Documentación de YCNY

Bienvenidos! Esta es la documentación oficial de YCNY

Áreas de la documentación:

***Herramientas utilizadas durante el desarrollo**	***Librerías importadas**
***Guia de uso** Pasos a seguir para el uso de YCNY Manejo de la carga de los archivos parquet y csv	***Pipeline** Gráfico, descripción y rendimiento
***Métricas** Uso de filtros de lluvia y outliers	***Diccionario de columnas** Tabla con columnas usadas y descartadas
***Reglas de negocio**	

– Herramientas utilizadas durante el desarrollo

Durante el desarrollo del análisis fue necesario la utilización de diversas herramientas, que se complementaron con la información brindada. En un inicio se usó *Python* porque es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Además que se caracteriza por ser un lenguaje de programación de rápida escritura y que cuenta con un código abierto.

Se lo complemento en una segunda etapa con un sistema de gestión de bases de datos relacional, en este caso en particular optamos por el uso de *MySQL*, dado que no solo es un software Open Source, sino que su velocidad al realizar las operaciones, lo hace uno de los gestores con mejor rendimiento. Cuenta con un bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.

Para el análisis de los datos se optó por el uso de *Power bi*, que nos da la posibilidad de infraestructura en la nube, páginas de control personalizables, análisis guiado interactivo, buena percepción del informe, amplia disponibilidad de base de datos y sobre todo permite hacer múltiples análisis de datos en un solo informe o panel.

Es importante recordar que es una herramienta colaborativa, que permite que estén conectados a ella un alto número de usuarios y, además, que los paneles e informes se puedan publicar o compartir con toda la compañía. Esto facilita que se fomente la cultura de trabajo en equipo y el análisis de datos dentro de la organización. Al estar alojado en la nube, solo se necesita tener conexión a Internet para poder acceder a toda la información.

– Librerías importadas

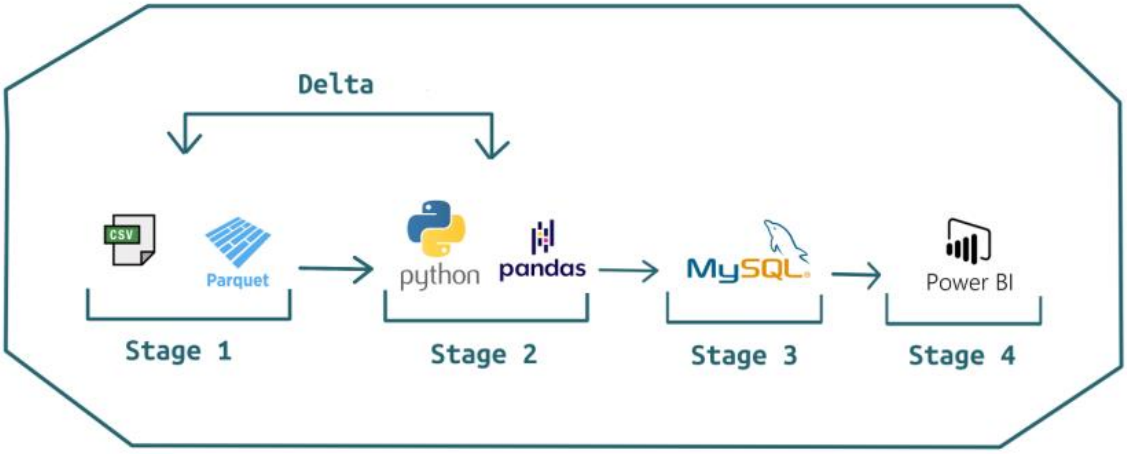
Se utilizaron diversas librerías de Python durante el desarrollo del análisis, las mismas están mencionadas con una breve descripción en los párrafos siguientes.
Numpy, librería muy importante para el ecosistema de Python ya que es la base de todos los cálculos científicos y muchas de las librerías de Machine Learning.
Pandas, una librería de código abierto de variada utilidad en el ámbito de Data Science y Machine Learning, ya que ofrece unas estructuras muy poderosas y flexibles que facilitan la manipulación y tratamiento del dato.
Datetime, proporciona clases para la manipulación de fechas y horas simples y complejas.
Glob, permite encontrar todos los nombres de ruta que coinciden con un patrón especificado.
Shutil, tiene funciones que te permiten copiar, mover, renombrar y borrar archivos.
Mysql,sqlalchemy, multiprocessing, permiten conectar Python con sql y realizar consultas desde Python.

– Guía de uso

Pasos a seguir para el uso de YCNY y el manejo de la carga de los archivos parquet y csv. Para la carga de nuevos registros a la base de datos hay que agregar a la carpeta 'data' los archivos correspondientes al periodo:

*parquet con los datos de los viajes descargado desde https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page, nombrado como el siguiente ejemplo: "yellow_tripdata_2018-01.parquet"
*csv con los datos del clima detallado por hora descargado desde, Weather Data & Weather API | Visual Crossing, nombrado como el siguiente ejemplo: "new york 2018-01-01 to 2018-02-01.csv"
*Una vez corrido el script de python, dentro de la carpeta "data/cargados" se podrán ver los archivos que fueron cargados con éxito.

– Pipeline



Descripción y rendimiento:

Stage 1: Carga de los archivos necesarios (viajes de taxis son, .parquet y datos del clima .csv).

Stage 2: Limpieza, tratamiento y armado de los data frames necesarios para la carga en MySQL. Control para la carga de deltas.

Stage 3: Carga de datos en base de datos previamente creada.

Stage 4: Conexión con Power BI y elaboración de informes necesarios.

Rendimiento:
En pruebas nuestro pipeline procesa 1.000.000 de registros en un tiempo promedio de 3 minutos.
Estos resultados fueron obtenidos con un hardware con las siguientes características:

- i5-7200U
- 12 Gb ram ddr4
- Disco SSD 970 -evo 250GB
- GPU GeForce MX150 (2gb ddr5)

Stage 1:
Ver guía de uso.

Stage 2:
A partir de la carpeta disponibilizada, donde encuentra el archivo .ipynb junto con una carpeta “data”, en la cual se deberán colocar los archivos que se requieran procesar y enviar al datawarehouse. Dentro de esta misma carpeta “data” se va a encontrar una carpeta llamada “cargados”, en la cual se alojarán los archivos (csv para el clima y .parquet para los datos de taxis), luego de la primera carga o de corresponder a una carga incremental.
Por medio de una función se buscan los archivos en la carpeta “data”, y de haber coincidencia en los nombres esperados (“yellow t” para taxis, y “new york 20” para clima) se procede con la ingesta de los datos. A partir de este punto se comienza con el proceso de normalización en la cual se lleva a cabo la eliminación de columnas innecesarias, la limpieza y eliminación de los datos con errores, se marcan los outliers (1 si es outlier, 0 si no lo es), en caso de que alguna de las siguientes columnas se encuentren fuera de los parámetros definidos: Trip distance, Fare amount, Total amount, tip amount.
Declaramos las credenciales de nuestra base de datos y hacemos la creación del motor de la misma.
Posteriormente se crean nuevos dataframes que corresponderá a las tablas dimensionales alojadas en el datawarehouse, las cuales serán cargadas por única vez en la primera ejecución del pipeline siendo controladas por un switch.

Stage 3:
Se normalizan los nombres de las columnas para la carga dentro del data warehouse. Luego se realiza la carga de los data frame que correspondan a MySQL.
Por medio del conector se cargan los registros en el datawarehouse y a partir de un Store Procedure con la ayuda de dos tablas auxiliares para las tablas de hechos de clima y taxis, se guardan de forma definitiva los registros evitando cargar registros duplicados.
Por último, pero no menos importante, se moverán los archivos que fueron procesados a la carpeta “cargados”.

Stage 4:

A partir de las tablas dimensionales y una vista de la tabla de taxis se importan los datos en un dashboard de power bi y aplicando los filtros correspondientes se obtienen los resultados buscados.

– Métricas

Para obtener las métricas del informe presentado, se usaron filtros de fechas, condiciones climáticas y de outliers.

En los viajes promedio, la distancia promedio, ticket promedio y cantidad de viajes por hora se usaron como filtro las condiciones climáticas (lluvia, nieve, despejado,...etc).

Mientras que para obtener las métricas de la relación formas de pago, el porcentaje viajes con propina, la composición de la facturación, el ticket promedio con/sin lluvia y la facturación total, se filtró por outliers

– Diccionario de columnas

Nombre del campo	Descripción
IdVendor	Un código que indica el proveedor de TPEP que proporcionó el registro. 1= Tecnologías móviles creativas, LLC; 2= VeriFone Inc.
tpep_pickup_datetime	Fecha y la hora en que se activó el medidor.
tpep_dropoff_datetime	Fecha y hora en que se desconectó el medidor.
passenger_count	Número de pasajeros en el vehículo.
trip_distance	La distancia del viaje transcurrido en millas reportada por el taxímetro.
IdRateCode	El último código de tarifa vigente al final del viaje. 1= Tarifa estándar 2=JFK 3=Nueva York 4 = Nasáu o Westchester 5=Tarifa negociada
IdPayment_Type	Un código numérico que significa cómo el pasajero pagó por el viaje. 1= tarjeta de crédito 2= Efectivo 3= Sin cargo 4= Disputa
fare_amount	Tarifa de tiempo y distancia calculada por el taxímetro.
extra	Varios extras y recargos. Actualmente, esto solo incluye los cargos de $0.50 y $1 por la hora pico y por la noche.
mta_tax	Impuesto MTA de $0.50 que se activa automáticamente según el medidor tasa en uso.
tip_amount	Recargo de mejora de $0.30 viajes evaluados en el lanzamiento de la bandera. El recargo por mejora comenzó a cobrarse en 2015.
tolls_amount	Importe total de todos los peajes pagados en el viaje.
improvement_surcharge	Recargo de mejora de $0.30 viajes evaluados en el lanzamiento de la bandera. El recargo por mejora comenzó a cobrarse en 2015.
total_amount	El monto total cobrado a los pasajeros. No incluye propinas en efectivo.
IdSemana	Número de la semana del año
IdFecha	Código que concatena año/mes/dia/hora del viaje
IdPUborough	Código que marca el borough donde se inicia el viaje
IdDOborough	Código que marca el borough donde finaliza el viaje
outlier	Marca si el registro tiene un outlier en alguna columna 0=No, 1=Si
IdTaxi	Código único de identificación del viaje

– Reglas de negocio

Se crea una función para limpieza de los registros, se trabaja con las siguientes columnas:

"fare_amount": La cual representa la tarifa plana, si el dato es menor o igual a cero se elimina el registro.
"trip_distance": Representa la distancia de recorrido del viaje, el cual si el dato es menor o igual a cero se elimina el registro.

"total_amount": Representa la facturación total de cada viaje, el cual si el dato es menor o igual a cero se elimina el registro.

"RatecodeID": Representa el tipo de tarifa de viaje. Cuando el Ratecode toma los valores de 6 o 99 se eliminan los registros por no ser de interés para este análisis.

"PUBorough": Representa el lugar donde se inició el viaje, si el dato es "Unknown" se elimina el registro.

"DOBorough": Representa el lugar donde culminó el viaje, si el dato es "Unknown" se elimina el registro.