sales-dashboard

July 27, 2024

Sales Dashboard with Tableau

Link to the Tableau Public for Dashboard- https://public.tableau.com/app/profile/abha6243/viz/SalesDashboard_

Problem statement: Create an insightful, intuitive dashboard visualizing the various aspects of sales data from an e-commerce company. You have to delve into the sales data of an e-commerce company, with information about the transactions, the products and the customer demographics and present useful insights with visualizations.

We are using Jupyter Notebook for rough data exploration and pre-processing before importing them to Tableau for dashboard creation. Following are some assumptions and conversions we made for a better understanding of the data:

1. Converted the ‘DAY’ column (Transaction.csv) using the calculated field in the tableau, considering the starting date as ‘01-01-2020’
2. Replace ” with the ‘Unknown’ placeholder in CURR_SIZE_OF_PRODUCT(product.csv), as this seems like the most suitable option for the current scenario.
3. Create time categories like dawn, morning, afternoon, and night to find time trends during the day.
4. Created revenue KPIs

[2]:
```python
#Importing libraries
import numpy as np
import pandas as pd
```

DEMOGRAPHIC DATASET

[117]:
```python
df_demographic = pd.read_csv("hh_demographic.csv")
```

[118]:
```python
df_demographic.shape
```

[118]:
```
(801, 8)
```

[119]:
```python
#columns present in the data
df_demographic.columns
```

[119]:
```
Index(['AGE_DESC', 'MARITAL_STATUS_CODE', 'INCOME_DESC', 'HOMEOWNER_DESC',
       'HH_COMP_DESC', 'HOUSEHOLD_SIZE_DESC', 'KID_CATEGORY_DESC',
       'household_key'],
      dtype='object')
```

[120]:
```
df_demographic.dtypes
```

[120]:
```
AGE_DESC               object
MARITAL_STATUS_CODE    object
INCOME_DESC            object
HOMEOWNER_DESC         object
HH_COMP_DESC           object
HOUSEHOLD_SIZE_DESC    object
KID_CATEGORY_DESC      object
household_key           int64
dtype: object
```

[121]:
```
df_demographic.head()
```

[121]:
```
  AGE_DESC MARITAL_STATUS_CODE INCOME_DESC HOMEOWNER_DESC      HH_COMP_DESC  \
0      65+                   A      35-49K      Homeowner  2 Adults No Kids
1    45-54                   A      50-74K      Homeowner  2 Adults No Kids
2    25-34                   U      25-34K        Unknown     2 Adults Kids
3    25-34                   U      75-99K      Homeowner     2 Adults Kids
4    45-54                   B      50-74K      Homeowner     Single Female

  HOUSEHOLD_SIZE_DESC KID_CATEGORY_DESC  household_key
0                   2      None/Unknown              1
1                   2      None/Unknown              7
2                   3                 1              8
3                   4                 2             13
4                   1      None/Unknown             16
```

[122]:
```
df_demographic.tail()
```

[122]:
```
    AGE_DESC MARITAL_STATUS_CODE INCOME_DESC HOMEOWNER_DESC      HH_COMP_DESC  \
796    35-44                   U      50-74K      Homeowner  2 Adults No Kids
797    45-54                   A      75-99K      Homeowner           Unknown
798    45-54                   U      35-49K        Unknown       Single Male
799    25-34                   U      50-74K      Homeowner  2 Adults No Kids
800    25-34                   U   Under 15K        Unknown     2 Adults Kids

    HOUSEHOLD_SIZE_DESC KID_CATEGORY_DESC  household_key
796                   2      None/Unknown           2494
797                   3                 1           2496
798                   1      None/Unknown           2497
799                   2      None/Unknown           2498
800                   3                 1           2499
```

[124]:
```
# Is there any missing value in the dataset?
df_demographic.isna().sum()
# np.any(df_demographic.isna())
```

[124]:

```
AGE_DESC                0
MARITAL_STATUS_CODE     0
INCOME_DESC             0
HOMEOWNER_DESC          0
HH_COMP_DESC            0
HOUSEHOLD_SIZE_DESC     0
KID_CATEGORY_DESC       0
household_key           0
dtype: int64
```

[15]:

```
# Is there any duplicate value in the dataset ?
np.any(df_demographic.duplicated())
```

[15]:

```
False
```

[16]:

```
# Basic information about the dataset
df_demographic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 801 entries, 0 to 800
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   AGE_DESC             801 non-null    object
 1   MARITAL_STATUS_CODE  801 non-null    object
 2   INCOME_DESC          801 non-null    object
 3   HOMEOWNER_DESC       801 non-null    object
 4   HH_COMP_DESC         801 non-null    object
 5   HOUSEHOLD_SIZE_DESC  801 non-null    object
 6   KID_CATEGORY_DESC    801 non-null    object
 7   household_key        801 non-null    int64
dtypes: int64(1), object(7)
memory usage: 50.2+ KB
```

[17]:

```
# Basic statistical description of the dataset
# since most columns are object so description of columns with 'object' datatype
df_demographic.describe(include = 'object')
# df_demographic.nunique()
```

[17]:

```
       AGE_DESC MARITAL_STATUS_CODE INCOME_DESC HOMEOWNER_DESC  \
count       801                 801         801            801
unique        6                   3          12              5
top       45-54                   U      50-74K      Homeowner
freq        288                 344         192            504

            HH_COMP_DESC HOUSEHOLD_SIZE_DESC KID_CATEGORY_DESC
count                801                 801               801
```

```
unique                   6                    5                 4
top      2 Adults No Kids                    2      None/Unknown
freq                   255                  318               558
```

'AGE_DESC' - estimated age range 'MARITAL_STATUS_CODE' - marital status (A-MARRIED,B-SINGLE,U-UNKNOWN) 'INCOME_DESC' - Household income 'HOMEOWNER_DESC' - homeowner, renter 'HH_COMP_DESC' - household composition 'HOUSEHOLD_SIZE_DESC' - size of household upto 5+ 'KID_CATEGORY_DESC' - number of children present upto 3+ 'household_key' - unique for each household

[127]:
```
# # Total number of transactions made by each age_desc
# np.round(df_demographic['AGE_DESC'].value_counts(normalize = True) * 100, 2)
# # np.round(df_demographic['AGE_DESC'].value_counts(normalize = True) * 100,␣
 ↪2).cumsum()
```

PRODUCT DATASET

[135]:
```
df_product= pd.read_csv("product.csv")
```

[136]:
```
df_product.head(10)
```

[136]:
```
   PRODUCT_ID  MANUFACTURER    DEPARTMENT     BRAND             COMMODITY_DESC  \
0       25671             2       GROCERY  National                   FRZN ICE
1       26081             2  MISC. TRANS.  National  NO COMMODITY DESCRIPTION
2       26093            69        PASTRY   Private                      BREAD
3       26190            69       GROCERY   Private       FRUIT - SHELF STABLE
4       26355            69       GROCERY   Private              COOKIES/CONES
5       26426            69       GROCERY   Private           SPICES & EXTRACTS
6       26540            69       GROCERY   Private              COOKIES/CONES
7       26601            69       DRUG GM   Private                   VITAMINS
8       26636            69        PASTRY   Private          BREAKFAST SWEETS
9       26691            16       GROCERY   Private         PNT BTR/JELLY/JAMS

            SUB_COMMODITY_DESC CURR_SIZE_OF_PRODUCT
0         ICE - CRUSHED/CUBED                22 LB
1  NO SUBCOMMODITY DESCRIPTION
2         BREAD:ITALIAN/FRENCH
3                  APPLE SAUCE                50 OZ
4             SPECIALTY COOKIES                14 OZ
5           SPICES & SEASONINGS               2.5 OZ
6  TRAY PACK/CHOC CHIP COOKIES                16 OZ
7             VITAMIN - MINERALS            300CT(1)
8          SW GDS: SW ROLLS/DAN
9                        HONEY                12 OZ
```

[139]:
```
df_product.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 92353 entries, 0 to 92352
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   PRODUCT_ID            92353 non-null  int64
 1   MANUFACTURER          92353 non-null  int64
 2   DEPARTMENT            92353 non-null  object
 3   BRAND                 92353 non-null  object
 4   COMMODITY_DESC        92353 non-null  object
 5   SUB_COMMODITY_DESC    92353 non-null  object
 6   CURR_SIZE_OF_PRODUCT  92353 non-null  object
dtypes: int64(2), object(5)
memory usage: 4.9+ MB
```

[140]:

```
df_product.describe(include = 'object')
```

[140]:

```
       DEPARTMENT     BRAND                  COMMODITY_DESC  \
count       92353     92353                           92353
unique         44         2                             308
top       GROCERY  National  GREETING CARDS/WRAP/PARTY SPLY
freq        39021     78537                            2785

       SUB_COMMODITY_DESC CURR_SIZE_OF_PRODUCT
count               92353                92353
unique               2383                 4345
top        CARDS EVERYDAY
freq                 1005                30607
```

[138]:

```
# Is there any missing value in the dataset?
df_product.isna().sum()
# np.any(df_product.isna())
```

[138]:

```
PRODUCT_ID              0
MANUFACTURER            0
DEPARTMENT              0
BRAND                   0
COMMODITY_DESC          0
SUB_COMMODITY_DESC      0
CURR_SIZE_OF_PRODUCT    0
dtype: int64
```

[ ]:

```
#we found that in 'CURR_SIZE_OF_PRODUCT' values have whitespace, so below we␣
 ↪are replacing them with 'Unknown' placeholder.
```

[80]:

```
df_product_or = pd.read_csv("product.csv")
```

```
[83]: #Replaced N/A values with unknown placeholder
      df_product_or['CURR_SIZE_OF_PRODUCT'] = df_product_or['CURR_SIZE_OF_PRODUCT'].
       ↪str.strip().replace('', 'Unknown')
```

```
[141]: df_product_or.head()
```

```
[141]:    PRODUCT_ID  MANUFACTURER    DEPARTMENT     BRAND            COMMODITY_DESC  \
       0       25671             2       GROCERY  National                  FRZN ICE
       1       26081             2  MISC. TRANS.  National  NO COMMODITY DESCRIPTION
       2       26093            69        PASTRY   Private                     BREAD
       3       26190            69       GROCERY   Private      FRUIT - SHELF STABLE
       4       26355            69       GROCERY   Private             COOKIES/CONES

                   SUB_COMMODITY_DESC CURR_SIZE_OF_PRODUCT
       0             ICE - CRUSHED/CUBED                22 LB
       1  NO SUBCOMMODITY DESCRIPTION              Unknown
       2           BREAD:ITALIAN/FRENCH              Unknown
       3                    APPLE SAUCE                50 OZ
       4              SPECIALTY COOKIES                14 OZ
```

```
[ ]: #download the file to use in tableau
     output_file_path = '/mnt/data/cleaned_product_sizes.csv'
     df_product1.to_csv(output_file_path, index=False)
```

```
[ ]:
```

TRANSACTION DATA

```
[26]: df_transaction = pd.read_csv("transaction_data.csv")
```

```
[95]: df_transaction.shape
```

```
[95]: (1298486, 13)
```

```
[28]: df_transaction.head()
```

```
[28]:    Unnamed: 0  household_key    BASKET_ID  DAY  PRODUCT_ID  QUANTITY  \
      0         863              1  42229641145  706     5585510         1
      1         864              1  33658816354  452    12330539         1
      2         865              1  30578772112  235     1082185         1
      3         866              1  34010015588  472      952924         1
      4         867              1  33065796045  412      965395         3

         SALES_VALUE  STORE_ID  RETAIL_DISC  TRANS_TIME  WEEK_NO  COUPON_DISC  \
      0         2.49       436         0.00        1531      102          0.0
      1         1.25       436        -0.54        1417       65          0.0
      2         0.87       436        -0.60        1705       34          0.0
```

```
3          2.79       436          0.00         1125        68           -1.0
4          2.07       436         -0.90         1408        60            0.0

   COUPON_MATCH_DISC
0                0.0
1                0.0
2                0.0
3                0.0
4                0.0
```

```
[29]: df_transaction.tail()
```

```
[29]:          Unnamed: 0  household_key    BASKET_ID  DAY  PRODUCT_ID  QUANTITY  \
1298481         1498           2500  28057122795   90     6514008         1
1298482         1499           2500  41599946638  659      997479         1
1298483         1500           2500  34850276975  507      859237         1
1298484         1501           2500  41493000826  651     1087167         1
1298485         1502           2500  41297427214  636      948670         2

         SALES_VALUE  STORE_ID  RETAIL_DISC  TRANS_TIME  WEEK_NO  COUPON_DISC  \
1298481         4.99       327         -0.4        1946       14          0.0
1298482         1.99       447          0.0        1815       95          0.0
1298483         1.29       330          0.0        1542       73          0.0
1298484         3.29       327          0.0        1234       94          0.0
1298485        15.60       447          0.0        1910       92          0.0

         COUPON_MATCH_DISC
1298481                0.0
1298482                0.0
1298483                0.0
1298484                0.0
1298485                0.0
```

```
[142]: df_transaction.columns
```

```
[142]: Index(['Unnamed: 0', 'household_key', 'BASKET_ID', 'DAY', 'PRODUCT_ID',
              'QUANTITY', 'SALES_VALUE', 'STORE_ID', 'RETAIL_DISC', 'TRANS_TIME',
              'WEEK_NO', 'COUPON_DISC', 'COUPON_MATCH_DISC'],
             dtype='object')
```

RECOMDENDATIONS

1. Targeted marketing: Since the majority of transactions are made by homeowners, people falling into the age group of 35–54, having a household size of 1 or 2, having an income of 35K–74K, and during night time of the day, it would be beneficial to tailor marketing strategies to cater to their preferences and needs. This could include specific promotions, product offerings, or advertising campaigns designed to attract these customers.

2. Engage with new homeowners: As a significant portion of sales transactions (67.52%) come from customers who are homewoners, it presents an opportunity to engage with them. Targeted marketing, welcoming offers, and incentives for them can help capture their loyalty and increase their spending.

3. Time-based marketing: With a significant number of sales(83.87%) done during the night time, tailoring marketing strategies to target night time can lead to better results. Allocating resources, promotions, and events based on the customer concentration in each city can help drive sales.

4. Emphasise popular product departments: Since the majority of sales transactions are concentrated in just five product departments (grocery, drugGM, KIOSK Gas, product, and meat), allocating resources and promotions towards these categories can maximise sales potential. Highlighting these popular departments and offering attractive deals can encourage more purchases. Ensuring a wide range of options and competitive pricing, can capitalise on customer demand and drive overallsales.

5. Increase focus on married customers: Given that 46.6% of total revenue is generated by married customers, dedicating efforts to cater to their needs and preferences can help drive more sales. Understanding their motivations and targeting them with personalised offers can enhance their shopping experience and loyalty.

6. Optimise revenue from specific age groups: Since a majority of transactions are made by customers between the ages of 35 and 54. It is important to focus marketing efforts on this demographic. Offering products and services that align with their interests and values can maximise revenue generation.

7. Personalised offers for high spenders: identifying customers with high total spending, such as homeowners or customers of a specific age groups, allows for targeted marketing and personalised offers. Providing exclusive discounts, loyalty rewards, or special privileges toThese customers can encourage repeat purchases and increase customer satisfaction.

8. Implement a loyalty programmeme: Implement a loyalty programmeme that offers incentives, rewards, and exclusive deals to encouragerepeat purchases and increase customer retention. Targeted loyalty programmes can be designed for homeowners, married customers,and customers in specific age groups.