SPRINT 2

ETL Google Maps

Se necesita para el procedimiento instalar paquetes que no están instalados por default, esto son:

1. `gdown`
2. `google-api-python-client`
3. `google-auth`
4. `google-auth-oauthlib`
5. `google-auth-httplib2`

Una vez instalados se procede a la extracción de la data y como primer paso concatenamos toda la data de la carpeta States para hacer un **merge** con la carpeta Metadata.

Estados:

Se usó Pandas para guardar la data de estas carpetas en dataframes independientes y poder continuar con su manipulación y modelamiento.
Entrando en detalles, como estamos enfocados en negocios dedicados a la parte de turismo, creamos un filtrado con las categorías que nos son relevantes para este fin, entre estas: `'museum'`, `'park'`, `'Hotel'`, `'Motel'`, `'Hostel'`, `'restaurant'`, `'Restaurant'`, `'forest'`, `'gallery'`, `'mall'`, `'pub'`, `'Zoo'` y `'roller coaster'`

Comenzando la parte de transformación, empezamos con protocolos fundamentales como lo es el eliminar los valores duplicados y columnas innecesarias.
El dataframe de Google Maps cuenta con `89.946.359` de datos, sus columnas son: ‘user_id’, ’name’, ’time’, ‘rating’, ‘text’, ‘pics’, ‘resp’, ‘gmap_id’, ‘date’ y ‘state’, donde en primera instancia la mayoría de las columnas no presentan valores nulos, excepto por tres:

- ‘text’ — %43.70 missing
- ‘pics’ — %97.23 missing
- ‘resp’ — %87.74 missing

Debido a que el contenido de ‘text’ nos resulta sumamente importante para un análisis de sentimiento en futuras instancias, solamente borramos las columnas ‘pics’ que está relacionada a imágenes las cuales no nos son de utilidad seguido de

que presentan muy poca cantidad de datos, y la columna ‘resp’ debido a que son datos de la respuesta que se hace a las reviews generadas por los usuarios, lo cual no nos es información necesaria para el proyecto.
Teniendo la columnas necesarias, empezamos viendo la cantidad de estados presentes en Estados Unidos. El dataframe cuenta con la información para los 51 estados, por lo que apuntamos a seleccionar los estados más culturales y turísticos de USA, entre ellos: `'California'`, `'Florida'`, `'New_York'`, `'Nevada'`, `'Indiana'`, `'Massachusetts'`, `'Illinois'`, `'District_of_Columbia'`, `'Louisiana'`, `'Hawaii,` `''South_Carolina'`

Con estos estados en mente contamos entonces con un dataframe filtrado con base en ellos.

SITES:

Repitiendo la modalidad anterior, se comprobó la calidad de los datos en función de querer mejorarla, eliminando así valores duplicados y columnas que no nos fueran relevantes. De las columnas que presentaba este dataframe: ‘name’, ‘address’, ‘gmap_id’, ‘description’, ‘latitude’, ‘longitude’, ‘category’, ‘avg_rating’, ‘num_of_reviews’, ‘price’, ‘hours’,’MISC’, ‘state’, ‘relative_results’ y ‘url’; varias de ellas presentaban un gran porcentaje de datos perdidos. De un total de `277.783` datos:

- description — %71.15 missing
- price — %63.55 missing
- hours — %19.83 missing
- MISC — %7.86 missing
- state — %18.98 missing
- relative_results — %17.38 missing

Teniendo esta información presente, nos encargamos de eliminar las columnas que presentan sumado a la cantidad de datos faltantes, irrelevancia para el tipo de información que estamos utilizando. De esta forma, eliminamos de este dataframe las columnas: `'price'`, `'url'`, `'hours'`, `'description'`, `'state'`, `'MISC'` y `'relative_results'`

Una vez hecho esto, hicimos una unión entre los dataframes de States y Sites a partir de la columna `'gmap_id'`.
Seguido de esta unión, se realizó un nuevo dataframe por un proceso para desanidar el contenido de la columna ‘category’ utilizando una función integrada dentro del archivo utils llamada ‘explode_column’. Se reemplazaron los valores nulos de este nuevo dataframe con las categorías desanidadas.

El dataframe cuenta con 1258 categorías únicas, por lo que hicimos un filtrado para las categorías que corresponden unicamente a negocios que son de nuestro interés procesar: `'museum', 'park', 'Hotel', 'Motel', 'Hostel', 'restaurant', 'Restaurant', 'forest', 'gallery', 'mall', 'pub', 'Zoo', 'roller coaster'`.
Tuvimos cuidados con la sensibilidad a mayúsculas y minúsculas, creamos una máscara booleana para filtrar las filas y la máscara fue aplicada para conservar sólo las filas contenedoras de esas palabras clave. Una vez aplicado el filtro, volvimos a agrupar el dataframe que había sido desanimado, pero con la diferencia de que ahora solo contiene las categorías y elementos de nuestro interés, luego de eliminar nuevamente valores duplicados, este proceso termina dejándonos un dataframe de `10.298.007` de datos.

También mediante un proceso más avanzado, utilizando este dataframe fue extraído el nombre de la ciudad de cada elemento a partir de la utilización de una API la cual toma las coordenadas para asignar de esta forma la ciudad localizada guardando esta nueva información en un dataframe independiente.
Con este nuevo dataframe se hizo el procedimiento fundamental de limpieza y manejo de valores nulos y duplicados, y luego de restablecer los índices, fueron combinados este nuevo dataframe con aquellos previamente modelados a partir de las columnas `"name_x", "longitude"` y `"latitude"` .

Con el dataframe ya terminado, se modificaron los nombres de la columna para una mejor visualización. La modificación en cuestión fue:

- `"name_x"` → `'Business_Name',`
- `"address"` → `'Address',`
- `"City":` → `'City',`
- `"state":` → `'State',`
- `"latitude":` → `'Latitude',`
- `"longitude":` → `'Longitude',`
- `"avg_rating":` → `'Ranking',`
- `"num_of_reviews":` → `'Review_Count',`
- `"date":` → `'Date',`
- `"count_checkin":` → `'Checkin_Count',`
- `"user_id":` → `'User_Id',`
- `"rating":` → `'Stars',`
- `"text":` → `'Text',`
- `"name_y":` → `'User_Name',`
- `"category":` → `'Category'`

Se guardó el dataframe en un archivo .parquet y quedó a disposición para futuro uso

ETL Yelp

Se carga la Data correspondiente a YELP. Inicialmente utilizando 5 dataframes diferentes para organizar la información:

- **Business**
- **Checkin**
- **Review**
- **Tlp**
- **User**

BUSINESS DATAFRAME:

Se observa el contenido *head* del dataframe, el cuenta con 14 columnas con la información de:

- **Business ID** Contiene STR de identificación del negocio
- **Name** STR Contiene el nombre del negocio
- **Address** STR Contiene la dirección completa del negocio
- **City** STR Contiene la ciudad donde está ubicado el negocio
- **State** STR Contiene el estado donde está ubicado el negocio
- **Postal Code** STR Contiene el código postal de la ubicación del negocio
- **Latitude** Float Contiene la Latitud
- **Longitude** Float Contiene la Longitud
- **Is Open** INT Contiene 1 y 0 para indicar si el negocio está abierto o está cerrado respectivamente.
- **Stars** Float Contiene la cantidad de estrellas, entre 0 y 5
- **Review_count** Int Contiene el recuento de reviews del negocio
- **Attributes** STR contiene atributos del negocio como valores
- **Categories** Contiene las categorías del negocio
- **Hours**: STR Contiene el horario en que se encuentra abierto

Para la propuesta del proyecto que estamos llevando a cabo, varias de las columnas son relevantes como también otras no lo son. Por lo tanto eliminamos aquellas irrelevantes las cuales son: “Postal_code”, “is_open”,”attributes”,”hours”

CHECK IN DATAFRAME:

Contiene dos columnas:

- **Business id** id del negocio
- **Date** combinación fecha-hora

En Date la combinación fecha-hora no nos resulta útil para nuestros fines, por lo que procedemos modificarla por un conteo de la cantidad de esos registros que tiene por id de negocio.

REVIEW DATAFRAME:

El dataframe Reviews contiene 9 columnas:

- **Review_id** id de la reseña
- **User_id** id del usuario que escribió la reseña
- **Business_id** id del negocio
- **Stars** cantidad de estrellas
- **Useful** Cantidad de votos como reseña útil
- **Funny** Cantidad de votos como reseña graciosa
- **Cool** Cantidad de votos como reseña cool
- **Text** Reseña en inglés
- **Date** Fecha de la publicación de la reseña

Se sacan las columnas "review_id", "useful", "funny", "cool", "date" ya que su contenido nos resulta irrelevante para el desarrollo del proyecto.

TIP DATAFRAME:

Se puede observar el contenido del dataframe Tip el cual tiene 5 columnas

La información que contiene este dataframe es principalmente para almacenar tip o reseñas más breves orientadas a dar consejos. Ésta información no la vamos a utilizar debido a que es una información repetida de los reviews pero de forma breve.

USER DATAFRAME:

Tenemos este dataframe orientado a la información del usuario que haya hecho la review, organizado con 22 columnas:

- **user_id**
- **name**
- **review_count**
- **yelping_since**
- **useful**
- **funny**
- **cool**
- **elite**
- **friends**
- **fans**
- **average_stars**
- **compliment_hot**
- **compliment_more**
- **compliment_profile**
- **compliment_cute**
- **compliment_list**
- **compliment_note**
- **compliment_plain**
- **compliment_cool**
- **compliment_funny**
- **compliment_writer**
- **compliment_photos**

La mayoría de éstas columnas guardan cantidades de elementos que no nos son relevantes
De esta forma dejamos al dataframe User, con las únicas columnas relevantes que son 'user_id' y 'name', las cuales almacenan los id de usuarios y respectivos nombres

UNIFICACIÓN DE LA INFORMACION

La información proporcionada en el archivo Tip tiene reseñas abreviadas de los usuarios. Como el archivo Review contiene información más detallada de los comentarios y puntuaciones por parte de los usuarios, procedimos a utilizar la misma para unificar los archivos y no usar el archivo Tip.

Se crea un dataframe principal el cual va a unir la información de los dataframes: Business, Review, Checkin y User
Las tablas se vinculan a través de **business_id** y **user_id**

Una vez unida la información, se eliminan los valores duplicados.

Fue necesario hacer un tratamiento con respecto a la información correspondiente a la columna "categories", desanidar la información que contiene y filtrar aquellos de nuestro interés (negocios ya filtrados en datos correspondientes a Google Maps)
Entre estos:
'Museum', 'Park', 'Hotel', 'Motel', 'Hostel', 'Restaurant', 'Restaurant', 'Forest', 'Gallery', 'Mall', 'Agency', 'Rental', 'Pub', 'Zoo', 'Roller Coaster'.

Una vez filtrado el Dataframe y reagrupado, terminamos generando un archivo parquet el cual contiene toda la información filtrada separada en las columnas:

- **'Business_Id',**
- **'Business_Name',**
- **'Address',**
- **'City',**
- **'State',**
- **'Latitude',**
- **'Longitude',**
- **'Ranking',**
- **'Review_Count',**
- **'Checkin_Count',**
- **'User_Id',**
- **'Stars',**
- **'Text',**
- **'User_Name',**
- **'Category’**

EDA

Análisis exploratorio de datos de información suministrada por el sitio Yelp y Google Maps

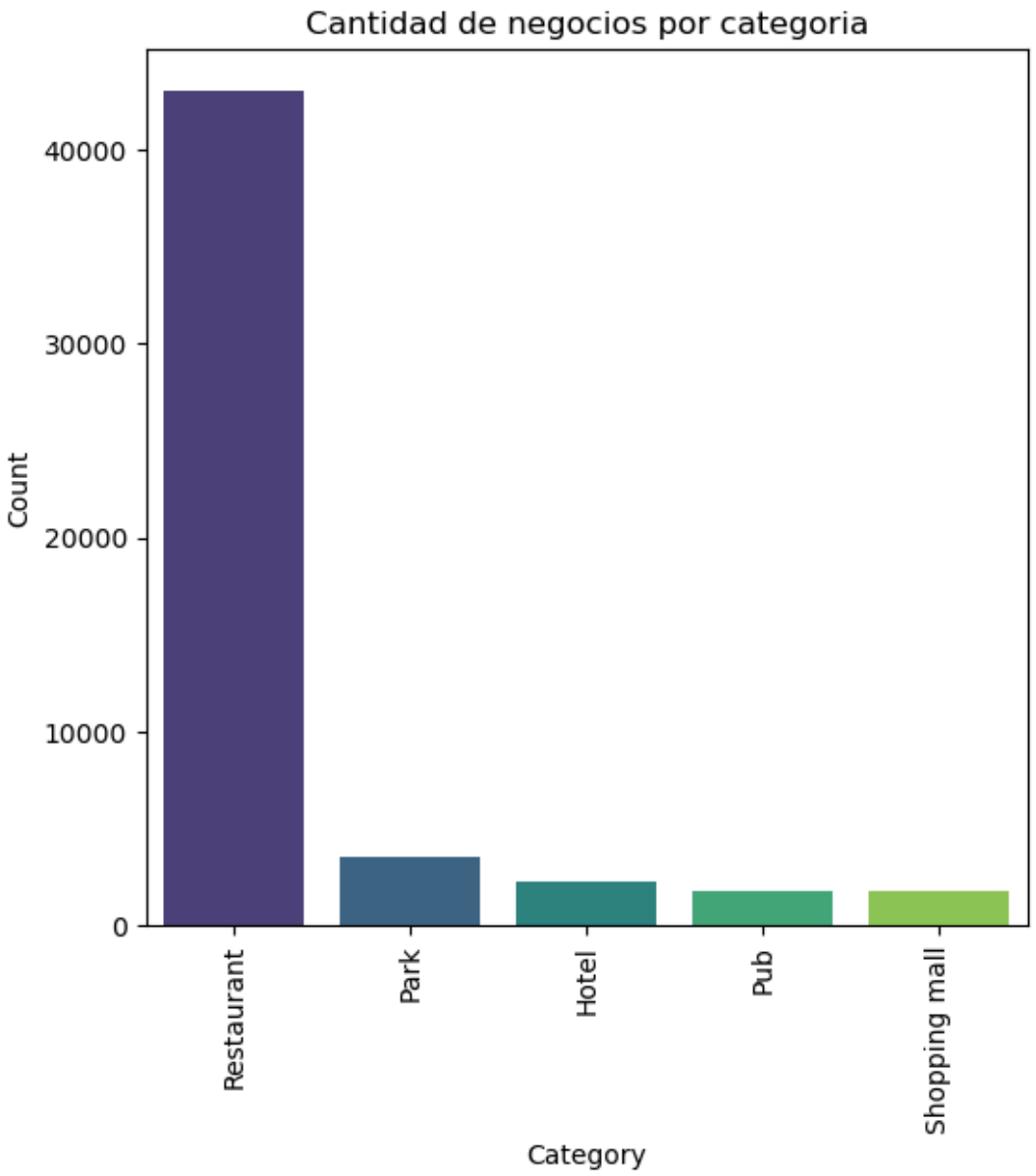
Se cargan los datos desde el Data Lake utilizando pandas.
Como es de nuestro interés un análisis unificado con toda la información, realizamos una concatenación de la información suministrada por ambos sitios en un dataframe.
Estandarizamos valores de la columna states, donde dejamos todos los valores en el mismo tipo de valor (ya que había algunos valores con siglas y otros con nombre completo) y nos queda de esta manera un dataframe con 6.165.047 elementos, los cuales están detallados por las columnas por la información de las columnas: Business_Id, Business_Name, Address, City, State, Latitude, Longitude, Ranking, Review_Count, Date, Checkin_Count, User_Id, Stars, Text, User_Name, Category

Como es de nuestro interés la evaluación por negocios, inicialmente determinamos la cantidad de negocios de los cuales se aporta información, en este caso: 52.854

Se hizo un tratamiento de variables categóricas, desanidando los datos que contiene la columna 'categories' y haciendo un análisis por categoría en un total de 6.394.976 de elementos.

En un primer análisis, se chequeó la cantidad de negocios existentes según la categoría del mismo, y el porcentaje que presentan estas categorías respecto al total de negocios, obteniendo como resultado lo siguiente:

	Category	business_count	porcentaje
8	Restaurant	43034	79.793073
6	Park	3530	6.545279
2	Hotel	2239	4.151524
7	Pub	1809	3.354224
10	Shopping mall	1795	3.328265
0	Art gallery	786	1.457391
4	Museum	472	0.875176
3	Motel	151	0.279982
11	Zoo	53	0.098272
1	Hostel	31	0.057480



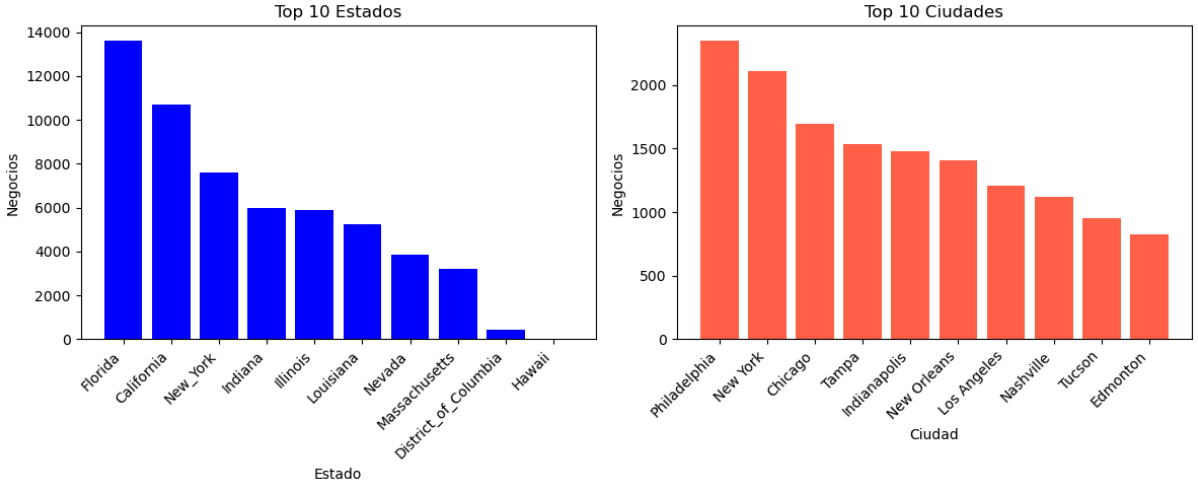
Se puede observar la gran diferencia que hay entre la cantidad de negocios dedicados a restaurantes y la cantidad dedicada al resto de categorías.

Se observó la cantidad de reviews que contiene el database por categoría, donde se pudo notar que las primeras 5 categorías contienen aproximadamente el 90% de la cantidad de negocios que tuvieron alguna reseña, sumado a que los mismos tienen la mayor cantidad de reviews registrados, indicándonos en que las mismas contamos con mayor cantidad de información a la hora de toma de decisión para el inversor.

Analizando por ciudad, determinamos la cantidad de ciudades que contiene la base de Yelp: `3.945`

Ya que contamos con gran cantidad de ciudades, generamos primero un dataframe con la cantidad de negocios por ciudades y solamente analizar el Top 100.
Del Top100 se realizó una revisión de la cantidad de reviews por ciudades y se hizo una unificación de la información mencionada.

Contamos la cantidad de negocios que existen por estado, contamos también la cantidad de negocios que existen por ciudad y utilizamos esa información para crear dos gráficas:



Observación:

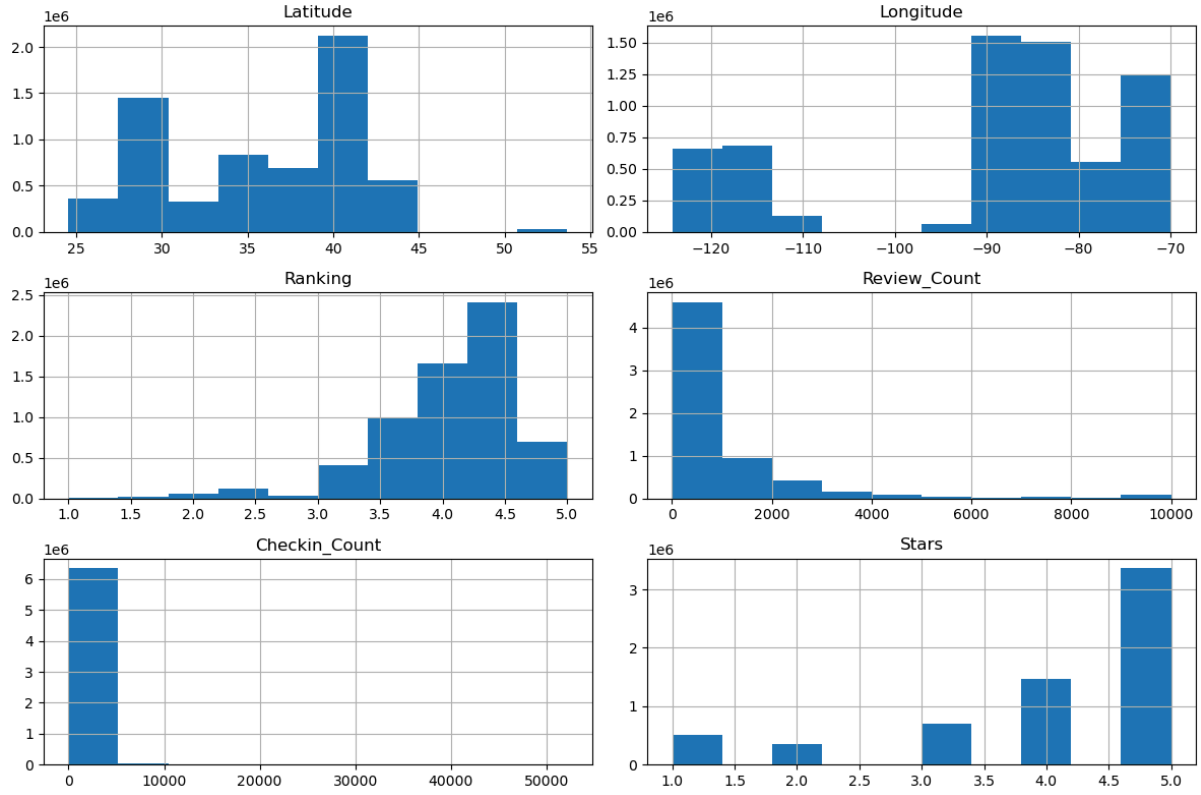
Se pudo ver que dentro del Top10 de ciudades, contienen la mayor cantidad de negocios que tuvieron alguna reseña.
Ese mismo Top10 de ciudades a su vez contiene también la mayor cantidad de reviews, indicándonos que esas ciudades son lugares con mucha recurrencia.
Aquellas ciudades que tienen gran cantidad de reviews y poca cantidad de negocios, es señal que contienen potenciales negocios con mucha recurrencia; lo mismo se aplica a la inversa.

Se ha definido una relación ‘B/R’ entre la Cantidad de negocios sobre la Cantidad de reviews, por ciudad, como un aporte para ayudarnos a visualizar a priori ciudades con potenciales negocios

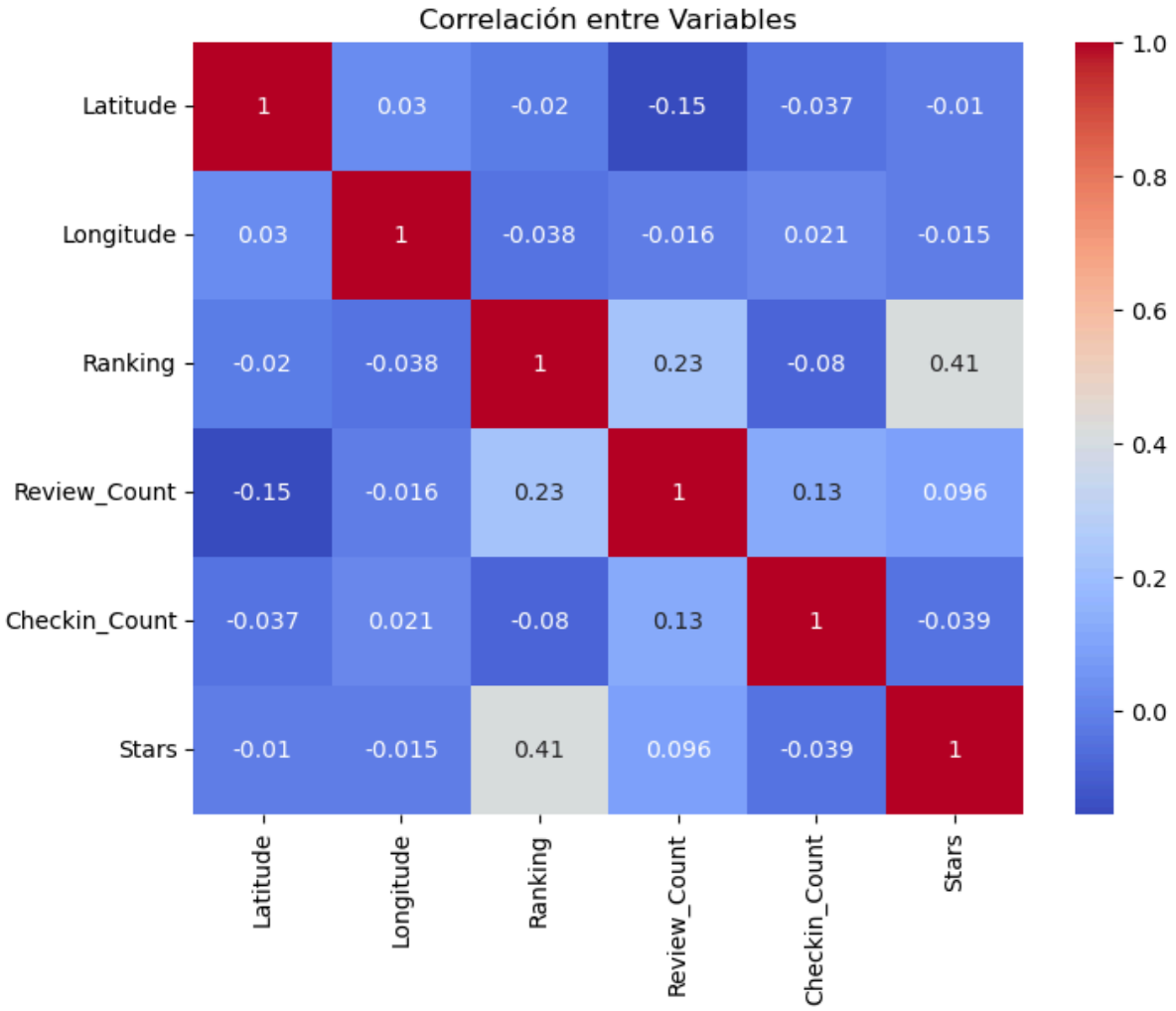
Tratamiento de variables numéricas

Para la columna de stars_y se convirtieron los espacios vacíos en NaN, para rellenar esos NaN con las calificaciones que se encuentran en ‘stars_x’.
Hasta el momento tenemos un total de 806.432 elementos.

Se graficaron histogramas para visualizar y comprender la distribución de los datos numéricos:
Seleccionando las variables "latitude", "longitude", "stars_x", "review_count", "count_checkin" y "stars_y".



- Con respecto a la latitud podemos ver una zona muy marcada entre 28 y 40 donde se encuentra la mayor cantidad de los negocios; con respecto a la longitud de la zona marcada se encuentra entre -120/-110 y -90/-80.
- La puntuación de los negocios (‘stars_x’) aportada por los sitios se encuentra en su mayoría dentro de 3.5 y 4.5.
- La mayoría de los negocios tienen entre 100 y 600 reviews aportados por los usuarios.
- La puntuación aportada por los usuarios (‘stars_y’) es más crítica para observar; varía entre 2.0 a 5.0, lo cual toma relevancia al momento de la elección del negocio por parte del inversionista.



En el diagrama de correlación podemos ver que no se presenta relación entre las variables, excepto para la relación de reviews-checkin y puntuación x-y las cuales ya sabemos que guardan una concordancia.

Tratamiento de Variables Numéricas y Categóricas

Nos resulta importante adicionar al tratamiento de variables categóricas, información de la puntuación de los negocios aportada por el sitio Yelp ‘Ranking’ y ‘Stars’; para ello propondremos agrupar dichos datos determinando el promedio de los mismos.

Se calculó el promedio de ambas calificaciones y se agrupó por negocio incluida la información que la describe.

Al analizar por ciudad, agrupamos por ciudad y contamos la cantidad de negocios únicos, cantidad de reviews, calculamos el promedio de la calificación, sumamos la cantidad de checkin, y al unificar la información quedamos con los datos del Top25:

Notamos que el promedio calculado de las puntuaciones agrupadas por ciudades no presenta mucha variación. Si existe relación directa de la cantidad de reviews aportados por los usuarios con la cantidad de checkin registrados en dichos negocios.

Proponemos la relación entre las variables para poder visualizar a priori aquellas ciudades con potencial de negocios; un valor alto representa un mayor potencial o atractivo para dicha ciuda:

rel_per_city = (reviews_count x count_checkin x avg_punt) / (business_count * 10000)

	City	business_count	Review_Count	avg_punt	Checkin_Count	rel_per_city
5	New Orleans	1403	319496.0	3.850583	505615.0	4433.576906
0	Philadelphia	2347	307589.0	3.547298	685279.0	3185.831246
4	Indianapolis	1479	267327.0	3.654205	287139.0	1896.530172
7	Nashville	1116	169569.0	3.535631	307724.0	1653.143531
3	Tampa	1531	203359.0	3.635636	309690.0	1495.532075
12	Reno	698	116980.0	3.645676	238430.0	1456.783892
15	Santa Barbara	459	82432.0	3.810067	153085.0	1047.486474
11	Saint Louis	773	87738.0	3.546487	225732.0	908.656607
8	Tucson	954	107753.0	3.439804	221139.0	859.171870
23	St. Petersburg	295	69885.0	3.826402	35739.0	323.962620

Con la fórmula propuesta podemos ver que a pesar de que una ciudad tenga una calificación promedio alta, el índice puede ser bajo al no tener suficiente cantidad de checkin en los negocios y por consiguiente baja cantidad de reviews; Tener un índice alto y baja cantidad de negocios comentados, nos aporta información de que en dicha ciudad los negocios son de alta concurrencia y buena calidad, o que existen otros atractivos turísticos por los cuales concurren los usuarios.

Al analizar por categoría, hacemos como hicimos por ciudades y, agrupamos por categoría y contamos la cantidad de negocios únicos, la cantidad de reviews, calculamos promedio de la calificación y sumamos la cantidad de cehckin. Junto a la fórmula utilizada anteriormente.

	Category	business_count	Review_Count	avg_punt	Checkin_Count	rel_per_category
0	Restaurant	43034	7.669834e+06	3.960139	3556203.0	25099.858936
3	Pub	1809	2.361230e+05	3.763973	351962.0	1729.187077
2	Hotel	2239	1.548480e+05	3.280289	335068.0	760.145166
6	Museum	472	7.289200e+04	4.448060	57456.0	394.678876
8	Zoo	53	4.699800e+04	4.450560	8305.0	327.761404
1	Park	3530	3.006010e+05	4.045455	92644.0	319.154044
4	Shopping mall	1795	4.464070e+05	4.285652	1910.0	20.357144
9	Hostel	31	1.680000e+03	3.620139	1187.0	2.328754
5	Art gallery	786	3.630400e+04	4.651488	792.0	1.701567
10	National forest	27	6.480000e+03	4.645608	27.0	0.301035
7	Motel	151	7.912000e+03	3.359885	160.0	0.281679
11	Roller coaster	5	4.130000e+02	4.276194	5.0	0.017661

Podemos ver que la categoría ‘Pubs’ se encuentra en segunda posición debido a que tiene gran cantidad de checkin y reviews, y baja la cantidad de negocios, dándonos un indicio que son empresas que brindan buen servicio y tienen mucha concurrencia, presentándose como una potencial oportunidad de negocio.

Dimensional Tables

A partir de los archivos generados en los procesos de ETL, se hizo una lectura de esta información para almacenar la data en dataframes utilizando la herramienta Pandas para proceder a hacer los modelos de Entidad-Relación.

Se creó la TABLA: ‘CATEGORY’

Category_id	Category
1.	Restaurant
2.	Pub
3.	Hotel
4.	Zoo
5.	Museum
6.	Park
7.	Hostel
8.	Shopping mall
9.	Art gallery
10.	Motel
11.	Roller coaster
12.	National forest

Se creó la TABLA: ‘CITY’

Primero se crea un diccionario con los datos de los estados. Teniendo como columnas State y State_Init, guardando de esta forma el nombre de los estados y las siglas. Para la tabla CIty se utiliza los datos filtrados del dataframe total_data, copiando únicamente la información de las columnas ‘City’ y ‘State’. De esta forma nos queda una tabla con 6.253 valores con las columnas City_Id, City y State.
Se combina este dataframe de ciudades con las siglas de los estados a partir de la columna ‘State’ y luego de añadir una columna de ‘State_Id’ contenedora de un número de identificación para cada uno de los estados, reorganizando y usando las columnas necesarias para esta tabla nos termina quedando:

City_Id	City	State_Id

Se creó la TABLA STATE:

Utilizando lo trabajado anteriormente, realizamos una tabla estado la cual contiene las columnas

State_Id	State	State_Init

Se creó la TABLA USER:

La tabla user va a contener dos columnas ‘User_id’ para la identificación del usuario y ‘User_Name’, para el nombre del usuario.

User_Id	User_Name

Se creó la TABLA BUSINESS:

La cual va a contener las columnas relacionadas a los negocios que existan en la base de datos. Luego de hacer un merge con la información contenida por el dataframe utilizado para la creación de la tabla City, la tabla business termina teniendo las siguientes columnas:

Business_Id	Business_Name	Address	Category	City_Id	Latitude	Longitude

Se creó la TABLA Facts REVIEW:

Donde se va a almacenar la información relacionada con las reseñas de los usuarios a los negocios teniendo por categorías:

Review_Id	Business_Id	Date	User_Id	Category_Id	Ranking	Stars	Text

Se creó la TABLA Facts CHECKIN:

La cual va a almacenar la información de cuando la gente va a estos negocios.

Checkin_Id	Business_Id	Checkin_Count

Teniendo todas estas tablas organizadas y creadas, mandamos la información de estas tablas a ser almacenadas en el Data Warehouse