Individual/Group HW 1 (Peer-review Policy applies)

Digital Marketing Analytics

2019 Part-time

(NO extensions---please start as soon as possible)

This HW constitutes 20 % of your course grade.

You can consult freely (internet, Help, tutorials) for all programming but you should do the work by yourself.

See the Instructions for many helpful tips.

INDIVIDUAL --- This part is individual so you all start off with familiarity with the data individually. This accounts for 5%. I strongly recommend you accomplish this task as quickly as possible.

1. (10 points) Create a database and load the data files given for the group HW to your database---*this may be more troublesome than it sounds because of the .csv format, and dates, so start early. If you are having trouble loading with SQL, try the pandas routine, but do use SQL for the queries.*
   Submission (in a pdf or a text file):
   - a query (query only) to identify the top 5 customer locations by average spend.
   - If you are using pandas the Python source code you used
   - any auxiliary queries/code that you did to load data, clean up data

   (read the description to find the field where this information lies; check the format or put it in the right format; decide what to do with missing data; join tables appropriately to extract)

GROUP---This part is group. It accounts for 15 % of the grade

2. The data set represents multi-channel sales campaigns and sales of a gifts company.

In this homework you will analyse the data set to gain "insights" into the effectiveness of the various direct-marketing channels---specifically catalog mailing vs. email. This is an open-ended HW so no ``right" answer, but you are expected to do substantial work. The grade depends on both effort and scope of your work as well as depth.

A. (5 points) Raise **two** interesting analytical questions you can ask the data, and study and analyse the data to give answers to your questions. Sample question: which channel has better response rates, catalog mailing or email?

Split the data into two parts (randomly, or by time, according to the question you are studying) to validate all your data-based targeting recommendations below. The point of this splitting is to mimic a sample mailing (which realistically we cannot do in class)

B. (10 points) Segment using the RFM dimensions (5 quantiles for each dimension). Estimate response rates for each RFM cell. Make a decision on how many to mail. Validate on a sample mailing (i.e. pretend the second part of the data you set aside is a sample mailing) to calculate ROI of the campaign (say it costs a normalized $1 per mailing and an average profit of $30 per purchase).
C. (5 points) If you can use another dimension (one or more) to target, which ones will you use? Justify based on (a) business/common-sense (b) data and a validation sample.

I could have given you cleaned up data---but instead am giving you raw data plus instructions, so you get practice in data wrangling and also the SQL skills you learned

Notes and instructions for 1:

.csv format loading into database tables is finicky and you may have trouble even loading the file into a database. Try

(a) *saving them as tab-delimited files (native format for postgresql) and then*
(b) *loading it after first creating tables (use the CREATE scripts available in pgAdmin after right-clicking on the table name). Read Postgresql Help on loading*
(c) *Check you are loading dates properly*

Options: COPY is fast and simple with only a few parameters, but is very finicky; another option is pandas read_csv (say with sqlalchemy). The latter has as many options as a Mercedes luxury car, so you will spend equal time with either one.

Other ways, such as reading line by line, or with R could turn out to be way too slow.

You can take either strategy (for the loading part), but do the queries using SQL as, if you take some care in indexing and writing the queries, significantly faster than all other methods.

If you are using the COPY command, you will face the problem of creating the table. What I do is take the header, copy into Excel and transpose it and then format it into a CREATE command with the fields to create the table. Then use COPY.

If there are 100+ fields as in the Summary table (you only really need the Orders table for RFM, but in case you have to…). One option is to create the table using pandas and import only a line, and then use the COPY command if the data is too big and pandas turns out to be slow.

Now, once you have the table and columns set up, as I mentioned, csv files might be troublesome to load. Convert to tab-delimited first to make your life easier.

Here is a sample loading script

Eg query to load in SQL (the file `cleaned2.txt` in the query is in tab-delimited format)

```
COPY
dataset9_cusorders(custid,ordernum,orderdate,linedollars,gift,recipnum)
FROM  'F:\\classes\\data sets\\Data set 9\\DMEFcustomer orders
cleaned2.txt' NULL ' '
```

in pandas

```
from sqlalchemy import create_engine
import pandas as pd
file='F:\\classes\\data sets\\marketing edge datasets\\Data set
9\\DMEFExtractSummaryV01.CSV'
```

```
engine =
create_engine('postgresql://MyDBServer:PasswordToMyDB@localhost/digital
class')
df = pd.read_csv(file,nrows-=2 )
df.to_sql('dataset9_summary', engine)
```

You may still have to work to get it going…

Notes and instructions for 2:

1. Please use SQL on Postgresql at least for storage and to do joins or filters across the four tables. Some parts you can probably do easier with pandas (say loading data), but I insist on SQL for the main queries and for practice.
2. Try to use the SQL aggregate and window functions rather than `pandas` for summarizing data– they are more low-level but more compact. Hint: Look into `ntile`.
   On the other hand, if you are more comfortable with pandas, you are free to use pandas instead.

3. Index and link your tables for faster queries and analysis.