de ventas de un producto a un cliente y la proporción que representan estas ventas respecto al total del cliente. Estos indicadores fueron ponderados de manera equitativa, con un peso de 0.5 para cada uno. El puntaje final refleja la afinidad de cada cliente con un producto.

Después, los clientes fueron ordenados según su puntaje de éxito calculado, obteniendo una lista priorizada que predice cuáles serían los clientes más compatibles con el producto analizado.

Luego, se comparó la lista de clientes predicha con los datos reales para medir la precisión en diferentes niveles: los top 1, 5, 10, 20, 50 y 100 clientes más compatibles según el modelo. Esto permite evaluar qué tan bien el modelo identifica a los clientes que efectivamente interactuaron con el producto.

Para cada top de evaluación (1, 5, 10, 20, 50 y 100), se calculó la proporción de coincidencias entre los clientes predichos por el modelo y los clientes reales que adquirieron el producto.

Resultados

Una vez realizada la evaluación, es posible comparar la precisión del modelo. Por ejemplo, si en el top 5 da un resultado de 20% significa que el modelo, en su ordenamiento del éxito calculado de los clientes, pudo predecir que 1 de los top 5 realmente estuviera en el top 5 al comparar con los datos reales de los clientes. Esto nos permite ajustar los hiperparámetros del modelo para obtener una mejor precisión. Lo que resulta en una lista ordenada de todos los clientes con su mejor producto y el indicador calculado de compatibilidad. Este listado ayuda a asignar una prioridad a los clientes de forma que los clientes más arriba en la lista poseen una mejor compatibilidad con el producto introducido y tendrán mejores resultados de adoptar el producto nuevo en un acercamiento, esto en contraste con los que se encuentran más debajo en el listado. Esto nos ayuda a optimizar la asignación de recursos de marketing enfocado a clientes mejor receptivos.

Conclusiones

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla nec malesuada libero, et accumsan orci. Etiam tempus, eros id rutrum rhoncus, tortor nisi ultrices lacus, eu rutrum sem orci sed velit. In rhoncus, tellus eu suscipit scelerisque, urna libero pharetra ante, sit amet porta tortor lacus ut lacus. Aliquam fringilla nec nibh eleifend suscipit. Etiam vulputate elit a pellentesque accumsan. Nulla leo velit, viverra ut odio sit amet, congue vehicula enim. Sed nec volutpat ipsum, sed venenatis quam. Sed iaculis gravida finibus. Fusce laoreet convallis felis, sed fringilla tellus auctor eu.

Discusiones

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla nec malesuada libero, et accumsan orci. Etiam tempus, eros id rutrum rhoncus, tortor nisi ultrices lacus, eu rutrum sem orci sed velit. In rhoncus, tellus eu suscipit scelerisque, urna libero pharetra ante, sit amet porta tortor lacus ut lacus. Aliquam fringilla nec nibh eleifend suscipit. Etiam vulputate elit a pellentesque accumsan. Nulla leo velit, viverra ut odio sit amet, congue vehicula enim. Sed nec volutpat ipsum, sed venenatis

quam. Sed iaculis gravida finibus. Fusce laoreet convallis felis, sed fringilla tellus auctor eu.

References

[1] P. Virtanen et al., "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261-272, 2020.

https://www.arcacontal.com/