Project: Mobile App Usage Analysis

- ETL using Python

Context:

You receive app data usage (every month) from several mobile users. The data is limited to a bunch of apps from well-known publishers. The goal of the exercise is to do exploratory analysis and perform ETL operations to generates multiple target files like statistical data along with rankings of individual Apps based on total users and average minutes per user and rankings of the publishers as per total devices across applications.

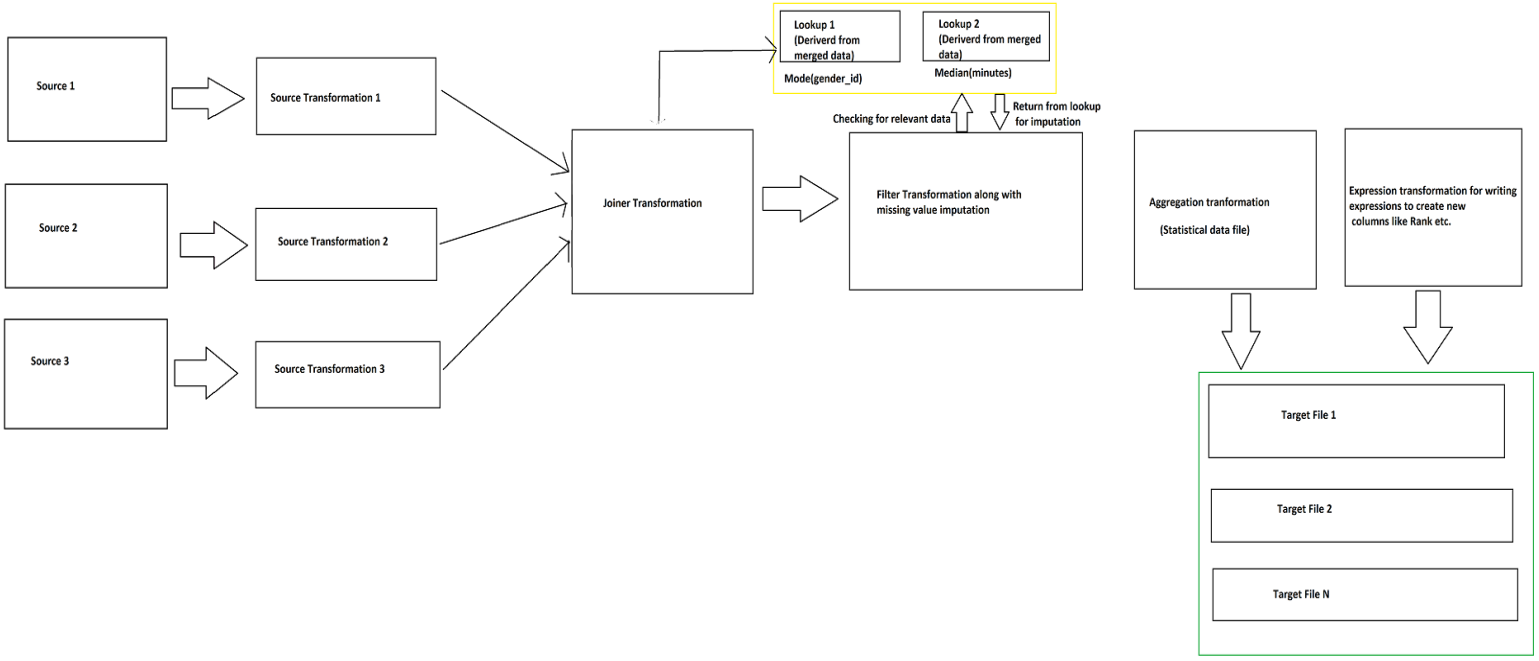
Expected of final dataset:

device_id	gender_id	app_name	minutes	Publisher
B-52-51319	2	Amazon Mobile (Mobile App)	30.0888	Amazon Sites
B-52-24909	1	Amazon Mobile (Mobile App)	30.2022	Amazon Sites
B-52-13680	2	Amazon Mobile (Mobile App)	30.2400	Amazon Sites
B-52-34618	2	Amazon Mobile (Mobile App)	30.3534	Amazon Sites
B-52-17841	1	Amazon Mobile (Mobile App)	30.4668	Amazon Sites

Dataset

Files:
1) User_activity- Monthly aggregated data on minutes spent on an app by a device
2) publishers - Publisher - app_name mapping
3) device_demographics- Demographic information of a device

Flow diagram - ETL



Steps to be followed:

1. 3 different datasets coming from external vendors to which joins need to be performed as per the below schema:

device_id	gender_id	app_name	minutes	Publisher

Transformation 1:

2. Before performing joins, check for duplicates in data in each data source and clean them individually.
3. Check for duplicates in joined dataset (if any)
4. Sort the data based on columns – "app_name" and "minutes".

*(**Sorting priority**: app_name – 1 and minutes – 2)*

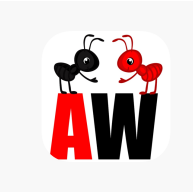
Transformation 2:

5. Create an intermediate file (lookup file) using group by on column 'app_name' and 'Publisher' take the mode across gender id. E.g.

	app_name	Publisher	gender_id
0	Amazon Mobile (Mobile App)	Amazon Sites	2.0
1	Amazon Music with Prime Music (Mobile App)	Amazon Sites	2.0
2	Facebook (Mobile App)	Facebook	2.0
3	Facebook Messenger (Mobile App)	Facebook	2.0
4	Google Play (Mobile App)	Google Sites	2.0

6. For any NaN values in the gender_id column, replace missing values using the table created from step 5.

Transformation 3:



7. Create another intermediate file (lookup file) using group by on column 'app_name' and 'Publisher' take the median across minutes column id. E.g.

	app_name	Publisher	minutes
0	Amazon Mobile (Mobile App)	Amazon Sites	47.1555
1	Amazon Music with Prime Music (Mobile App)	Amazon Sites	34.3980
2	Facebook (Mobile App)	Facebook	2764.6542
3	Facebook Messenger (Mobile App)	Facebook	1349.6490
4	Google Play (Mobile App)	Google Sites	57.6639

8. For any NaN values in the minute's column, replace missing values using the table created from step 7.

Transformation 4:

9. Filter the data across minutes column if the value is greater than (median + standard deviation) across app_name or minutes value id less than 30 mins

   (**hint**: perform group by over app_name to calculate median() and std())

Transformation 5:

10. Aggregate the from output to step 9 and create a statistical dataset (as per 10.1 and 10.2) with count, min, mean, Qaurtile1 (25 percentile), median, Qaurtile2 (75 percentile), standard deviation, and max values across App_Names, and across "app_name + gender_id"

The output should look like this:

10.1

	app_name	count	min	mean	Quartile1	median	Quartile3	std	max
0	Amazon Mobile (Mobile App)	830	30.0888	126.152832	55.95345	101.7198	168.72030	83.574544	358.8732
1	Amazon Music with Prime Music (Mobile App)	126	32.2812	155.959200	62.68185	107.1630	210.82950	119.192210	477.7164
2	Facebook (Mobile App)	1896	30.6180	2260.659742	675.60885	1948.5144	3628.09125	1730.470075	5919.2154
3	Facebook Messenger (Mobile App)	1905	30.6180	1327.781819	313.28640	1003.8546	2090.49120	1157.069162	4337.3610
4	Google Play (Mobile App)	1845	30.0132	132.946196	53.52480	94.3488	164.33550	109.616952	608.8446

10.2

app_name	gender_id	count	min	mean	median	std	max
Amazon Mobile (Mobile App)	**1**	305	30.2022	127.963782	108.03240	84.003641	358.8732
	2	525	30.0888	125.100756	97.33500	83.386443	358.4196
Amazon Music with Prime Music (Mobile App)	**1**	37	33.2262	169.330208	145.37880	115.209676	443.5830
	2	89	32.2812	150.400466	103.11840	121.011133	477.7164
Facebook (Mobile App)	**1**	712	34.9272	2292.127602	1981.34370	1750.601988	5914.7172
	2	1184	30.6180	2241.736503	1914.11640	1718.715840	5919.2154

Load the data created in step 10 into the target system. (export it to CSV file)

Transformation 6:

11. Create a new data frame with columns as app_name, “total_minutes”, “total devices”, “Average Minutes per Device Per App”. Use the below expression for creating this column:

Total min per App =df.groupby('app_name')['minutes'].sum()

Total device per App=df.groupby('app_name')['device_id'].count()

the output should look like this:

	app_name	total_minutes	total_devices	Avg_time_spend_per_device
0	Amazon Mobile (Mobile App)	1.047069e+05	830	126.152832
1	Amazon Music with Prime Music (Mobile App)	1.965086e+04	126	155.959200
2	Facebook (Mobile App)	4.286211e+06	1896	2260.659742
3	Facebook Messenger (Mobile App)	2.529424e+06	1905	1327.781819
4	Google Play (Mobile App)	2.452857e+05	1845	132.946196

Transformation 7:

12. Use data generated in step 11 to create a ranking system for Apps based on minutes spend over the app_name and rank based on total users on different apps and load the data into the target system (a separate file for each ranking system)

	app_name	total_minutes	total_devices	Avg_time_spend_per_device	Rank (Duration based)	Rank (user based)
3	Facebook Messenger (Mobile App)	2.529424e+06	1905	1327.781819	2.0	1.0
2	Facebook (Mobile App)	4.286211e+06	1896	2260.659742	1.0	2.0
4	Google Play (Mobile App)	2.452857e+05	1845	132.946196	16.0	3.0
20	YouTube (Mobile App)	9.013666e+05	1751	514.772468	4.0	4.0
5	Google Search (Mobile App)	6.081739e+05	1747	348.124726	8.0	5.0

13. Create and load files for Publisher Ranking as well. Data Format should be:

Publisher	total_devices	Rank
Google Sites	5343	1.0
Facebook	4890	2.0
Amazon Sites	1092	3.0
Snapchat, Inc	820	4.0

Loading the Target files:

Load the following data into the Target system (Feel free to create database connections using pandas to load data into different tables – Not Mandatory)

Target system A: (Structured tabulated data)

a. Output for step 10
b. Output from step 12 (2 different files for 2 different ranking systems with Publisher Name)
c. Output from step 13

Target System B: (Graphs)

a. Pie chart to show avg time spend per device per app
b. Plot to show users across Apps across gender.