Data Analysis

“Breve análisis de *Olist* y del Embudo de Ventas”

Proyecto Final

Alumno: Céspedes Jaén Abdiel

Expertos:

- **SQL & Mongo DB:** Ramírez Andrés Carlos
- **R:** Franco Jenner
- **Python (1):** Arias José Ramón
- **Python (2):** Peña Olivares Omar

Grupo: data - analysis - gdl - 20 - 06

Fecha de entrega: Jueves 25 de marzo de 2021

Breve análisis de *Olist* y del Embudo de Ventas

Índice

I. Resumen

En el presente trabajo se pretende abordar, mediante un breve análisis con respecto a un conjunto de bases de datos, el comportamiento histórico de una ecommerce brasileña y de su embudo de ventas en un corto horizonte de tiempo, pretendiendo, de esta manera, fungir a su vez como evidencia para demostrar el conjunto de habilidades obtenidas a lo largo del curso de *"Data Analysis"* de BEDU.

El presente es un documento compuesto, en primer lugar, por una introducción que explica al lector desde la justificación hasta el planteamiento del problema; seguido por un marco teórico en el que se profundiz en la temática a la cual pertenecen las BBDD para facilitar el entendimiento; después, se cuenta con un desarrollo que muestra paso a paso el proceso mediante el cual se da respuesta a las preguntas que se deslindan tanto de la problemática principal como de los objetivos; una conclusión que pretende reflejar tanto el cierre del proyecto como un resumen de los aprendizajes adquiridos a lo largo del curso y, finalmente, un glosario en el apartado de anexo con conceptos facilitar el entendimiento.

1.1. Breve aclaración

Si bien el lenguaje de programación usado para el análisis de las BBDD fue Python, este es un proyecto que el autor - el alumno - lleva desarrollando desde el primer módulo del curso, por lo que muchas de las peticiones/solicitudes de información que se hicieron a las BBDD, se hicieron también en SQL y en Mongodb. Asimismo, debo aclarar el hecho de que, para el módulo de R, se exploró la posibilidad de cambiar el proyecto y, por lo tanto, en dicho módulo se realizó un análisis de un conjunto de BBDD de los repositorios de BEDU con respecto al comportamiento del Covid. Todo lo antes mencionado cuenta con documentos de evidencia cargados en el repositorio del alumno, cuya liga es la siguiente:

https://github.com/AbdielCJ/checkpoint1.git

II. Introducción

Brazilian E-Commerce Public Dataset by Olist, descrito mediante las palabras que corresponden a la descripción de las BBDD en Kaggle, es "un conjunto público de datos de una ecommerce brasileña de órdenes hechas a *Olist Store*" (Olist, 2019). Así pues, como si de una oportuna coincidencia se tratara, y debido a la necesidad de contar no con una, sino con varias BBDD relacionadas entre sí para cursar el primer módulo del curso, el experto Andrés C. Ramírez me sugirió que utilizara dicho conjunto de datos. Lo anterior lo describo como una oportuna coincidencia porque, a pesar de no haber estudiado algo relacionado con el *marketing* o las ventas, actualmente me encuentro laborando en el departamento de *marketing* de

una empresa. El tema del embudo de ventas se ha vuelto esencial para mí entenderlo desde que cursé mis prácticas profesionales para contar con el sustento teórico necesario a partir del cual pudiera yo contribuir al incremento de ventas para la empresa en la que actualmente laboro.

Si bien, debo aclarar, hay considerables diferencias entre el embudo de ventas analizado a lo largo del presente proyecto y el embudo de ventas de la empresa para la que actualmente trabajo, fueron los conceptos teóricos que describen la naturaleza de los procesos de ventas lo que despertó mi curiosidad al respecto. Sin profundizar mucho al respecto, la principal diferencia es el modelo de negocio, siendo el de *Olist Store* es B2C, mientras que el de la empresa para que trabajo es B2B, es decir, que el uno está enfocado en vender a un cliente directo, mientras que el otro tiene por cliente a otras empresas, respectivamente.

Así pues, como no pude establecer un símil directo entre mi labor actual y este conjunto de BBDD, decidí hallar una justificación diferente para el análisis, la cual, menciono a continuación.

2.1. Planteamiento del problema

Poniéndome en el lugar de una persona que quisiera emprender un negocio y recurriera a la infraestructura de *Olist Store* para ofertar sus productos, la problemática a resolver consiste en querer conocer el método a través del cual se pueda tanto adquirir como interpretar información clave de las BBDD para poder tomar las mejores decisiones en cuanto a posicionamiento de marca se refiere.

2.2. Objetivos

2.2.1. General

Presentar un marco teórico y el método a través del cual se aborde el análisis de BBDD de una ecommerce y su embudo de ventas para justificar la toma de decisiones.

2.2.2. Particulares

- Presentar paso a paso el método realizado para el análisis de las BBDD.
- Generar preguntas clave de investigación para orientar a la búsqueda de áreas de oportunidad para poder incursionar en el posicionamiento de un nuevo negocio.
- Proporcionar respuestas a las preguntas previamente señaladas para justificar la toma de decisiones.

III. Marco teórico

En el presente apartado se detallan una serie de temáticas relacionadas con el carácter mercadológico que tiene el proyecto debido a su enfoque en relación con el embudo de ventas y la infraestructura de la ecommerce por medio de la cual los vendedores ofertan sus productos. Es necesario ahondar en ciertos temas para entender los datos que se reflejan en las BBDD analizadas más adelante.

3.1. Marketing

De acuerdo con Kerin et al (2004), quienes a su vez citan a la *American Marketing Association*, definen este concepto como "... el proceso de planear y ejecutar la concepción, fijación de precios, promoción y distribución de ideas, bienes y servicios para crear intercambios que satisfagan los objetivos individuales y organizacionales". Así pues, y de acuerdo con los autores de *"Marketing",* la razón de ser de este proceso se basa en el intercambio que surge a partir de la relación de un comprador y un vendedor.

Gracias a las tecnologías de la información y la comunicación (TIC's), se ha llegado a la posibilidad de crear diversos canales tanto de distribución como de prospección, por medio de los cuales esta relación antes descrita basada en el intercambio de bienes y/o servicios a incrementado su alcance e impacto en la sociedad. Diversas estrategias han surgido a raíz de esto y con el propósito de optimizar la rentabilidad de las empresas, por lo que ahora nos compete profundizar en dos de los principales enfoques, comparar sus ventajas y desventajas para ir descifrando el propósito del embudo de ventas.

3.1.1. Outbound marketing

Por medio de una infografía creada por DElia (2014) podemos ver que se le conoce como "Viejo *Marketing*", definiéndolo ella como "cualquier tipo de *marketing* que interrumpe/obliga los productos o servicios a los clientes". Por su parte, Bel (2020), nos ofrece su definición, mencionando que son todos aquellos métodos mercadológicos que consisten en captar la atención de los clientes, pero a través de métodos directos. Algo muy importante a destacar es la inclusión de la palabra "unidireccional" en la definición de Bel (2020), dando a entender que la información sólo fluye del que oferta al que se espera que consuma, nunca al revés, por lo que, para conocer VOC se deben de recurrir a otros medios ajenos al método de oferta.

3.1.2. Inbound Marketing

Samsing (2018) nos ofrece la siguiente definición: "es una estrategia que se basa en atraer clientes con contenido útil, relevante y agregando valor en cada una de las etapas del recorrido del comprador". La parte que menciona lo de agregar valor es muy importante porque, si bien tanto en el *inbound* como en el *outbound marketing*

hay procesos, se puede apreciar que aquí la comunicación no es unidireccional porque, como principales diferencias, podemos reconocer que, en primer lugar, se busca atraer al potencial cliente en vez de llegar a él por medio de un método directo y, por otro lado, se le agrega valor en cada etapa, existiendo entonces un flujo de comunicación bilateral, una relación más definida entre comprador y vendedor.

El *inbound marketing* es el enfoque principal del embudo de ventas. Como se verá a detalle más adelante, el embudo es precisamente un proceso a través de cuyas etapas, además de agregar valor al potencial cliente, se le guía para hacerlo propenso a convertirse en un cliente.

3.1.3. Inbound vs Outbound

Si bien es fundamental hacer una comparativa entre una estrategia y la otra, es importante mencionar desde ahora que estas no son mutuamente excluyentes, por lo que una empresa bien podría implementar ambas y, de esa manera, generar resultados positivos de una combinación fructífera. Tal es lo que reconoce un artículo de Inboundcycle (2020), en el cual reconoce sus diferencias, pero también la posibilidad de emplear ambas estrategias a la vez.

Por su parte, si bien DElia (2014) no niega que puedan ser aplicadas al mismo tiempo, enfatiza más en los beneficios del *inbound* sobre el *outbound marketing,* entre los cuales están:

- La comunicación es interactiva y bilateral.
- Los clientes son atraídos, no forzados.
- Los mercadólogos, dado el canal de comunicación bilateral, son capaces de aportar más valor y apoyar a la conversión de potenciales clientes a clientes.

Es importante destacar que el *inbound* saca el mejor provecho, pues se basa en el potencial de las TIC ‘s como por medio de campañas en redes sociales, por medio de las cuales lanza contenido de valor que atrae a potenciales clientes a interactuar con la empresa.

3.2. El producto

Gracias a Thompson (2006) y a su artículo *“Concepto de Producto”*, el cual aborda este concepto desde un punto de vista mercadológico, sabemos que el producto no está limitado meramente a objetos, sino que también incluye a personas, servicios, eventos, experiencias, entre otros. Además, Kotler et al (1998) con *“Fundamentos de mercadotecnia”,* menciona que la definición del producto está en función de sus “atributos, especificaciones o condiciones”.

En el caso que nos atañe a lo largo de este proyecto, el lector será testigo de que, debido a los tipos de negocios que recurren a *Olist,* los productos sí se limitan en esta ocasión a objetos, pero no podemos pasar por alto, por un lado, la gran cantidad de artículos ofertados y que las reseñas de estos - la voz del cliente (VOC) - están a su vez subyugados por un servicio, aquel en el que recurren a los socios de servicios logísticos de *Olist* a manera de subcontratación (*outsourcing*) para hacer entrega del producto, como se verá más adelante.

3.3. El mercado

Volviendo nuevamente con Kerin et al (2004), sabemos gracias a ellos que el mercado consiste en aquel público que consta de tres condiciones:

- Que sean personas.
- Que tengan deseo por el producto/servicio.
- Que tengan la capacidad para adquirir tal producto/servicio.

Debido a la ya mencionada potencialización que proveen las TIC 's, aquí manifestada por medio de una ecommerce, se podrá comprobar que muchos mercados convergen en un mismo embudo. La segmentación es de los pasos más importantes para las empresas ya que solo así pueden identificar a su público o a sus públicos objetivos; de nada sirve ofrecer el producto o servicio si se le ofrece a las personas equivocadas. He aquí que resalta una de las principales áreas en las que el análisis de datos puede ser de gran ayuda, pues, mediante a técnicas de adquisición y visualización de información, se pueden conocer estos mercados y la cantidad de personas que han interactuado con ellos.

3.4. El embudo de ventas

Llamado *sales funnel* en inglés, un embudo de ventas, de acuerdo con Zhel (2020), es un concepto que engloba todo lo correspondiente al proceso que atraviesa un cliente antes de convertirse en uno. La idea es que, a medida en que una persona, un usuario o un interesado atraviesa las etapas del embudo, hay una mayor probabilidad de que se convierta en un cliente, ya que, como se mencionó antes, al formar parte de las estrategias de *inbound marketing,* se le va añadiendo valor a la persona a lo largo del proceso, obteniendo de manera recíproca, un mayor compromiso por parte del prospecto a corresponder a los siguientes pasos de conversión. A medida de que el *lead* va avanzando a lo largo del proceso del embudo de ventas, decimos que se va “calentando”.

Ahora bien, es importante a su vez rescatar dos conceptos importantes antes mencionados. El primero es el de prospecto, traducido al inglés como *lead,* y que, de acuerdo con Pérez (2019), “se trata de un potencial cliente de tu marca que demostró interés en consumir tu producto o servicio”. De este concepto se

desprende a su vez otros dos igualmente importantes y que se verán más a profundidad cuando se expliquen las etapas del embudo, los cuales son *marketing qualified lead (MQL)* y *Sales Qualified Lead (SQL* - importante no confundir este concepto con el lenguaje de programación orientado a gestionar bases de datos relacionales*).*

El otro concepto importante a rescatar es el de conversión. Una conversión es toda aquella acción que nosotros como vendedores determinamos y que, una vez que el *lead* la realiza, decimos que el prospecto se ha "convertido". La conversión más importante que podemos reconocer es la de haber comprado nuestro producto o servicio, sin embargo, la palabra conversión no se limita solo a ello, pues, como ya se mencionó, es toda aquella acción que determinemos y, para poder medir el viaje del *lead* a lo largo del embudo de ventas, es importante establecer una acción de conversión por cada una de las etapas de dicho embudo. Entre otros, algunos ejemplos de conversión pueden ser:

- Descargar un *Lead Magnet.*
- Llenar con nuestros datos un formulario.
- Agendar una reunión con el vendedor.
- Dar clic en un *CTA.*

3.4.1. Beneficios

Si bien la idea del embudo de ventas forma parte de la estrategia del *inbound marketing,* como ya mencionó, es posible que la empresa implemente, a su vez, una estrategia de *outbound* con el propósito de "dirigir" a los *leads* a la etapa inicial del embudo y, entonces puedan pasar a ser "guiados" a lo largo de sus etapas. Podría decirse que sería como darles un "empujoncito" a aquellos potenciales clientes que aún no conocen nuestra marca.

Enfocándonos específicamente en el embudo de ventas, Máñez (2020) enlista tres beneficios del embudo:

- **Generar confianza:** Como ya se mencionó, este beneficio se alinea precisamente con el propósito de una estrategia de *inbound marketing.*
- **Transformar desconocidos en clientes:** Tal y como se mencionaba en el párrafo introductorio de este subtema, a veces hay personas que sí están interesados en lo que nuestra marca pueda ofrecerles, pero puede que no nos conozcan todavía, así pues, gracias a las actividades de conversión del embudo y de su propuesta de agregar valor, se genera esta dinámica entre *lead* y vendedor para transformar al primero no solo en cliente, sino más bien en un cliente fidelizado, que compre más veces.
- **Incrementar la facturación:** Máñez (2020) alude a los términos de *front end* y de *back end* para explicar este punto, diciendo que este beneficio de

incremento de facturación se apoya en la dinámica del embudo, en el cual una persona va recibiendo valor a medida que va atravesando las etapas del proceso, estando consciente de ciertas ofertas visibles (*front end*) y otras que no lo están hasta que el prospecto pase por una conversión (*back end*).

3.4.2. Etapas del embudo

A grandes rasgos, el embudo de ventas se divide en tres etapas, las cuales son: la parte superior del embudo (*TOFU - Top Of the Funnel*), la parte de en medio (*MOFU - Middle Of the Funnel*) y la parte inferior (*BOFU - Bottom Of the Funnel*). Toda empresa es libre de subdividir las etapas anteriormente mencionadas de acuerdo con las estrategias que quiera añadir, pero esta división básica está diseñada para representar a groso modo que, en la parte superior, las personas apenas nos están conociendo, en la parte de en medio, ya nos conocen y están interactuando en la dinámica para ver si avanzan o no en el proceso y en la parte inferior se encuentran aquellas personas más propensas a convertirse a clientes. En otras palabras, en la etapa de TOFU se encuentran los MQL ‘s, es decir, *leads* potenciales para recibir contenido de valor por parte de *marketing* y en la etapa de BOFU se encuentran los SQL ‘s, es decir, *leads* potenciales para que se les venda.

De entre todas las esquematizaciones que una empresa le puede dar a su embudo de ventas para personalizarlo, hay una muy popular en particular y que se va a detallar y esa es la estructura del modelo AIDA (Atención, Interés, Deseo -*Decision or Desire* en inglés-, Acción). Zhel (2020) nos provee de una explicación detallada de cada etapa:

- **Atención:** El prospecto conoce nuestra propuesta de valor, ya sea un producto o un servicio. Zhel (2020) nos recuerda el carácter digitalizado de estos procesos, pues hablamos de *marketing* digital y puntualiza en que, en esta etapa, el prospecto visita nuestra página web. Esta etapa debe ser complementada con la presencia de la marca por medio de campañas en redes sociales, pues tratamos con MQL ‘s.
- **Interés:** Zhel (2020) menciona que aquí la persona ya empieza a buscar posibles soluciones a su problema. En la etapa anterior se le despertó la consciencia para percatarse de su problema, en esta etapa ya nos conoce y empieza a seguir nuestra propuesta de valor de cerca. Es momento de recurrir a la dinámica bilateral en que se comparte contenido que aporte valor al prospecto y que él se comprometa con nosotros al realizando acciones de conversión
- **Deseo:** Ya sea deseo o decisión, esta etapa se caracteriza por mostrar un mayor compromiso por parte del cliente. El *lead* ya está caliente, está en un proceso de toma de decisiones y posiblemente dispuesto a pasar a la siguiente etapa.

- **Acción:** Finalmente, una vez llegada a esta etapa, el prospecto ya es un SQL y está dispuesto a llegar a un acuerdo con nosotros para convertirse en cliente.

Figura 1. Embudo de ventas, modelo AIDA.

Una vez analizadas estas etapas, en el apartado de desarrollo del presente trabajo, el lector podrá percatarse, al comparar el tamaño de la BBDD que representan a los MQL 's de *Olist* contra la de aquellos que ya son clientes, que hubo una reducción en la cantidad, esto tiene todo el sentido del mundo ya que los usuarios se filtran principalmente, entre otras razones circunstanciales, por las mismas razones que se dieron para poder considerar a una persona como parte del mercado y, por ende, de nuestro *Target/Buyer Persona.*

3.5. Contexto

Olist fue el proveedor de la base de datos empleada, el cual es el mayor departamento de almacenes en los mercados de Brasil. *Olist* conecta pequeños negocios a lo largo de Brasil a canales con un contrato sencillo. Los comerciantes pueden vender sus productos en las tiendas de Olist y enviarlas directamente a los clientes usando los socios de logística de Olist. Una vez que al cliente se le hace llegar su producto, también se le envía una encuesta vía email para dar su reseña con respecto al proceso de compra.

Sus características permiten la visualización de su información a través de múltiples dimensiones: desde el estatus de la orden, el precio, el pago y el rendimiento del flete, la locación del cliente, atributos del producto y reseñas hechas por los propios clientes. La base de datos incluye también una tabla enfocada a la geolocalización, sustentada gracias a códigos postales.

3.6. Diagrama de Entidad - Relación (ERR)

3.6.1. Funnel

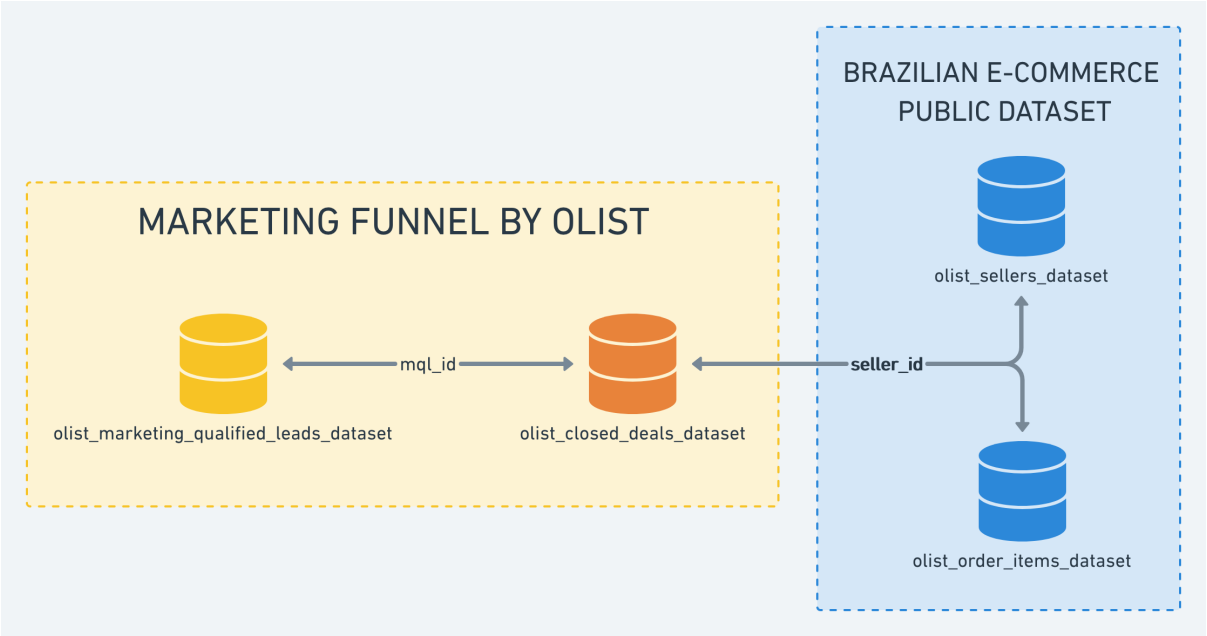


Figura 2. Diagrama de Entidad - Relación del Funnel de *Olist*.

3.6.2. Ecommerce

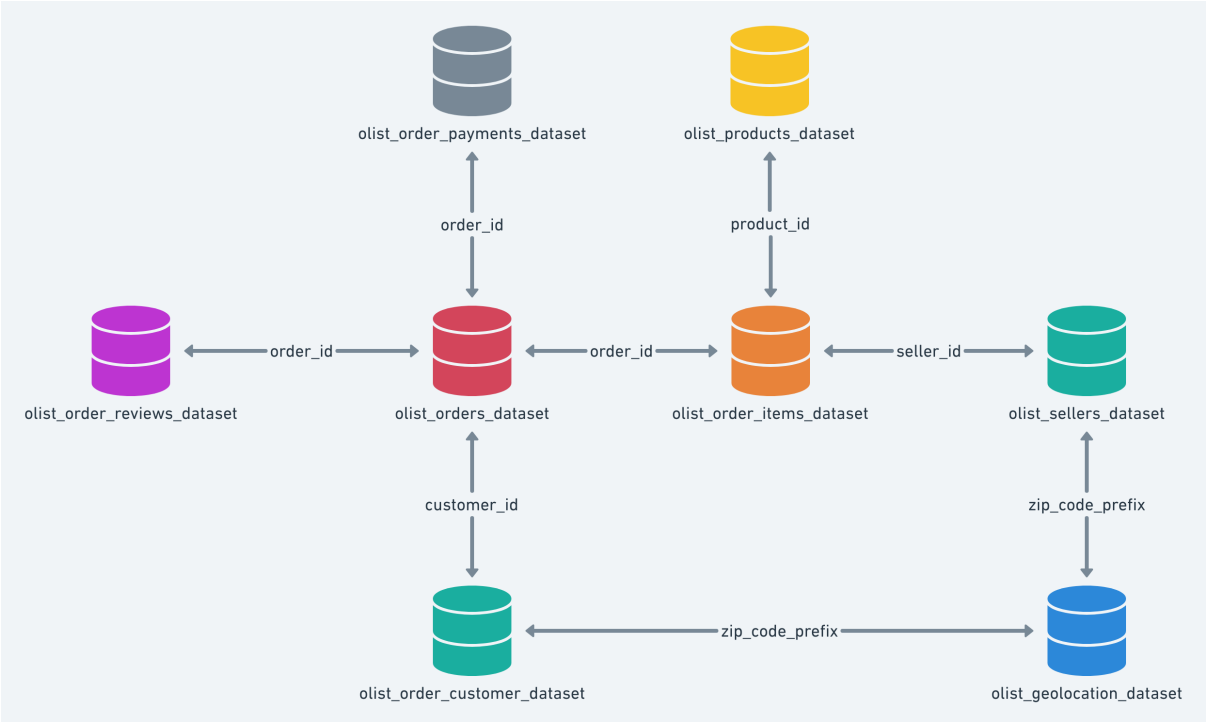


Figura 3. Diagrama de Entidad - Relación de la ecommerce *Olist*.

IV. Desarrollo

A lo largo de esta sección, se mostrará el procedimiento realizado para poder dar respuesta a un conjunto de preguntas que tienen como propósito fungir como guía para cumplir con los objetivos previamente enunciados. Como bien se comentó, el proyecto se realizó haciendo uso del lenguaje Python porque, además de la facilidad, es mucho más óptimo al momento de cargar BBDD con muchos registros, a diferencia de hacerlo en MySQL Workbench.

4.1. Preguntas de investigación

A continuación, se enlistan las interrogantes que tienen como propósito ser una guía para el cumplimiento de los objetivos. Así pues, las interrogantes son:

Sobre el comportamiento de los *leads*:

1. ¿Cuál es el origen más común del cual provienen los *leads*?
2. ¿Cuál es el segmento de negocio más popular?
3. ¿Cuáles son los tipos de lead más comunes?
4. ¿Cuál es el tipo de negocio más popular?
5. ¿Cuántos *leads,* de los que ya hicieron trato, tienen una compañía con respecto al total?
6. ¿Cuál es el promedio de los ingresos mensuales declarados de estos *leads*?

Sobre los productos:

1. ¿Cuáles son los productos más populares de la E-Commerce?
2. ¿Cuáles son sus precios?
3. ¿A qué vendedores corresponden dichos productos populares?

Sobre los tipos de pago:

1. ¿Cuáles son los tipos de pago existentes?
2. ¿Cuáles son los tipos de pago más populares?
3. ¿Cuáles son los respectivos valores de este tipo de pagos y para qué productos?
4. ¿Para qué productos van los anteriores tipos de pago?

Sobre las reseñas:

1. ¿Cómo se relaciona el análisis de sentimientos con la puntuación de las reseñas?
2. ¿En qué difiere con respecto al estatus de envío?

Sobre los modelos:

1. Modelo que puede predecir el si un *lead* tiene empresa o no de acuerdo con:
    a. El comportamiento.
    b. El tipo de *lead*.
    c. El segmento de negocio.
    d. El tipo de negocio.
    e. Su ingreso mensual.

4.2. Análisis preliminar

4.2.1. Limpieza de BBDD y tipos de datos

Se eliminaron columnas cuya información no era relevante para dar respuesta a las preguntas de investigación, pero se conservaron las llaves para poder relacionar las BBDD. Asimismo, se sintetizan las columnas que quedaron de cada BBDD tras su limpieza y el tipo de dato que contiene cada columna.

4.2.1.1. BBDD Funnel

MQL:

1 - MQL

```
[ ] mql = pd.read_csv('olist_marketing_qualified_leads_dataset.csv')
    mql_clean = mql.drop(columns=['first_contact_date', 'landing_page_id'])
    print(mql_clean.shape)
    mql_clean.head(3)
```

```
(8000, 2)
```

	mql_id	origin
0	dac32acd4db4c29c230538b72f8dd87d	social
1	8c18d1de7f67e60dbd64e3c07d7e9d5d	paid_search
2	b4bc852d233dfefc5131f593b538befa	organic_search

```
mql_clean.isna().sum()
```

```
mql_id     0
origin    60
dtype: int64
```

```
mql_clean = mql_clean.dropna()
```

```
mql_clean.isna().sum()
```

```
mql_id    0
origin    0
dtype: int64
```

```
mql_clean.shape
```

```
(7940, 2)
```

```
mql_clean.dtypes
```

```
mql_id    object
origin    object
dtype: object
```

Tratos:

2 - Tratos

```
[ ] deals = pd.read_csv('olist_closed_deals_dataset.csv')
    deals_clean = deals.drop(columns=['sdr_id', 'sr_id', 'won_date', 'has_gtin', 'average_stock', 'declared_product_catalog_size'])
    print(deals_clean.shape)
    deals_clean.head(3)
```

```
(842, 8)
```

	mql_id	seller_id	business_segment	lead_type	lead_behaviour_profile	has
0	5420aad7fec3549a85876ba1c529bd84	2c43fb513632d29b3b58df74816f1b06	pet	online_medium	cat	
1	a555fb36b9368110ede0f043dfc3b9a0	bbb7d7893a450660432ea6652310ebb7	car_accessories	industry	eagle	
2	327174d3648a2d047e8940d7d15204ca	612170e34b97004b3ba37eae81836b4c	home_appliances	online_big	cat	

```
deals_clean.isna().sum()
```

```
mql_id                      0
seller_id                   0
business_segment            1
lead_type                   6
lead_behaviour_profile    177
has_company               779
business_type              10
declared_monthly_revenue    0
dtype: int64
```

```
deals_clean['has_company'] = deals_clean['has_company'].fillna(False)
deals_clean['business_segment'] = deals_clean['business_segment'].fillna('Unknown')
deals_clean['lead_type'] = deals_clean['lead_type'].fillna('Unknown')
deals_clean['lead_behaviour_profile'] = deals_clean['lead_behaviour_profile'].fillna('Unknown')
deals_clean['business_type'] = deals_clean['business_type'].fillna('Unknown')
```

```
deals_clean.isna().sum()
```

```
mql_id                    0
seller_id                 0
business_segment          0
lead_type                 0
lead_behaviour_profile    0
has_company               0
business_type             0
declared_monthly_revenue  0
dtype: int64
```

```
deals_clean.dtypes
```

```
mql_id                    object
seller_id                 object
business_segment          object
lead_type                 object
lead_behaviour_profile    object
has_company                 bool
business_type             object
declared_monthly_revenue   int64
dtype: object
```

4.2.1.2. BBDD E-Commerce

Vendedores:

Vendedores

```
sellers = pd.read_csv('olist_sellers_dataset.csv')
sellers_clean = sellers.drop(columns=['seller_state'])
print(sellers_clean.shape)
sellers_clean.head(3)
```

```
(3095, 3)
```

	seller_id	seller_zip_code_prefix	seller_city
0	3442f8959a84dea7ee197c632cb2df15	13023	campinas
1	d1b65fc7debc3361ea86b5f14c68d2e2	13844	mogi guacu
2	ce3ad9de960102d0677a81f5d0bb7b2d	20031	rio de janeiro

```
sellers_clean.isna().sum()
```

```
seller_id                 0
seller_zip_code_prefix    0
seller_city               0
dtype: int64
```

```
sellers_clean.dtypes
```

```
seller_id                 object
seller_zip_code_prefix     int64
seller_city               object
dtype: object
```

Geolocalización:

Geolocalización

```
geolocation = pd.read_csv('olist_geolocation_dataset.csv')
geolocation_clean = geolocation.drop(columns=['geolocation_state'])
print(geolocation_clean.shape)
geolocation_clean.head(3)
```

```
(1000163, 4)
```

	geolocation_zip_code_prefix	geolocation_lat	geolocation_lng	geolocation_city
0	1037	-23.545621	-46.639292	sao paulo
1	1046	-23.546081	-46.644820	sao paulo
2	1046	-23.546129	-46.642951	sao paulo

```
geolocation_clean.isna().sum()
```

```
geolocation_zip_code_prefix    0
geolocation_lat                0
geolocation_lng                0
geolocation_city               0
dtype: int64
```

```
geolocation_clean.dtypes
```

```
geolocation_zip_code_prefix      int64
geolocation_lat                float64
geolocation_lng                float64
geolocation_city                object
dtype: object
```

Clientes:

Clientes

```
customers = pd.read_csv('olist_customers_dataset.csv')
customers_clean = customers.drop(columns=['customer_unique_id', 'customer_state'])
print(customers_clean.shape)
customers_clean.head(3)
```

```
(99441, 3)
```

	customer_id	customer_zip_code_prefix	customer_city
0	06b8999e2fba1a1fbc88172c00ba8bc7	14409	franca
1	18955e83d337fd6b2def6b18a428ac77	9790	sao bernardo do campo
2	4e7b3e00288586ebd08712fdd0374a03	1151	sao paulo

```
customers_clean.isna().sum()
```

```
customer_id                0
customer_zip_code_prefix   0
customer_city              0
dtype: int64
```

```
customers_clean.dtypes
```

```
customer_id                object
customer_zip_code_prefix    int64
customer_city              object
dtype: object
```

Órdenes:

Órdenes

```
orders = pd.read_csv('olist_orders_dataset.csv')
orders_clean = orders.drop(columns=['order_purchase_timestamp', 'order_approved_at',
print(orders_clean.shape)
orders_clean.head(3)
```

```
(99441, 3)
```

	order_id	customer_id	order_status
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered
1	53cdb2fc8bc7dce0b6741e2150273451	b0830fb4747a6c6d20dea0b8c802d7ef	delivered
2	47770eb9100c2d0c44946d9cf07ec65d	41ce2a54c0b03bf3443c3d931a367089	delivered

```
orders_clean.isna().sum()
```

```
order_id        0
customer_id     0
order_status    0
dtype: int64
```

```
orders_clean.dtypes
```

```
order_id        object
customer_id     object
order_status    object
dtype: object
```

Artículos ordenados:

Artículos ordenados

```
order_items = pd.read_csv('olist_order_items_dataset.csv')
order_items_clean = order_items.drop(columns = ['shipping_limit_date', 'freight_value'])
print(order_items_clean.shape)
order_items_clean.head(3)
```

```
(112650, 5)
```

	order_id	order_item_id	product_id	
0	00010242fe8c5a6d1ba2dd792cb16214	1	4244733e06e7ecb4970a6e2683c13e61	4843
1	00018f77f2f0320c557190d7a144bdd3	1	e5f2d52b802189ee658865ca93d83a8f	dd7c
2	000229ec398224ef6ca0657da4fc703e	1	c777355d18b72b67abbeef9df44fd0fd	5b51

```
order_items_clean.isna().sum()
```

```
order_id         0
order_item_id    0
product_id       0
seller_id        0
price            0
dtype: int64
```

```
order_items_clean.dtypes
```

```
order_id          object
order_item_id      int64
product_id        object
seller_id         object
price            float64
dtype: object
```

Productos:

Productos

```
products = pd.read_csv('olist_products_dataset.csv')
products_clean = products.drop(columns=['product_name_lenght',
print(products_clean.shape)
products_clean.head(3)
```

```
(32951, 2)
```

	product_id	product_category_name
0	1e9e8ef04dbcff4541ed26657ea517e5	perfumaria
1	3aa071139cb16b67ca9e5dea641aaa2f	artes
2	96bd76ec8810374ed1b65e291975717f	esporte_lazer

```
products_clean.isna().sum()
```

```
product_id                0
product_category_name    610
dtype: int64
```

```
products_clean['product_category_name'] = products_clean['product_category_name'].fillna('Unknown')
```

```
products_clean.isna().sum()
```

```
product_id                0
product_category_name     0
dtype: int64
```

```
products_clean.dtypes
```

```
product_id                object
product_category_name     object
dtype: object
```

Pagos:

Pagos

+ Código + Texto

```
payments = pd.read_csv('olist_order_payments_dataset.csv')
payments_clean = payments.drop(columns = ['payment_sequential', 'payment_installments'])
print(payments_clean.shape)
payments_clean.head(3)
```

```
(103886, 3)
```

	order_id	payment_type	payment_value
0	b81ef226f3fe1789b1e8b2acac839d17	credit_card	99.33
1	a9810da82917af2d9aefd1278f1dcfa0	credit_card	24.39
2	25e8ea4e93396b6fa0d3dd708e76c1bd	credit_card	65.71

```
payments_clean.isna().sum()
```

```
order_id          0
payment_type      0
payment_value     0
dtype: int64
```

```
payments_clean.dtypes
```

```
order_id          object
payment_type      object
payment_value    float64
dtype: object
```

Reseñas:

8 - Reseñas

+ Código + Texto

```
[ ] reviews = pd.read_csv('olist_order_reviews_dataset.csv')
    reviews_clean = reviews.drop(columns=['review_id', 'review_creation_date', 'review_answer_timestamp'])
    print(reviews_clean.shape)
    reviews_clean.head(3)
```

```
(100000, 4)
```

	order_id	review_score	review_comment_title	review_comment_message
0	73fc7af87114b39712e6da79b0a377eb	4	NaN	NaN
1	a548910a1c6147796b98fdf73dbeba33	5	NaN	NaN
2	f9e4b658b201a9f2ecdecbb34bed034b	5	NaN	NaN

```
reviews_clean.isna().sum()
```

```
order_id                    0
review_score                0
review_comment_title    88285
review_comment_message  58247
dtype: int64
```

```
reviews_clean['review_comment_title'] = reviews_clean['review_comment_title'].fillna('Unkown')
reviews_clean['review_comment_message'] = reviews_clean['review_comment_message'].fillna('Unkown')
```

```
reviews_clean.isna().sum()
```

```
order_id                0
review_score            0
review_comment_title    0
review_comment_message  0
dtype: int64
```

```
reviews_clean.dtypes
```

```
order_id                object
review_score             int64
review_comment_title    object
review_comment_message  object
dtype: object
```

4.2.2. Análisis exploratorio

En el presente apartado se calcularon medidas de tendencia central, de dispersión y la representación gráfica de los datos por medio de visualizaciones con respecto a las columnas con datos numéricos de todas las BBDD. Si bien hay varias columnas con datos de tipo int y float, algunas correspondían a columnas de ID o de latitud y longitud, por lo que las únicas columnas que entraron en este análisis fueron: la declaración mensual de ingreso de la tabla de tratos, el precio de la tabla de artículos ordenados y el valor de pago de la tabla de pagos para los tres análisis, la otra columna evaluada fue el de la puntuación de las reseñas, pero a esta última se le dio un tratamiento diferente.

4.2.2.1. Medidas de tendencia central

```
# De la tabla de deals_clean
print(f"La media es: {round(deals_clean['declared_monthly_revenue'].mean(), 2)}")
print(f"La mediana es: {deals_clean['declared_monthly_revenue'].median()}")
print(f"La moda es: {deals_clean['declared_monthly_revenue'].mode()}")
```

```
La media es: 73377.68
La mediana es: 0.0
La moda es: 0    0
dtype: int64
```

```
# De la tabla de order_items_clean
print(f"La media es: {round(order_items_clean['price'].mean(), 2)}")
print(f"La mediana es: {order_items_clean['price'].median()}")
print(f"La moda es: {order_items_clean['price'].mode()}")
```

```
La media es: 120.65
La mediana es: 74.99
La moda es: 0    59.9
dtype: float64
```

```
# De la tabla de payments_clean
print(f"La media es: {round(payments_clean['payment_value'].mean(), 2)}")
print(f"La mediana es: {payments_clean['payment_value'].median()}")
print(f"La moda es: {payments_clean['payment_value'].mode()}")
```

```
La media es: 154.1
La mediana es: 100.0
La moda es: 0    50.0
dtype: float64
```

```
# De la tabla de reviews_clean
print(f"La moda es: {reviews_clean['review_score'].mode()}")
```

```
La moda es: 0    5
dtype: int64
```

4.2.2.2. Medidas de dispersión

```
# De la tabla de deals_clean
print(f"La desviación estándar es: {round(deals_clean['declared_monthly_revenue'].std(), 2)}")
```

```
La desviación estándar es: 1744799.18
```

```
# De la tabla de order_items_clean
print(f"La desviación estándar es: {round(order_items_clean['price'].std(), 2)}")
```

```
La desviación estándar es: 183.63
```

```
# De la tabla de payments_clean
print(f"La desviación estándar es: {round(payments_clean['payment_value'].std(), 2)}")
```

```
La desviación estándar es: 217.49
```

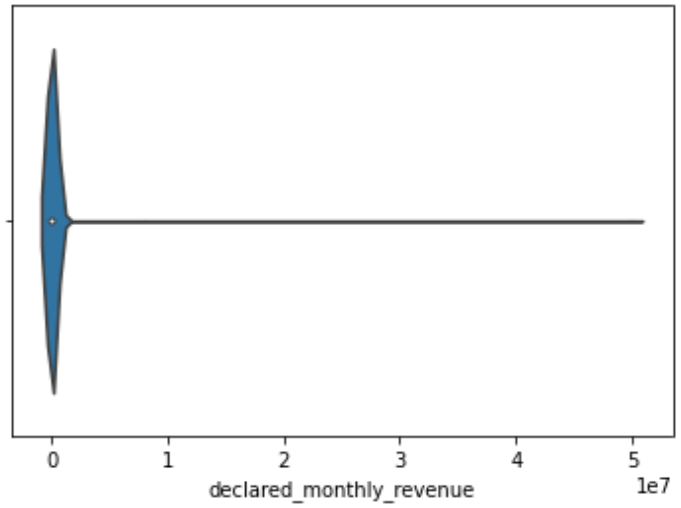
```
# De la tabla de reviews_clean
maximo_score = reviews_clean['review_score'].max()
minimo_score = reviews_clean['review_score'].min()

print(f"El valor máximo es: {maximo_score}")
print(f"El valor mínimo es: {minimo_score}")
print(f"El rango es: {maximo_score - minimo_score}")
```

```
El valor máximo es: 5
El valor mínimo es: 1
El rango es: 4
```

4.2.2.3. Visualizaciones

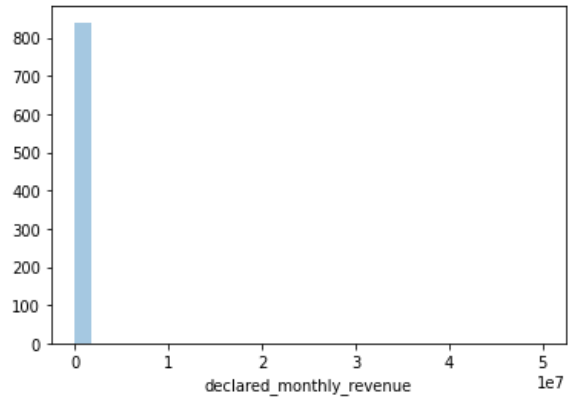
```
sns.violinplot(deals_clean['declared_monthly_revenue'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorat
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fb373c6ea5
```



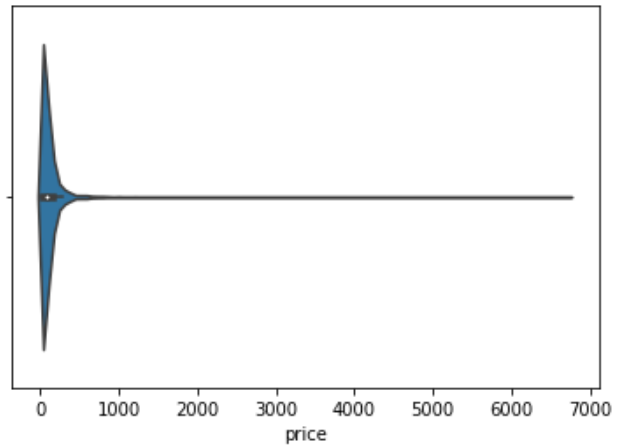

```
ax = sns.distplot(deals_clean['declared_monthly_revenue'], kde = False, norm_hist=False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `dis
  warnings.warn(msg, FutureWarning)
```



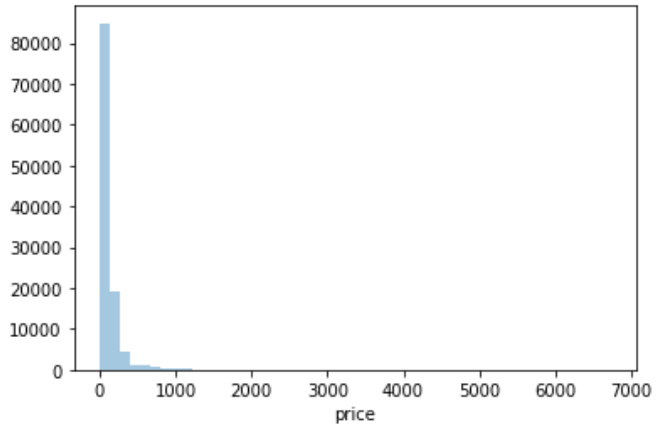
```
sns.violinplot(order_items_clean['price'])
```

```
/usr/local/lib/python3.7/dist-packages/seabor
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7
```



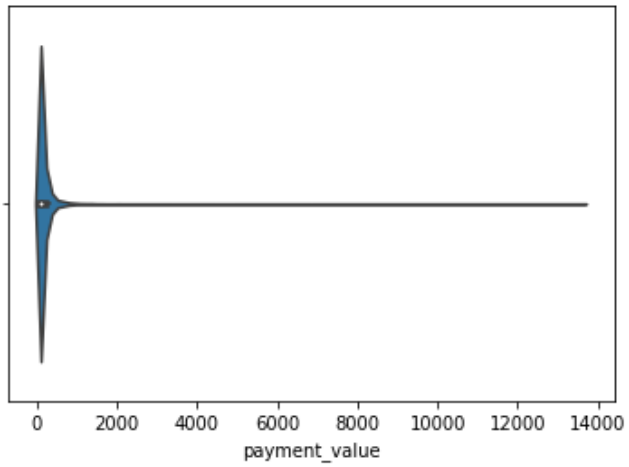

```
ax = sns.distplot(order_items_clean['price'], kde = False, norm_hist=False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: Futur
  warnings.warn(msg, FutureWarning)
```



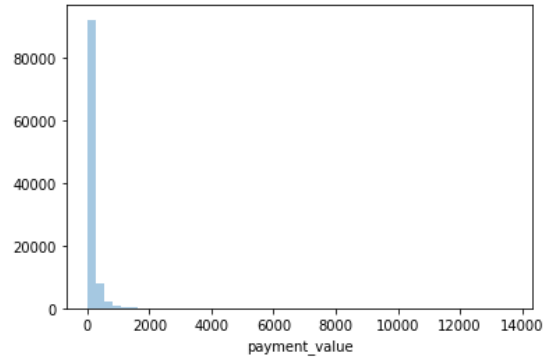

```
sns.violinplot(payments_clean['payment_value'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fb3
```



```
ax = sns.distplot(payments_clean['payment_value'], kde = False, norm_hist=False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarni
  warnings.warn(msg, FutureWarning)
```



Interpretaciones: Gracias al código se puede observar un comportamiento muy similar. Teniendo en cuenta el lienzo sobre el cual se pintan las visualizaciones, podemos concluir que hay muchísimos datos atípicos que es lo que genera una asimetría por la derecha en las tres distribuciones antes impresas, por ello las medidas de dispersión son sumamente elevadas y la moda se vuelve en una medida de tendencia central muy representativa de la población, pues nos muestra que, tanto los ingresos de los clientes, como los precios de los productos o los valores de pago no son sorprendentes, valiendo estos $0, $59.9 y $50, respectivamente. Esto representa el valor de la mayor cantidad de dinero que un emprendedor puede esperar con toparse, esto se complementará con los análisis posteriores.

4.3. Análisis de datos

4.3.1. Comportamiento de leads

4.3.1.1. Pregunta 1

1.1. Orígenes más populares

```
[53] mql_clean['origin'].unique()
```

```
array(['social', 'paid_search', 'organic_search', 'email', 'unknown',
       'referral', 'direct_traffic', 'display', 'other_publicities',
       'other'], dtype=object)
```

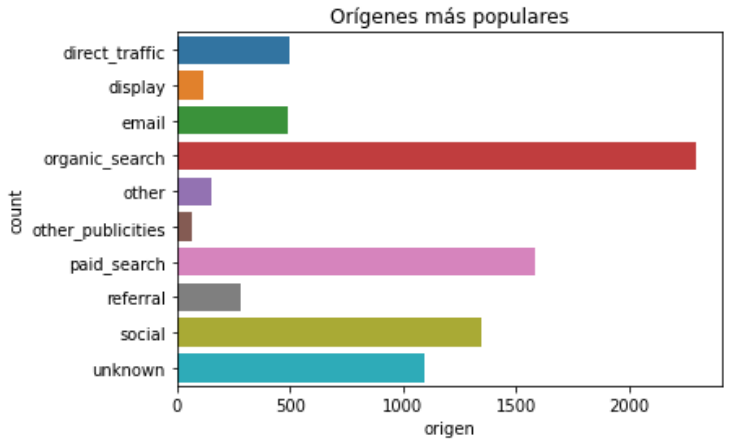
```
[54] origenes_mas_populares = mql_clean.groupby('origin')['mql_id'].count()
     origenes_mas_populares
```

```
origin
direct_traffic        499
display               118
email                 493
organic_search       2296
other                 150
other_publicities      65
paid_search          1586
referral              284
social               1350
unknown              1099
Name: mql_id, dtype: int64
```

```
origenes_mas_populares_cuenta = origenes_mas_populares
```

```
ax = sns.barplot(origenes_mas_populares_cuenta, origenes_mas_populares.index, orient = 'h')
ax.set_title('Orígenes más populares')
ax.set(xlabel='origen');
ax.set(ylabel='count');
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the fo
  FutureWarning
```



4.3.1.2. Pregunta 2

1.2. Segmentos de negocio más populares

```
[57] segmento_de_negocio_mas_popular = deals_clean.groupby('business_segment')['mql_id'].count()
     segmento_de_negocio_mas_popular = segmento_de_negocio_mas_popular.sort_values(ascending=False).head()
     segmento_de_negocio_mas_popular
```

```
business_segment
home_decor                          105
health_beauty                        93
car_accessories                      77
household_utilities                  71
construction_tools_house_garden      69
Name: mql_id, dtype: int64
```

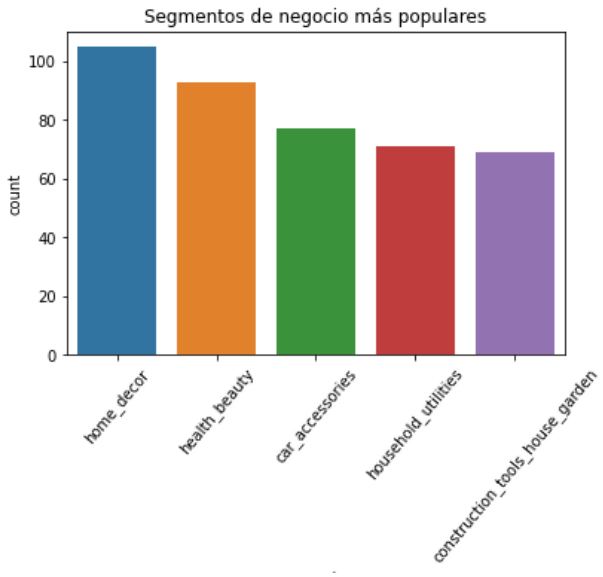
```
[58] segmento_de_negocio_mas_popular_DF = pd.DataFrame(segmento_de_negocio_mas_popular)
     segmento_de_negocio_mas_popular_DF
```

business_segment	mql_id
home_decor	105
health_beauty	93
car_accessories	77
household_utilities	71
construction_tools_house_garden	69

```
segmento_de_negocio_mas_popular_conteo = segmento_de_negocio_mas_popular_DF['mql_id']
segmento_de_negocio_mas_popular_conteo
```

```
business_segment
home_decor                          105
health_beauty                        93
car_accessories                      77
household_utilities                  71
construction_tools_house_garden      69
Name: mql_id, dtype: int64
```

```
ax = sns.barplot(segmento_de_negocio_mas_popular_DF.index, segmento_de_negocio_mas_popular_conteo)
ax.set_title('Segmentos de negocio más populares')
ax.set(xlabel='negocio');
ax.set(ylabel='count');
ax.set_xticklabels(ax.get_xticklabels(), rotation = 50);
```



4.3.1.3. Pregunta 3

1.3. Tipos de lead más comunes

```
[117] deals_clean['lead_type'].unique()
```

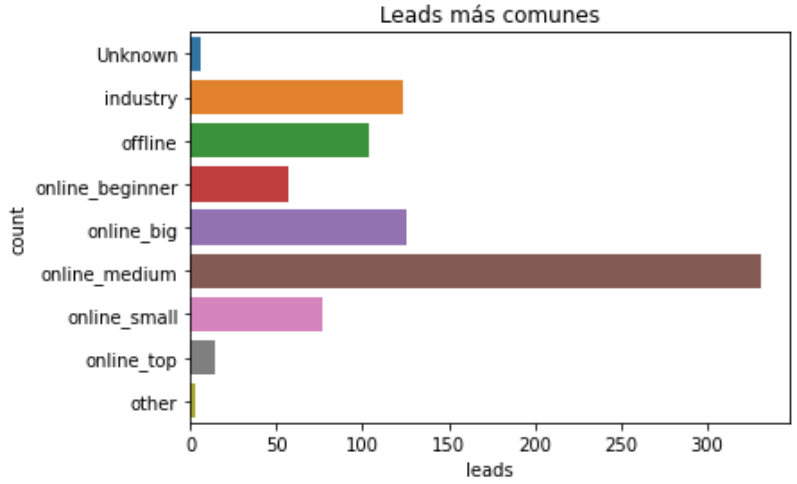
```
array(['online_medium', 'industry', 'online_big', 'online_small',
       'offline', 'online_top', 'online_beginner', 'other', 'Unknown'],
      dtype=object)
```

```
[118] leads_mas_comunes = deals_clean.groupby('lead_type')['mql_id'].count()
      leads_mas_comunes
```

```
lead_type
Unknown              6
industry           123
offline            104
online_beginner     57
online_big         126
online_medium      332
online_small        77
online_top          14
other                3
Name: mql_id, dtype: int64
```

```
ax = sns.barplot(leads_mas_comunes, leads_mas_comunes.index, orient = 'h')
ax.set_title('Leads más comunes')
ax.set(xlabel='leads');
ax.set(ylabel='count');
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarn
  FutureWarning
```



4.3.1.4. Pregunta 4

1.4. Tipo de negocio más popular

```
[120] deals_clean['business_type'].unique()
```

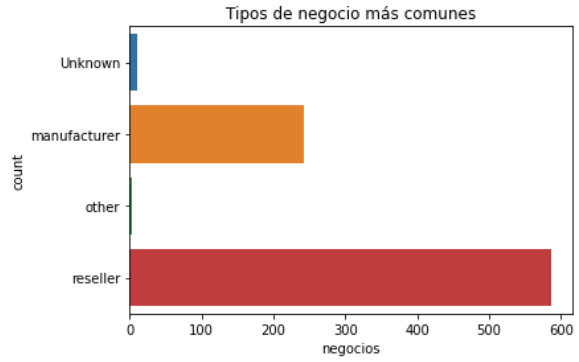
```
array(['reseller', 'manufacturer', 'other', 'Unknown'], dtype=object)
```

```
[123] tipos_negocio_mas_comunes = deals_clean.groupby('business_type')['mql_id'].count()
      tipos_negocio_mas_comunes
```

```
business_type
Unknown          10
manufacturer    242
other             3
reseller        587
Name: mql_id, dtype: int64
```

```
ax = sns.barplot(tipos_negocio_mas_comunes, tipos_negocio_mas_comunes.index, orient = 'h')
ax.set_title('Tipos de negocio más comunes')
ax.set(xlabel='negocios');
ax.set(ylabel='count');
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the i
  FutureWarning
```



4.3.1.5. Pregunta 5

1.5. Lead que tienen compañía

```
[125] tiene_o_no_compañia = deals_clean.groupby('has_company')['mql_id'].count()
      tiene_o_no_compañia
```

```
has_company
False    784
True      58
Name: mql_id, dtype: int64
```

```
[129] tiene_o_no_compañia_DF = pd.DataFrame(tiene_o_no_compañia)
      tiene_o_no_compañia_DF
```

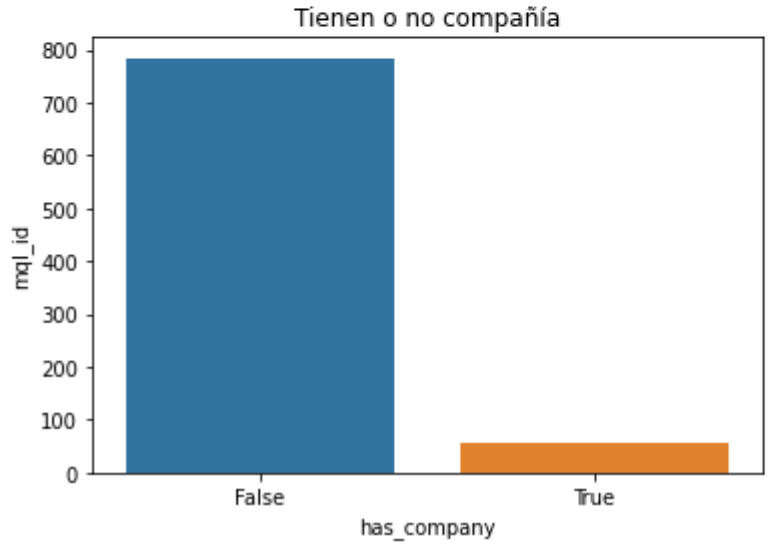
has_company	mql_id
False	784
True	58

```
tiene_o_no_compañia_conteo = tiene_o_no_compañia_DF['mql_id']
tiene_o_no_compañia_conteo
```

```
has_company
False    784
True      58
Name: mql_id, dtype: int64
```

```
ax = sns.barplot(tiene_o_no_compañia_DF.index, tiene_o_no_compañia_conteo)
ax.set_title('Tienen o no compañía');
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWar
  FutureWarning
```




```
si_tienen = tiene_o_no_compañia_DF.loc[True]
si_tienen
```

```
mql_id    58
Name: True, dtype: int64
```

```
no_tienen = tiene_o_no_compañia_DF.loc[False]
no_tienen
```

```
mql_id    784
dtype: int64
```

```
# Porcentaje de los que sí tienen con respecto al total:
round((si_tienen / (si_tienen + no_tienen)) * 100, 2)
```

```
mql_id    6.89
dtype: float64
```

4.3.1.6. Pregunta 6

1.6. Promedio de ingresos mensuales declarados

```
[133] round(deals_clean['declared_monthly_revenue'].mean(),2)

      73377.68
```

4.3.2. Productos

4.3.2.1. Pregunta 1

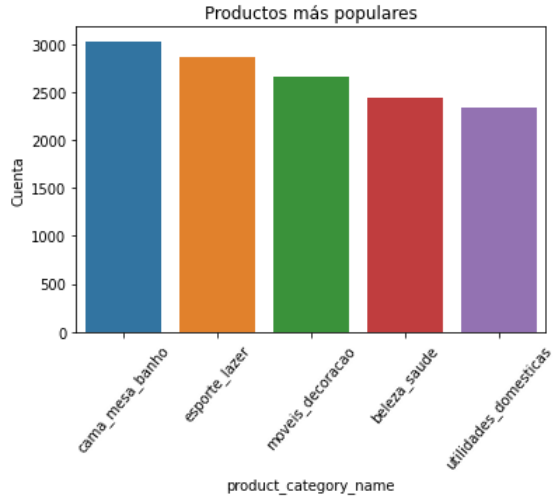
2.1. Productos más populares

```
[61] productos_conteo = products_clean.groupby('product_category_name')['product_id'].count()
     productos_conteo = pd.DataFrame(productos_conteo)
     productos_mas_populares = productos_conteo.sort_values('product_id', ascending=False)
     productos_mas_populares_grafica = productos_mas_populares.head()
     productos_mas_populares_grafica
```

product_category_name	product_id
cama_mesa_banho	3029
esporte_lazer	2867
moveis_decoracao	2657
beleza_saude	2444
utilidades_domesticas	2335

```
ax = sns.barplot(productos_mas_populares_grafica.index, productos_mas_populares_grafica['product_id'])
ax.set_title('Productos más populares')
ax.set(ylabel='Cuenta')
ax.set_xticklabels(ax.get_xticklabels(), rotation = 50);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following va
  FutureWarning
```



4.3.2.2. Pregunta 2

2.2. Precios de productos populares

```
[203] productos_y_ordenes_items = pd.merge(products_clean, order_items_clean, left_on='product_id', right_on='product_id').sort_index()
      productos_y_ordenes_items.head()
```

	product_id	product_category_name	order_id	order_item_id	seller_id	price
0	1e9e8ef04dbcff4541ed26657ea517e5	perfumaria	e17e4f88e31525f7deef66779844ddce	1	5670f4db5b62c43d542e1b2d56b0cf7c	10.91
1	3aa071139cb16b67ca9e5dea641aaa2f	artes	5236307716393b7114b53ee991f36956	1	b561927807645834b59ef0d16ba55a24	248.00
2	96bd76ec8810374ed1b65e291975717f	esporte_lazer	01f66e58769f84129811d43eefd187fb	1	7b07b3c7487f0ea825fc6df75abd658b	79.80
3	cef67bcfe19066a932b7673e239eb23d	bebes	143d00a4f2dde4e0364ee1821577adb3	1	c510bc1718f0f2961eaa42a23330681a	112.30
4	9dc1a7de274444849c219cff195d0b71	utilidades_domesticas	86cafb8794cb99a9b1b77fc8e48fbbbb	1	0be8ff43f22e456b4e0371b2245e4d01	37.90

```
[209] cama_mesa_banho = productos_y_ordenes_items[productos_y_ordenes_items['product_category_name'] == 'cama_mesa_banho']
      cama_mesa_banho['price'].unique()
```

```
array([  71.99,   41.99,  145.  ,  107.6 ,  180.  ,   89.9 ,  649.9 ,
         40.  ,   83.  ,   95.9 ,   95.  ,   99.9 ,  117.99,   99.88,
        109.  ,   29.99,   69.99,   59.  ,   24.9 ,   96.99,   64.9 ,
         95.03,  286.  ,   55.  ,   59.99,   49.  ,   48.  ,   25.5 ,
        112.9 ,  113.  ,   49.9 ,   74.99,   59.4 ,   34.9 ,   77.  ,
```

```
cama_mesa_banho_maximo = cama_mesa_banho['price'].max()
cama_mesa_banho_minimo = cama_mesa_banho['price'].min()
rango_cama_mesa_banho = cama_mesa_banho_maximo - cama_mesa_banho_minimo
rango_cama_mesa_banho
```

```
1992.99
```

```
esporte_lazer = productos_y_ordenes_items[productos_y_ordenes_items['product_category_name'] == 'esporte_lazer']
esporte_lazer['price'].unique()
```

```
array([ 79.8 ,  26.7 , 285.  , ...,  31.99,  42.31, 187.43])
```

```
esporte_lazer_maximo = esporte_lazer['price'].max()
esporte_lazer_minimo = esporte_lazer['price'].min()
rango_esporte_lazer = esporte_lazer_maximo - esporte_lazer_minimo
rango_esporte_lazer
```

```
4054.5
```

```
moveis_decoracao = productos_y_ordenes_items[productos_y_ordenes_items['product_category_name'] == 'moveis_decoracao']
moveis_decoracao['price'].unique()
```

```
array([   9.99,   59.9 ,   49.7 ,  111.9 ,   79.9 ,   69.9 ,   64.9 ,
         39.  ,  139.9 ,   39.9 ,  148.  ,  119.9 ,   75.9 ,  199.99,
```

```
moveis_decoracao_maximo = moveis_decoracao['price'].max()
moveis_decoracao_minimo = moveis_decoracao['price'].min()
rango_moveis_decoracao = moveis_decoracao_maximo - moveis_decoracao_minimo
rango_moveis_decoracao
```

```
1894.1
```

```
beleza_saude = productos_y_ordenes_items[productos_y_ordenes_items['product_category_name'] == 'beleza_saude']
beleza_saude['price'].unique()
```

```
array([ 29.9 ,  95.9 ,  89.9 , ...,  42.5 , 165.7 ,  30.97])
```

```
beleza_saude_maximo = beleza_saude['price'].max()
beleza_saude_minimo = beleza_saude['price'].min()
rango_beleza_saude = beleza_saude_maximo - beleza_saude_minimo
rango_beleza_saude
```

```
3122.8
```

```
utilidades_domesticas = productos_y_ordenes_items[productos_y_ordenes_items['product_category_name'] == 'utilidades_domesticas']
utilidades_domesticas['price'].unique()
```

```
array([ 37.9 ,  69.9 ,  79.99, ..., 108.99, 459.9 , 154.  ])
```

```
utilidades_domesticas_maximo = utilidades_domesticas['price'].max()
utilidades_domesticas_minimo = utilidades_domesticas['price'].min()
rango_utilidades_domesticas = utilidades_domesticas_maximo - utilidades_domesticas_minimo
rango_utilidades_domesticas
```

```
6731.94
```

4.3.2.3. Pregunta 3

2.3. Vendedores de dichos productos

```
[221] productos_ordenes_items_vendedores = pd.merge(productos_y_ordenes_items, sellers_clean, left_on='seller_id', right_on='seller_id').sort_index()
      productos_ordenes_items_vendedores.head()
```

	product_id	product_category_name	order_id	order_item_id	seller_id	price
0	1e9e8ef04dbcff4541ed26657ea517e5	perfumaria	e17e4f88e31525f7deef66779844ddce	1	5670f4db5b62c43d542e1b2d56b0cf7c	10.91
1	a035b83b3628decee6e3823924e0c10f	perfumaria	b18cb761efbe70da4838435a349abd07	1	5670f4db5b62c43d542e1b2d56b0cf7c	268.38
2	091107484dd7172f5dcfed173e4a960e	perfumaria	a7708ffa8966514c098d15e1abfa6417	1	5670f4db5b62c43d542e1b2d56b0cf7c	7.65
3	ccac9976bafbf7e587bd2c29302e2314	perfumaria	206d1a13596872a713dba14504fdf699	1	5670f4db5b62c43d542e1b2d56b0cf7c	268.38
4	2eadf6089620e82047e4d24101dc6759	perfumaria	f8bb4d404d187c79b86ccf852dfa345e	1	5670f4db5b62c43d542e1b2d56b0cf7c	16.88

```
[223] cama_mesa_banho_vendedor = productos_y_ordenes_items[productos_y_ordenes_items['product_category_name'] == 'cama_mesa_banho']
      cama_mesa_banho_vendedor['seller_id'].count()
```

```
11115
```

```
[224] esporte_lazer_vendedor = productos_y_ordenes_items[productos_y_ordenes_items['product_category_name'] == 'esporte_lazer']
      esporte_lazer_vendedor['seller_id'].count()
```

```
8641
```

```
moveis_decoracao_vendedor = productos_y_ordenes_items[productos_y_ordenes_items['product_category_name'] == 'moveis_decoracao']
moveis_decoracao_vendedor['seller_id'].count()
```

```
8334
```

```
beleza_saude_vendedor = productos_y_ordenes_items[productos_y_ordenes_items['product_category_name'] == 'beleza_saude']
beleza_saude_vendedor['seller_id'].count()
```

```
9670
```

```
utilidades_domesticas_vendedor = productos_y_ordenes_items[productos_y_ordenes_items['product_category_name'] == 'utilidades_domesticas']
utilidades_domesticas_vendedor['seller_id'].count()
```

```
6964
```

4.3.3. Tipos de pago

4.3.3.1. Pregunta 1

3.1. Métodos de pago existentes

```
[135] payments_clean['payment_type'].unique()
```

```
array(['credit_card', 'boleto', 'voucher', 'debit_card', 'not_defined'],
      dtype=object)
```

4.3.3.2. Pregunta 2

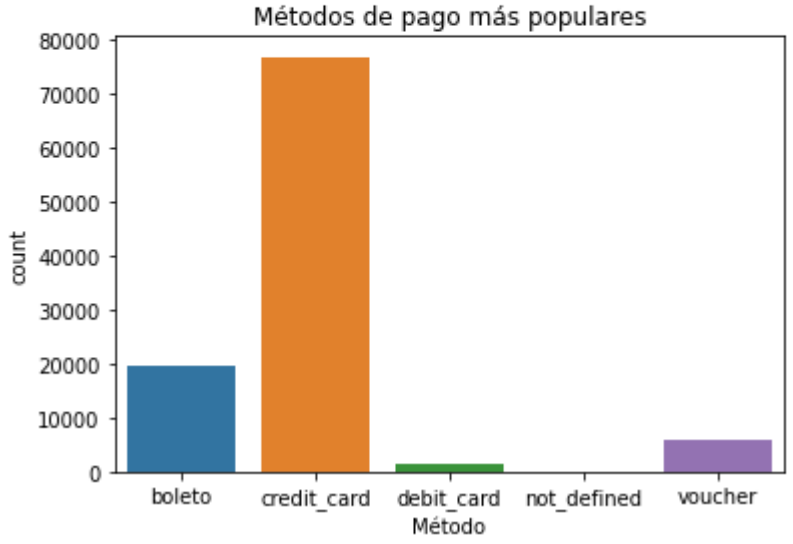
3.2. Métodos de pago más populares

```
[137] metodos_pago = payments_clean.groupby('payment_type')['order_id'].count()
      metodos_pago
```

```
payment_type
boleto         19784
credit_card    76795
debit_card      1529
not_defined        3
voucher         5775
Name: order_id, dtype: int64
```

```
ax = sns.barplot(metodos_pago.index, metodos_pago)
ax.set_title('Métodos de pago más populares')
ax.set(xlabel='Método');
ax.set(ylabel='count');
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_deco
  FutureWarning
```



4.3.3.3. Pregunta 3

3.3. Valores de los tipos de pago

```
[229] boleto_valor = payments_clean[payments_clean['payment_type'] == 'boleto']
      boleto_valor['payment_value'].unique()
```

```
array([ 51.95, 330.66, 283.34, ..., 141.78, 160.89, 363.31])
```

```
[231] boleto_valor_maximo = boleto_valor['payment_value'].max()
      boleto_valor_minimo = boleto_valor['payment_value'].min()
      rango_boleto_valor = boleto_valor_maximo - boleto_valor_minimo
      rango_boleto_valor
```

```
7263.26
```

```
credit_card_valor = payments_clean[payments_clean['payment_type'] == 'credit_card']
credit_card_valor['payment_value'].unique()
```

```
array([ 99.33,  24.39,  65.71, ..., 356.53, 205.71, 100.55])
```

```
credit_card_valor_maximo = credit_card_valor['payment_value'].max()
credit_card_valor_minimo = credit_card_valor['payment_value'].min()
rango_credit_card_valor = credit_card_valor_maximo - credit_card_valor_minimo
rango_credit_card_valor
```

```
13664.07
```

```
debit_card_valor = payments_clean[payments_clean['payment_type'] == 'debit_card']
debit_card_valor['payment_value'].unique()
```

```
array([227.12,  35.14,  76.39, ...,  61.72,  25.41,  61.63])
```

```
debit_card_valor_maximo = debit_card_valor['payment_value'].max()
debit_card_valor_minimo = debit_card_valor['payment_value'].min()
rango_debit_card_valor = debit_card_valor_maximo - debit_card_valor_minimo
rango_debit_card_valor
```

```
4432.12
```

```
voucher_valor = payments_clean[payments_clean['payment_type'] == 'voucher']
voucher_valor['payment_value'].unique()
```

```
array([ 45.17,  69.46,  50.8 , ...,  80.4 ,  41.89, 176.56])
```

```
voucher_valor_maximo = voucher_valor['payment_value'].max()
voucher_valor_minimo = voucher_valor['payment_value'].min()
rango_voucher_valor = voucher_valor_maximo - voucher_valor_minimo
rango_voucher_valor
```

```
3184.34
```

4.3.3.4. Pregunta 4

3.4. Productos con el valor de cada tipo de pago

```
payments_y_order_items = pd.merge(payments_clean, order_items_clean, left_on='order_id', right_on='order_id')
payments_y_order_items.head()
```

	order_id	payment_type	payment_value	order_item_id	product_id	
0	b81ef226f3fe1789b1e8b2acac839d17	credit_card	99.33	1	af74cc53dcffc8384b29e7abfa41902b	213b25e6f54661939f11
1	a9810da82917af2d9aefd1278f1dcfa0	credit_card	24.39	1	a630cc320a8c872f9de830cf121661a3	eaf6d55068dea77334e84
2	25e8ea4e93396b6fa0d3dd708e76c1bd	credit_card	65.71	1	2028bf1b01cafb2d2b1901fca4083222	cc419e0650a3c5ba7718
3	ba78997921bbcdc1373bb41e913ab953	credit_card	107.78	1	548e5bfe28edceab6b51fa707cc9556f	da8622b14eb17ae2831f
4	42fdf880ba16b47b59251dd489d4441a	credit_card	128.45	1	386486367c1f9d4f587a8864ccb6902b	cca3071e3e9bb7d12640

```
[73] payments_order_items_y_products = pd.merge(payments_y_order_items, products_clean, left_on='product_id', right_on='product_id')
     payments_order_items_y_products.head()
```

	order_id	payment_type	payment_value	order_item_id	product_id	seller_id	price
	9b1e8b2acac839d17	credit_card	99.33	1	af74cc53dcffc8384b29e7abfa41902b	213b25e6f54661939f11710a6fddb871	79.80
	ff56c7ee9d3fbd4f8d6	credit_card	93.72	1	af74cc53dcffc8384b29e7abfa41902b	213b25e6f54661939f11710a6fddb871	79.80
	f2d9aefd1278f1dcfa0	credit_card	24.39	1	a630cc320a8c872f9de830cf121661a3	eaf6d55068dea77334e8477d3878d89e	17.00
	3fa0d3dd708e76c1bd	credit_card	65.71	1	2028bf1b01cafb2d2b1901fca4083222	cc419e0650a3c5ba77189a1882b7556a	56.99

```
articulos_por_tipos_pagos = payments_order_items_y_products.groupby('payment_type')['product_category_name'].value_counts()
articulos_por_tipos_pagos
```

```
payment_type  product_category_name
boleto        informatica_acessorios       2158
              cama_mesa_banho              1875
              beleza_saude                 1860
              esporte_lazer                1772
              moveis_decoracao             1735
                                           ...
voucher       livros_importados               2
              moveis_colchao_e_estofado       2
              fashion_esporte                 1
              fashion_roupa_feminina          1
              pc_gamer                        1
Name: product_category_name, Length: 274, dtype: int64
```

```
articulos_pagos_df = pd.DataFrame(articulos_por_tipos_pagos)
articulos_pagos_df
```

		product_category_name
payment_type	product_category_name	
boleto	informatica_acessorios	2158
	cama_mesa_banho	1875
	beleza_saude	1860
	esporte_lazer	1772
	moveis_decoracao	1735
...	...	...
voucher	livros_importados	2
	moveis_colchao_e_estofado	2
	fashion_esporte	1
	fashion_roupa_feminina	1
	pc_gamer	1

```
boleto = articulos_pagos_df.loc['boleto'].head()
boleto
```

product_category_name	product_category_name
informatica_acessorios	2158
cama_mesa_banho	1875
beleza_saude	1860
esporte_lazer	1772
moveis_decoracao	1735

```
tarjeta_credito = articulos_pagos_df.loc['credit_card'].head()
tarjeta_credito
```

product_category_name	product_category_name
cama_mesa_banho	8959
beleza_saude	7566
esporte_lazer	6635
moveis_decoracao	6379
informatica_acessorios	5436

```
voucher = articulos_pagos_df.loc['voucher'].head()
voucher
```

product_category_name	product_category_name
cama_mesa_banho	847
moveis_decoracao	530
utilidades_domesticas	505
esporte_lazer	411
beleza_saude	389

```
tarjeta_debito = articulos_pagos_df.loc['debit_card'].head()
tarjeta_debito
```

product_category_name	product_category_name
beleza_saude	157
informatica_acessorios	148
cama_mesa_banho	142
esporte_lazer	127
utilidades_domesticas	113

4.3.4. Reseñas

4.3.4.1. Pregunta 1

4.1. Análisis de sentimientos

```
[240] reviews_clean.head()
```

	order_id	review_score	review_comment_title	review_comment_message
0	73fc7af87114b39712e6da79b0a377eb	4	Unkown	Unkown
1	a548910a1c6147796b98fdf73dbeba33	5	Unkown	Unkown
2	f9e4b658b201a9f2ecdecbb34bed034b	5	Unkown	Unkown
3	658677c97b385a9be170737859d3511b	5	Unkown	Recebi bem antes do prazo estipulado.
4	8e6bfb81e283fa7e4f11123a3fb894f1	5	Unkown	Parabéns lojas lannister adorei comprar pela I...

```
[243] !pip install TextBlob
```

```
Requirement already satisfied: TextBlob in /usr/local/lib/python3.7/dist-packages (0.15.3)
Requirement already satisfied: nltk>=3.1 in /usr/local/lib/python3.7/dist-packages (from TextBlob) (3.2.5)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from nltk>=3.1->TextBlob) (1.15.0)
```

```
from textblob import TextBlob
```

```
grouped_by_title = reviews_clean.groupby('review_comment_title')['review_comment_message'].max()
grouped_by_title = grouped_by_title.str.lower()
grouped_by_title = grouped_by_title.str.strip()
grouped_by_title = grouped_by_title.str.replace('[^\w\s]', '')
grouped_by_title = grouped_by_title.str.replace('\d', '')
grouped_by_title = grouped_by_title.str.replace('\\n', '')
grouped_by_title = grouped_by_title.dropna()
```

```
polarity = grouped_by_title.apply(lambda x: TextBlob(x).sentiment.polarity)
polarity.name = 'polarity'
```

```
df_with_polarity = pd.concat([grouped_by_title, polarity], axis=1)
```

```
df_with_polarity = pd.concat([grouped_by_title, polarity], axis=1)
```

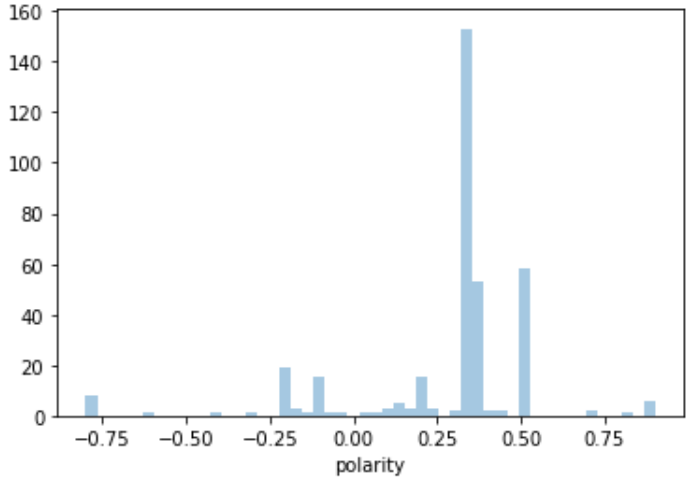
```
df_with_polarity['polarity'] != 0
```

```
review_comment_title
                  False
 10               False
 4                False
 5 estrelas       False
 Avaliando        False
                  ...
👍                False
👍                False
👍👍👍👍👍             False
💥💥💥💥💥             False
🔟                 False
Name: polarity, Length: 4601, dtype: bool
```

```
no_neutral = df_with_polarity[df_with_polarity['polarity'] != 0]
```

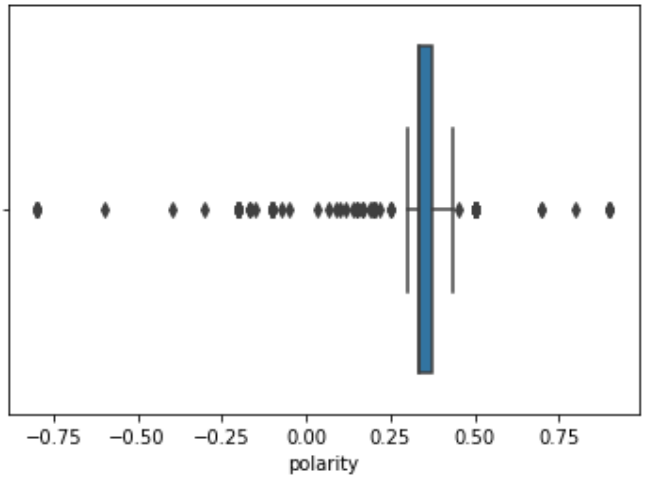
```
sns.distplot(no_neutral['polarity'], kde=False, norm_hist=False);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:25
  warnings.warn(msg, FutureWarning)
```




```
sns.boxplot(no_neutral['polarity']);
```

```
/usr/local/lib/python3.7/dist-packages/seaborr
  FutureWarning
```




```
no_neutral[no_neutral['polarity'] > 0.75]['review_comment_message']
```

```
review_comment_title
Celular para uso simples     foi comprado para meu cunhado utilizalo no dia...
Falta quantia                        eu comprei duas bonecas lol é só veio uma
Gostei do bolso externo           o bolso externo de zíper é ideal para moedas
Gostei do produto.               a informação está clara mas é sempre bom lembr...
Produto pequeno demais.          foi entregue mas produto muito aquem do solici...
Satisfeito com o produto.        perfume de um aroma muito agradável ideal para...
ÓTIMO PRODUTO                              produto de qualidade e no tamanho ideal
Name: review_comment_message, dtype: object
```

```
# Sobre la puntuación de las reseñas
reviews_clean['review_score'].unique()
```

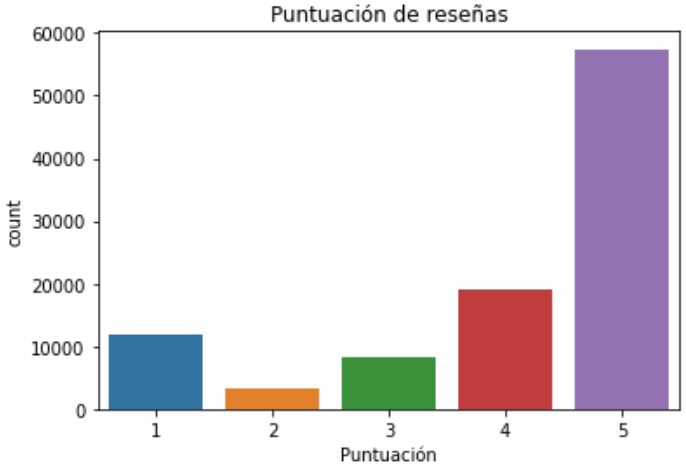
```
array([4, 5, 1, 3, 2])
```

```
puntuacion = reviews_clean.groupby('review_score')['order_id'].count()
puntuacion
```

```
review_score
1    11858
2     3235
3     8287
4    19200
5    57420
Name: order_id, dtype: int64
```

```
ax = sns.barplot(puntuacion.index, puntuacion)
ax.set_title('Puntuación de reseñas')
ax.set(xlabel='Puntuación');
ax.set(ylabel='count');
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_dec
  FutureWarning
```



4.3.4.2. Pregunta 2

4.2. Estatus de envío

```
[263] orders_clean['order_status'].unique()
```

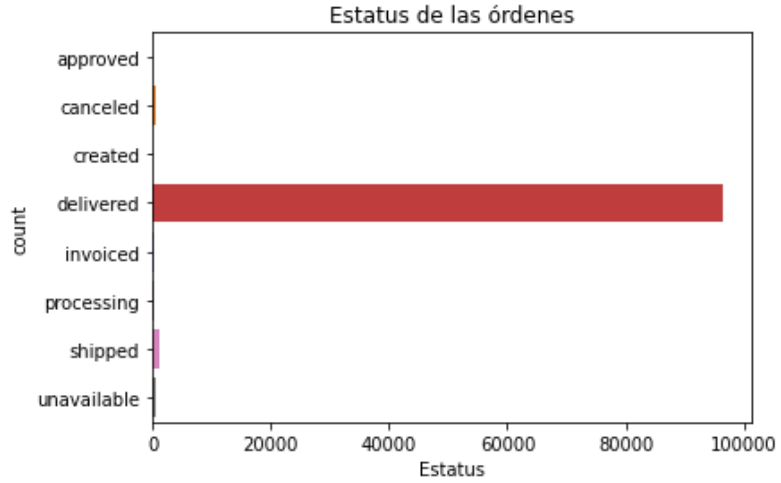
```
array(['delivered', 'invoiced', 'shipped', 'processing', 'unavailable',
       'canceled', 'created', 'approved'], dtype=object)
```

```
[264] estatus = orders_clean.groupby('order_status')['order_id'].count()
      estatus
```

```
order_status
approved           2
canceled         625
created            5
delivered      96478
invoiced         314
processing       301
shipped         1107
unavailable      609
Name: order_id, dtype: int64
```

```
ax = sns.barplot(estatus, estatus.index, orient = 'h')
ax.set_title('Estatus de las órdenes')
ax.set(xlabel='Estatus');
ax.set(ylabel='count');
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorato
  FutureWarning
```



4.3.5. Modelo

5. Modelo

```
273] deals_clean.head()
```

mql_id	seller_id	busin
:76ba1c529bd84	2c43fb513632d29b3b58df74816f1b06	
le0f043dfc3b9a0	bbb7d7893a450660432ea6652310ebb7	ca
:940d7d15204ca	612170e34b97004b3ba37eae81836b4c	hom
5bcae2bafb6dd6	21e1781e36faf92725dde4730a88ca0f	
167a2f6be528e0	ed8cb7b190ceb6067227478e48cf8dde	hom

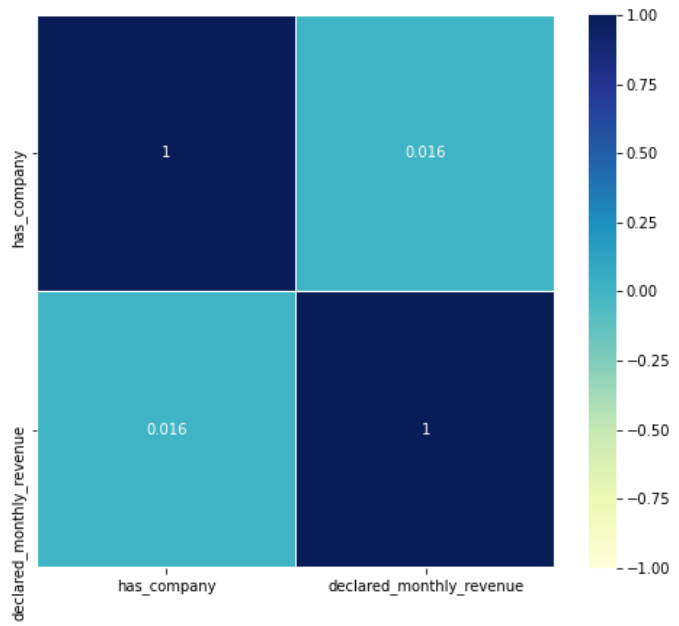
```
deals_clean['has_company'].unique()
```

```
array([False,  True])
```

```
deals_clean.corr()
```

	has_company	declared_monthly_revenue
has_company	1.000000	0.015617
declared_monthly_revenue	0.015617	1.000000

```
plt.figure(figsize=(8, 7))
ax = sns.heatmap(deals_clean.corr(), vmin=-1, vmax=1, annot=True, cmap="YlGnBu", linewidths=.5);
```




```
nuevo_deals_clean = deals_clean.drop(columns=['mql_id', 'seller_id', 'business_segment', 'lead_type', 'lead_behaviour_profile', 'business_type'])
nuevo_deals_clean.head()
```

	has_company	declared_monthly_revenue
0	False	0
1	False	0
2	False	0
3	False	0
4	False	0

```
X = nuevo_deals_clean.drop(columns=['has_company'])
y = nuevo_deals_clean['has_company']
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

```
logreg = LogisticRegression()
```

```
logreg.fit(X_train, y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
y_pred = logreg.predict(X_test)

y_pred
```

```
array([False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
```

```
logreg.score(X_test, y_test)
```

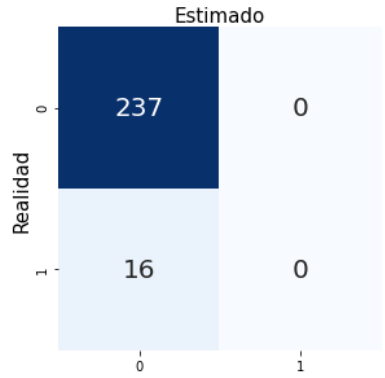
```
0.9367588932806324
```

```
from sklearn import metrics
```

```
cnf_matrix = metrics.confusion_matrix(y_test, y_pred)

cnf_matrix
```

```
array([[237,   0],
       [ 16,   0]])
```

```
class_names=[0,1] # name  of classes
fig, ax = plt.subplots(figsize=(4, 4))
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
# create heatmap
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="Blues" ,fmt='g', cbar=False, annot_kws={"size": 20})
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.ylabel('Realidad', fontsize=15, y=0.5)
plt.xlabel('Estimado', fontsize=15);
```




```
tn, fp, fn, tp = cnf_matrix.ravel()

print("Precision:",metrics.precision_score(y_test, y_pred))
print("Exactitud:",metrics.accuracy_score(y_test, y_pred))
print("Sensibilidad:",metrics.recall_score(y_test, y_pred))
print("Especificidad:", tn / (tn + fp))
```

```
Precision: 0.0
Exactitud: 0.9367588932806324
Sensibilidad: 0.0
Especificidad: 1.0
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_class
  _warn_prf(average, modifier, msg_start, len(result))
```

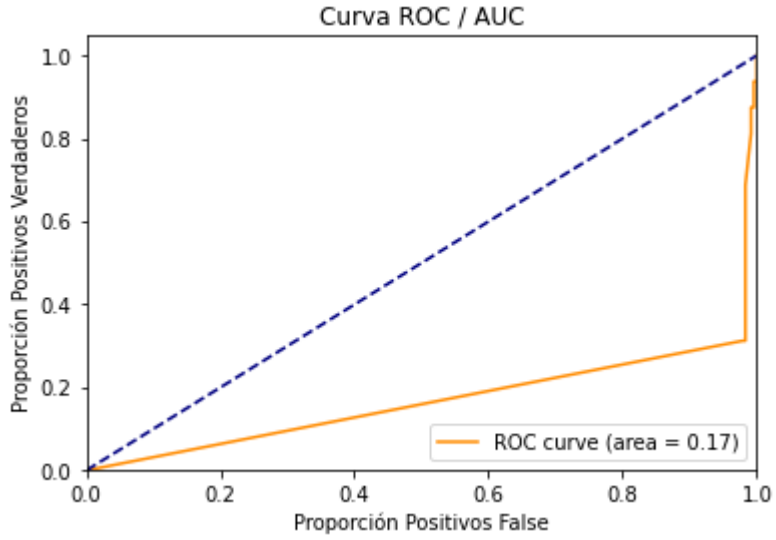
```
y_pred_proba = logreg.predict_proba(X_test)

y_pred_proba[:10,:]
```

```
array([[0.50163138, 0.49836862],
       [0.5       , 0.5       ],
       [0.5       , 0.5       ],
       [0.5       , 0.5       ],
       [0.5       , 0.5       ],
       [0.5       , 0.5       ],
       [0.5       , 0.5       ],
       [0.5       , 0.5       ],
       [0.5       , 0.5       ],
       [0.5       , 0.5       ]])
```

```
from sklearn.metrics import roc_curve, auc
```

```
fpr, tpr, _ = roc_curve(y_test, y_pred_proba[:, 1])
roc_auc = auc(fpr, tpr)
```

```
plt.figure()
plt.plot(fpr, tpr, color='darkorange',
         label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Proporción Positivos False')
plt.ylabel('Proporción Positivos Verdaderos')
plt.title('Curva ROC / AUC')
plt.legend(loc="lower right")
plt.show()
```



4.4. Resultados

En este apartado se procederá a dar una respuesta más concisa a las preguntas de investigación. Así pues, las respuestas a las interrogantes planteadas son:

Sobre el comportamiento de los *leads*:

1. ¿Cuál es el origen más común del cual provienen los *leads*?
   R: El origen más común por el cual las personas llegan a las páginas de aterrizaje de *Olist* es por medio de la búsqueda orgánica.

2. ¿Cuál es el segmento de negocio más popular?
   R: El segmento de negocio más popular es el dedicado a las decoraciones para el hogar.

3. ¿Cuáles son los tipos de lead más comunes?

R: Los tres tipos más comunes son los relacionados a las búsquedas en línea, siendo de nivel medio y alto, y aquellos enfocados a la industria.

4. ¿Cuál es el tipo de negocio más popular?
   R: El de los revendedores, seguido de lejos por las manufactureras.

5. ¿Cuántos *leads,* de los que ya hicieron trato, tienen una compañía con respecto al total?
   R: Muy pocos, solamente de 58 con respecto al total, conformando estos un porcentaje del 6.89% de la población.

6. ¿Cuál es el promedio de los ingresos mensuales declarados de estos *leads*?
   R: Es de $73,377.68; pero ya vimos que este dato es muy sensible a los datos atípicos, por lo que guiarse por la media de $0 es más representativo. Lo más probable es que la gran mayoría de los *leads* no proporcione información al respecto sobre su declaración de ingresos.

Sobre los productos:

1. ¿Cuáles son los productos más populares de la E-Commerce?
   R: Los cinco productos más populares son los artículos de casa (muy relacionado con el segmento de negocio) como son la cama, la mesa y del baño; seguidos de artículos de ocio y deportivos; muebles y decoraciones; salud y belleza y, finalmente, utilidades domésticas.

2. ¿Cuáles son sus precios?
   R: Los artículos de casa tienen un rango de precio de $1,992.99. Los artículos de ocio y deportivos tienen un rango de $4,054.5. Los muebles y las decoraciones tienen un rango de $1,894.1. Los artículos de salud y belleza tienen un rango de $3,122.8. Y las utilidades domésticas tienen un rango de $6,731.94.

3. ¿A qué vendedores corresponden dichos productos populares?
   R: En el mismo orden con el que anteriormente fueron nombrados, fueron los vendedores con los ID de 11115, 8641, 8334, 9670 y 6964, respectivamente.

Sobre los tipos de pago:

1. ¿Cuáles son los tipos de pago existentes?
   R: Tarjeta de crédito, boleto, voucher, tarjeta de débito y hay un rubro de método de pago no definido.

2. ¿Cuáles son los tipos de pago más populares?

R: De más a menos popular son: tarjeta de crédito, boleto, voucher y tarjeta de débito.

3. ¿Cuáles son los respectivos valores de este tipo de pagos y para qué productos?
   R: El boleto abarca un rango de $7,263.26; la tarjeta de crédito de $13,664.07; la de débito de $4,432.12 y el voucher de $3,184.34.

4. ¿Para qué productos van los anteriores tipos de pago?
   R: El boleto, de entre los más populares, va para accesorios de informática, productos domésticos, salud y belleza, deporte y ocio y para de decoraciones y muebles; la tarjeta de crédito va para artículos de domésticos, salud y belleza, deporte y ocio, muebles y decoraciones y accesorios de informática; el voucher va para artículos domésticos, muebles y decoraciones, utilidades domésticas, deporte y ocio y salud y belleza; y, finalmente, la tarjeta de débito va para salud y belleza, informática y accesorios, artículos domésticos, deporte y ocio y utilidades domésticas.

Sobre las reseñas:

1. ¿Cómo se relaciona el análisis de sentimientos con la puntuación de las reseñas?
   R: Los pasos se siguieron tal y como en el ejemplo proporcionado en el material didáctico de BEDU, a pesar de que se hace la aclaración de que TextBlob solo puede hacer análisis de sentimientos en inglés, alemán y francés, se decidió intentar para portugués.

   Tal y como se puede observar en las visualizaciones (gráfica de distribución y Boxplot), hay muchísimos datos atípicos y esto, sospecho, que es por el problema del idioma. De tal manera que, finalmente, la tendencia general de la polaridad tiende a rondar el 0.30, dando un resultado no tan positivo. Sin embargo, si lo comparamos contra el análisis de medidas de tendencia central que se hizo y, contra la gráfica de barras que muestra la frecuencia de las calificaciones, vemos que, en efecto, la gran mayoría tiene un 5 (calificación máxima) de puntuación, revelando cierta tendencia y afinidad entre lo encontrado en el análisis de sentimientos con lo encontrado por los conteos de las puntuaciones.

2. ¿En qué difiere con respecto al estatus de envío?
   R: Debido a que los estatus de orden si clasifican de acuerdo a entradas de texto, no se pudo realizan una correlación entre este dato y la puntuación de la review, sin embargo, podemos apreciar que la inmensa mayoría, para la fecha de capturados dichos datos, tienen el estatus de entregado, estando en sincronía con una buena satisfacción del cliente. Sin embargo, puede ser que

muchas de las razones por las cuales se les dio un puntaje alto en las reseñas fue más bien por la calidad del producto en sí mismo.

Sobre los modelos:

1. Modelo que puede predecir si un *lead* tiene empresa o no:
   R: Este modelo fue enfocado al análisis de la tabla de los tratos cerrados (*closed deals*), en la cual solo habían dos columnas que podían ser relacionadas: *declared_monthly_revenue* por tener datos numéricos, siendo una variable de tipo continua, y *has_company* la cual contiene datos de verdadero y falso, traducidos a 1 y 0, respectivamente, en consecuencia, siendo una variable de tipo discreta. Primero se hizo una correlación, cuyo resultado fue muy pobre, de solo 0.016. Tras realizar el modelo, se obtuvo un modelo con un valor cercano a 0, pues la curva ROC/AUC se fue por el lado de la proporción de falsos positivos. No es un modelo malo en sí mismo, pero elige positivos cuando en realidad son negativos y viceversa. En conclusión, y debido al análisis de correlación, el conocer la declaración mensual de ingresos de los *leads* no nos permite conocer con confianza la probabilidad de si estos cuentan o no con una compañía.

V. Conclusión

A lo largo del presente trabajo se dio respuesta a un conjunto de interrogantes planteadas con el propósito de servir como guía exploratoria para cualquier persona emprendedora que tenga en mente sumarse a la plataforma de *Olist* para ofrecer sus productos. *Olist* es la E-Commerce brasileña más grande y, por ello mismo, hemos visto que tiene mucho potencial que ofrecer, así como también está conformada con mucha competencia. Las BBDD analizadas estaban en su mayoría compuestas por datos descriptivos, cualitativos, por lo cual el mayor análisis que se realizó, con el propósito de encontrar áreas de oportunidad para incursionar, se realizó tras aplicar técnicas de conteo y representación visual. Si bien el análisis de sentimientos y el modelo de regresión logística no son óptimos, fue una buena oportunidad el buscar incluirlos afín que explotar al máximo las oportunidades que nos brinda Python para el análisis de datos.

Finalmente, considero que para cualquier emprendedor es grato y oportuno tener la posibilidad de contar como una infraestructura como *Olist,* pues, tras realizar este análisis y comprobar la buena reputación que se tiene tras revisar las reseñas, *Olist* ofrece la oportunidad de posicionar la marca de una persona por medio de la estructura del embudo de ventas, así como la de satisfacer a aquellas personas que se convierten en clientes mediante sus servicios logísticos. Hay muchas comodidades que cualquiera puede explotar, solo basta con analizar un poco al respecto de en dónde invertir y a qué público vender.

VI. Anexo

6.1. Glosario

BBDD: Abreviación de Base de Datos.
Marketing: De acuerdo con Kerin et al (2004) es "... el proceso de planear y ejecutar la concepción, fijación de precios, promoción y distribución de ideas, bienes y servicios para crear intercambios que satisfagan los objetivos individuales y organizacionales".
Inbound marketing: *Marketing* de atracción, aquel en donde se busca despertar interés en nuestro público objetivo con el propósito de que voluntariamente entre en una dinámica con nosotros, dinámica en la cual solo se le va guiando.
Outbound marketing: *Marketing* tradicional, aquel en donde se busca llegar por métodos directos al cliente. La atención que se despierta en los prospectos no necesariamente es parte de un interés genuino, pues no fueron ellos los que nos buscaron, sino que fuimos nosotros a ellos.
Embudo de ventas: Conceptualización de un proceso a lo largo del cual se busca convertir a un potencial cliente en un cliente por medio de una dinámica bilateral entre el vendedor y el *lead.* Compuesto por diversas etapas que buscan agregar valor a los prospectos a cambio de su compromiso con la marca y filtrar para separar a todas aquellas personas que sí formen parte del mercado de las que no lo son.
TOFU: *Top Of The Funnel* en inglés. Se refiere a la etapa del embudo de ventas en la que apenas hemos llamado la atención de las personas y empiezan tanto a conocernos como a interactuar con nosotros.
MOFU: *Middle Of The Funnel* en inglés. Se refiere a la etapa del embudo de ventas en la que los prospectos ya están más "calientes" (propensos a convertirse en clientes), en esta etapa ya despertamos deseo en los prospectos por adquirir nuestros productos o servicios por medio de la dinámica de agregar valor y recibir compromiso.
BOFU: *Bottom Of The Funnel* en inglés. Se refiere a la etapa del embudo de ventas en la que el prospecto está más "caliente" y ya está dispuesto a convertirse en cliente e, incluso, a fidelizarse con lo que ofrece nuestra marca.
AIDA: Modelo básico del embudo de ventas que divide al mismo en una serie de etapas cuyos nombres demuestran el estado que atraviesan los *leads* para convertirse en clientes. Las siglas representan: Atención - Interés - Deseo/Decisión - Acción.
Lead: El prospecto, forma parte del mercado al que nos dirigimos y, a diferencia de cualquier público, este forma parte de los que son más probables a convertirse en clientes.
Hot lead: Prospectos que, tras haber atravesado un proceso de interacción con el vendedor, se ven más comprometidos a responder a la dinámica y, por ende,

despertar paralelamente un genuino interés por el producto o servicio que ofrece. Es el prospecto más propenso a convertirse en un cliente.

MQL: *Marketing Qualified Lead* en inglés. Se refiere a todas aquellas personas que forman parte de nuestro público objetivo, pero que, debido a la etapa temprana de interacción con nosotros, son propensos a recibir contenido de *marketing,* no es un *hot lead* todavía.

SQL: *Sales Qualified Lead* en inglés. No se confundir con el lenguaje de programación. Se refiere a aquellas personas que ya son propensas a convertirse en clientes, pues ya atravesaron un proceso de interacción por el cual se ven interesadas a adquirir cierto producto o servicio.

Conversión: Cualquier acción determinada por el vendedor y que realice el prospecto. Cada vez que el prospecto realiza más acciones de conversión, recibe mayor valor y nos otorga mayor compromiso, volviéndose más propenso a convertirse en cliente.

Página de aterrizaje: *Landing Page* en inglés. Se refiere a todas aquellas páginas web especializadas en vender un producto o servicio de manera específica, a través de la cual se puede medir el tráfico de personas que la visitan y cuyo propósito, además de ofrecer valor (por medio de *Lead Magnets*) es contar con elementos que inciten a la conversión de los visitantes del sitio (por medio de *Call To Actions*).

B2C: *Business to Customer* en inglés. Se refiere al modelo de negocio que busca vender al cliente final.

B2B: *Business to Business* en inglés. Se refiere al modelo de negocio de empresas que tienen por clientes a otras empresas.

VOC: *Voice of the Client* en inglés. Se refiere a las necesidades, deseos o requerimientos de un cliente con respecto a un producto o un servicio.

Público objetivo: Llamado *Target/Buyer Persona* en inglés. Se refiere a ese segmento del público del mercado al cual le podemos resolver sus necesidades o deseos por medio de nuestros productos o servicios. El público al cual van dirigidos los esfuerzos de ventas.

Lead Magnet: El “imán de *leads*” es un contenido descargable, generalmente a través de las *Landing Pages* cuyos propósitos son aportar valor al prospecto a cambio de compromiso, por ejemplo, es usual que en algunas páginas se pueda descargar cierto contenido de interés como un E-Book a cambio de tener que dejar datos personales.

CTA: *Call To Action* en inglés. El “llamado a la acción” se refiere a todos aquellos eventos que incitan a que el prospecto realice, valga la redundancia, una acción o, dicho de otro modo, promueve a la interacción en la relación bilateral vendedor - potencial cliente con el propósito de que, al actuar, se genere una conversión.

Front end: De acuerdo con el blog de Digital House (s.f.) “es el responsable de la interacción directa del usuario, por lo que se desarrolla cuidando el lado más visual de las aplicaciones”. Se refiere a la interfaz por medio de la cual un usuario interacciona con una página web.

Back end: De acuerdo con el blog de Digital House (s.f.), se refiere a “la parte del sitio con la que los usuarios no tienen contacto”. Lo anterior se refiere a páginas web, corresponde al servidor.

6.2. Diagrama Entidad - Relación (Unificado)

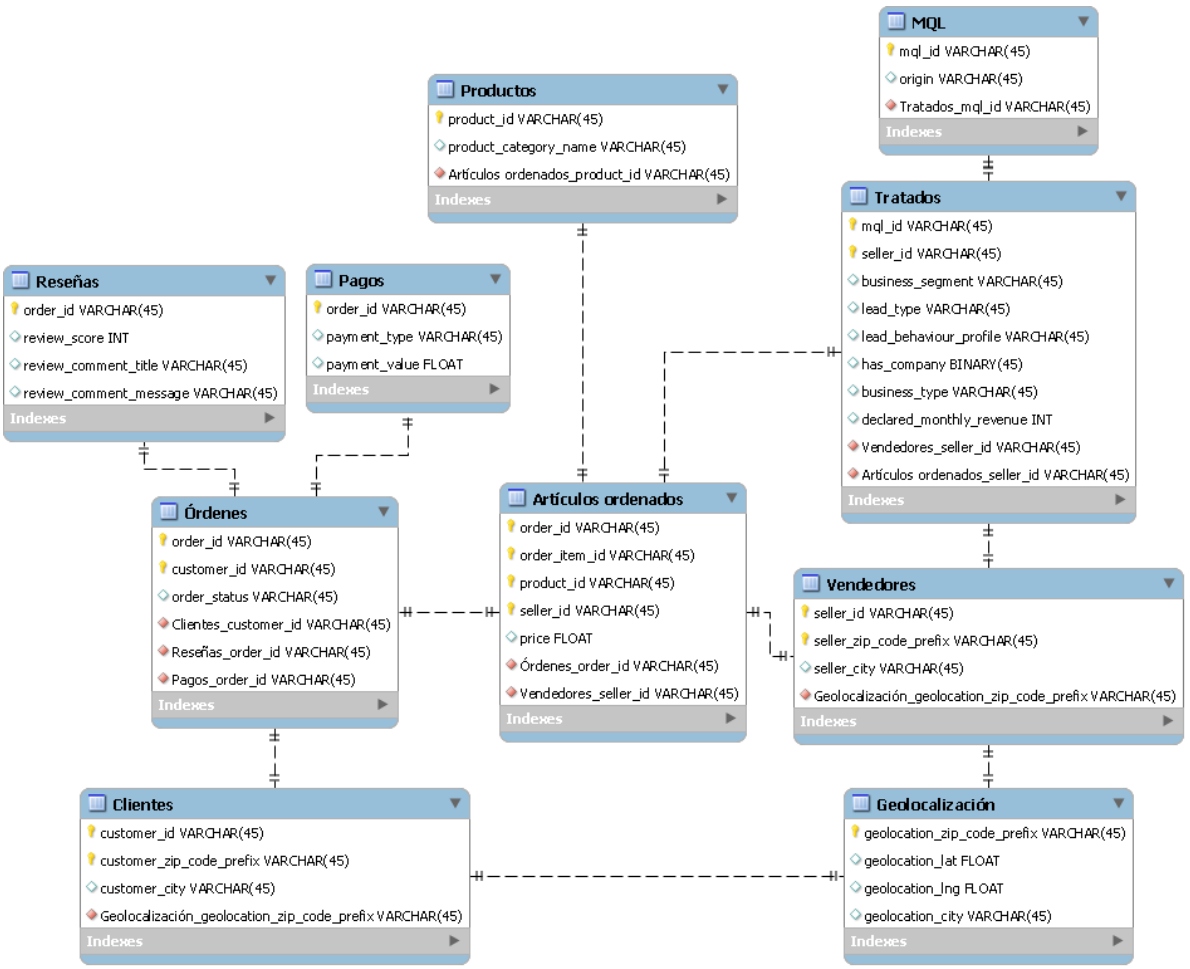


Figura 4. Diagrama de Entidad - Relación unificado de la ecommerce *Olist* y de su embudo de ventas.

6.3. Liga a presentación

https://docs.google.com/presentation/d/1KAABVyJBMeQDfyyWbGoGOT_OSQx1I5ooOVQv8fn5YvY/edit?usp=sharing

VII. Referencias bibliográficas

7.1. Fuentes

- Kerin, R. A., Berkowitz, E. N., Hartley, S. W., & Rudelius, W. (2004). *MARKETING* (Séptima ed.). McGrawHill.
- Digital House. (n.d.). *¿Cuál es la diferencia entre front-end y back-end?* Digital House. Obtenido en Marzo 22, 2021, de https://www.digitalhouse.com/ar/blog/cual-es-la-diferencia-entre-front-end-y-back-end
- Zhel, M. (2016). *The Beginner's Guide to a Sales Funnel*. mailmunch. Obtenido en Marzo 20, 2021, de https://www.mailmunch.com/blog/sales-funnel/
- Máñez, R. (2020). *Qué es un Funnel de Ventas y para qué sirve [Etapas + Ejemplos]*. rubén mañez. Obtenido en Marzo 20, 2021, de https://rubenmanez.com/funnel-de-ventas/
- Samsing, C. (2018, Junio 22). *¿Qué es Inbound Marketing?* HubSpot. Obtenido en Marzo 20, 2021, de https://blog.hubspot.es/marketing/que-es-inbound-marketing-slide-share
- Olist & et al. (n.d.). *Marketing Funnel by Olist*. kaggle. Obtenido en Marzo 20, 2021, de https://www.kaggle.com/olistbr/marketing-funnel-olist
- Olist & et al. (n.d.). *Brazilian E-Commerce Public Dataset by Olist*. kaggle. Obtenido en Marzo 20, 2021, de https://www.kaggle.com/olistbr/brazilian-ecommerce
- Bel, O. (2020, Octubre 29). *¿Qué es el outbound marketing? Ejemplos y direfencias con el inbound*. INBOUNDCYCLE. Obtenido en Marzo 20, 2021, de https://www.inboundcycle.com/blog-de-inbound-marketing/el-significado-de-outbound-marketing
- DElia, J. (2014, Septiembre 10). *Inbound vs Outbound: por qué el Outbound Marketing no te permite despegar [INFOGRAFIA]*. HubSpot. Obtenido en Marzo 20, 2021, de https://blog.hubspot.es/marketing/inbound-vs-outbound-marketing
- Pérez, L. (2019, Mayo 20). *¿Qué son los leads y por qué son tan importantes en el Marketing Digital?* rockcontent. Obtenido en Marzo 20, 2021, de https://rockcontent.com/es/blog/leads-1/

7.2. Figuras

- Figura 1. Embudo de ventas, modelo AIDA. Obtenido en Marzo 20, 2021 de https://apasionados.es/wp-content/imagenes/metodo-aida.jpg

- Figura 2. Diagrama de Entidad - Relación del Funnel de *Olist.* Obtenido en Marzo 20, 2021 de https://www.kaggle.com/olistbr/marketing-funnel-olist
- Figura 3. Diagrama de Entidad - Relación de la ecommerce *Olist.* Obtenido en Marzo 20, 2021 de https://www.kaggle.com/olistbr/brazilian-ecommerce
- Figura 4. Diagrama de Entidad - Relación unificado de la ecommerce *Olist* y de su embudo de ventas.