

Funnel analysis methodology and approach

SQL codes on data extraction

Funnel table extraction

```
WITH tabl
  AS (SELECT 'Downloads' AS step,
            Count(DISTINCT app_download_key) AS counts
    FROM metrocar
  UNION
  SELECT 'sign ups' AS step,
            Count(DISTINCT user_id) AS counts
    FROM metrocar
  UNION
  SELECT 'requests' AS step,
            Count(DISTINCT user_id) AS counts
    FROM metrocar
    WHERE request_ts IS NOT NULL
  UNION
  SELECT 'transaction' AS step,
            Count(DISTINCT user_id) AS counts
    FROM metrocar
    WHERE purchase_amount_usd IS NOT NULL
  UNION
  SELECT 'reviews' AS step,
            Count(DISTINCT user_id) AS counts
    FROM metrocar
    WHERE review IS NOT NULL
  ORDER BY counts DESC)
```

```

SELECT *,
       Lag(counts, 1)
         over ()
       AS shift,
       Round(1 - counts :: NUMERIC / ( Lag(counts, 1)
                                         over () ), 3)
       AS drop_off,
       Round(counts :: NUMERIC / ( Lag(counts, 1)
                                     over () ), 3)
       AS conversion_rate,
       Round(counts :: NUMERIC / (SELECT Count(DISTINCT user_id)
                                         FROM   metrocar), 3)
       AS
percentage_to_signups,
       Round(counts :: NUMERIC / (SELECT Count(DISTINCT app_download_key)
                                         FROM   app_downloads), 3)
       AS
percentage_downloads
FROM   tab1


```

step ▲	counts ▲	shift ▲	drop_off ▲	conversion_rate ▲	percentage_to_signups ▲	percentage_downloads ▲
Downloads	23608	(NULL)	(NULL)	(NULL)	1.340	1.000
sign ups	17623	23608	0.254	0.746	1.000	0.746
requests	12406	17623	0.296	0.704	0.704	0.525
transaction	6233	12406	0.498	0.502	0.354	0.264
reviews	4348	6233	0.302	0.698	0.247	0.184

Saved: as 'metroSummary.csv'

Second table

```
SELECT *  
FROM metrocar offset 50000
```

 metroCarEnd	10/10/2023 04:35
 metroCar350	10/10/2023 04:31
 metroCar300	10/10/2023 04:29
 metroCar250	10/10/2023 04:27
 metroCar200	10/10/2023 04:24
 metroCar150	10/10/2023 04:21

Since the data was very large, metro_car.csv was a union of batched csv downloads using SQL offset for each batch of 50,000 records as shown above. With 396,679 records, and only 50,000 records available for download at a time, the above SQL statement helped in batching the downloads with offsets in batches of +50,000.

Thereafter, Using the glob package in python, all the *.csv files were loaded and union into the data frame used in the python code snippet shown below.

```
import glob  
import pandas as pd  
  
# Define the pattern for the CSV files (e.g., all CSV files in a directory)  
file_pattern = '*.csv'  
  
# Use glob to find all files matching the pattern
```

```
csv_files = glob.glob(file_pattern)

df = pd.DataFrame()

# Loop through the files and read them into dataframes
for file in csv_files:
    data = pd.read_csv(file)
    df = pd.concat([df, data], ignore_index=True)

df.head()

df.shape

df.columns

df2 = df.groupby('user_id', as_index=False)['purchase_amount_usd'].sum()
df2.head()

df3 = df.drop_duplicates('user_id')

df3.shape

df3 = df3[~df3['user_id'].isna()] # remove the single null row in user_id column
df3.drop('purchase_amount_usd', axis = 1, inplace =True)

df3.shape # data in shape

frame = df2.merge(df3, on = 'user_id', how='inner') # merging both dataframes

frame.head()

frame.shape

frame.to_csv('metro_car.csv', index=False) # saved for tableau processing.
```

Both 'metro_car.csv' and 'metroSummary.csv' were used in the tableau as independent tables within the tableau workspace

All other data manipulations were implemented within the tableau workspace, such as data cleaning and presentation.

Link:

https://public.tableau.com/app/profile/samuel.obadan/viz/metro_16973517986500/METROCARFUNNELANALYSIS