Study of Sales through Customer Behaviour

MOTIVATION

- Black Friday! An extravaganza of sales, promotions, and long lines outside of stores.
- Retailers (such as Target, Best Buy, Amazon, and many others) hopes that consumers will take advantage of door-busting deals.
- But, How ?? how the retailers provide us with such good deals ?

BLACK FRIDAY DATASET

Purchase summary of various customers for selected high volume products of a retail company "**ABC Private Limited**" with respect to customer demographics.

OBJECTIVE

- Predicting the amount of purchase of a customer on black friday in order to provide promotional offers for a retail store ABC Pvt Ltd. ?
- Developing customer segments to define marketing strategy based on their spending habits ?
- Attempting to associate products in order to perform product recommendation and organize store layout.

DATA DESCRIPTION

Dimension : 538k x 12

User_ID	A number that uniquely identifies a Customer.
Product_ID	Unique code representing a product
Gender	Sex of the Customer. (M/F)
Age	Age of the Customer in bins. (All bins)
Occupation	ID number of occupation type of each customer ranging from 0-20(masked). Eg : 0=Doctor, 1=Engineer.....
City_Category	Category of the City representing which city the customer belongs to. (City A/ City B /City C)
Stay_In_Current_City_Years	Number of years of stay in current city. Since how many years the customer has been living in the current city. (0,1,2,3,4+)

DATA DESCRIPTION (Contd..)

Marital_Status	Whether customer is married or not. (1=Married, 0=Unmarried)
Product_Category_1	No. of subcategories to which the product belongs to in **Cloths** category. (18 means this product belongs to 18 different subcategories under Cloth)
Product_Category_2	No. of subcategories to which the product belongs to in **Electronics** category.
Product_Category_3	No. of subcategories to which the product belongs to in **Home Goods** category.
Purchase	Purchase amount.

NOTE : > A customer has bought multiple products and a product has been bought by multiple customers.
> A Product can belong to multiple subcategories. For example, running shoes belong to both fashion and sports categories.

DATA EXPLORATION

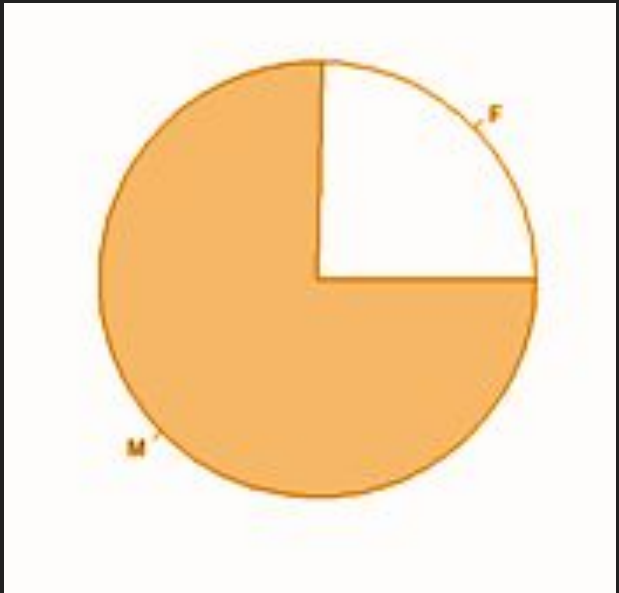
```
> summary(BlackF.raw.df)
    User_ID          Product_ID      Gender       Age          Occupation     City_Category Stay_In_Current_City_Years
 Min.   :1000001   P00265242:  1858   F:132197   0-17 : 14707   Min.   : 0.000   A:144638      0 : 72725
 1st Qu.:1001495   P00110742:  1591   M:405380   18-25: 97634   1st Qu.: 2.000   B:226493      1 :189192
 Median :1003031   P00025442:  1586              26-35:214690   Median : 7.000   C:166446      2 : 99459
 Mean   :1002992   P00112142:  1539              36-45:107499   Mean   : 8.083                 3 : 93312
 3rd Qu.:1004417   P00057642:  1430              46-50: 44526   3rd Qu.:14.000                 4+: 82889
 Max.   :1006040   P00184942:  1424              51-55: 37618   Max.   :20.000
                   (Other)   :528149              55+  : 20903
 Marital_Status   Product_Category_1 Product_Category_2 Product_Category_3    Purchase
 Min.   :0.0000   Min.   : 1.000    Min.   : 2.00      Min.   : 3.0       Min.   :  185
 1st Qu.:0.0000   1st Qu.: 1.000    1st Qu.: 5.00      1st Qu.: 9.0       1st Qu.: 5866
 Median :0.0000   Median : 5.000    Median : 9.00      Median :14.0       Median : 8062
 Mean   :0.4088   Mean   : 5.296    Mean   : 9.84      Mean   :12.7       Mean   : 9334
 3rd Qu.:1.0000   3rd Qu.: 8.000    3rd Qu.:15.00      3rd Qu.:16.0       3rd Qu.:12073
 Max.   :1.0000   Max.   :18.000    Max.   :18.00      Max.   :18.0       Max.   :23961
                                    NA's   :166986    NA's   :373299
```

- No outliers
- Missing values in Product_Category_2 & Product_Category_3

DATA EXPLORATION (Contd..)

- Analysis of variable - **GENDER**

CUSTOMER RATIO (Based on gender)

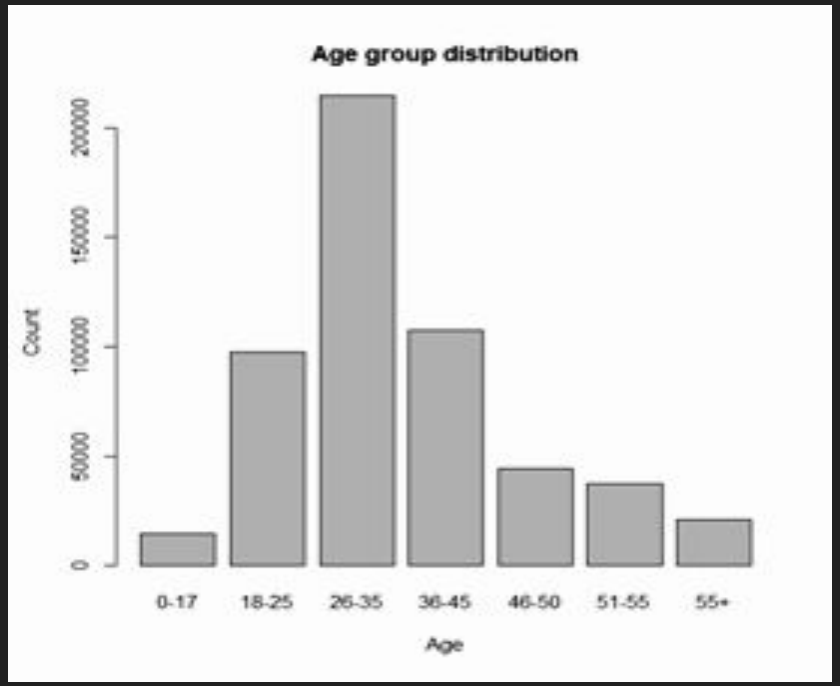


PURCHASE RATIO(Based on gender)

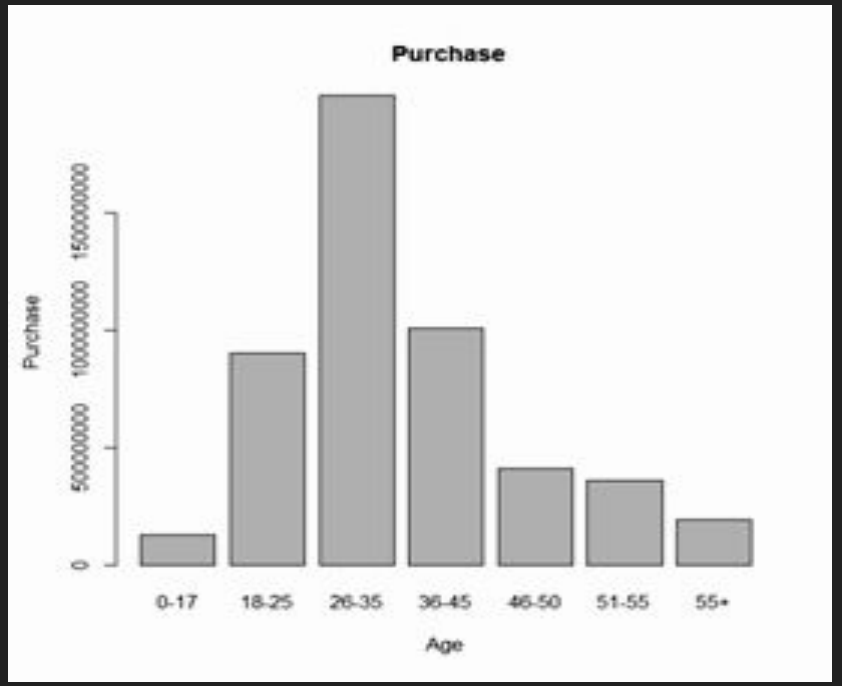
DATA EXPLORATION (Contd..)

- Analysis of variable - **AGE**

AGE GROUP DISTRIBUTION (Over Count)



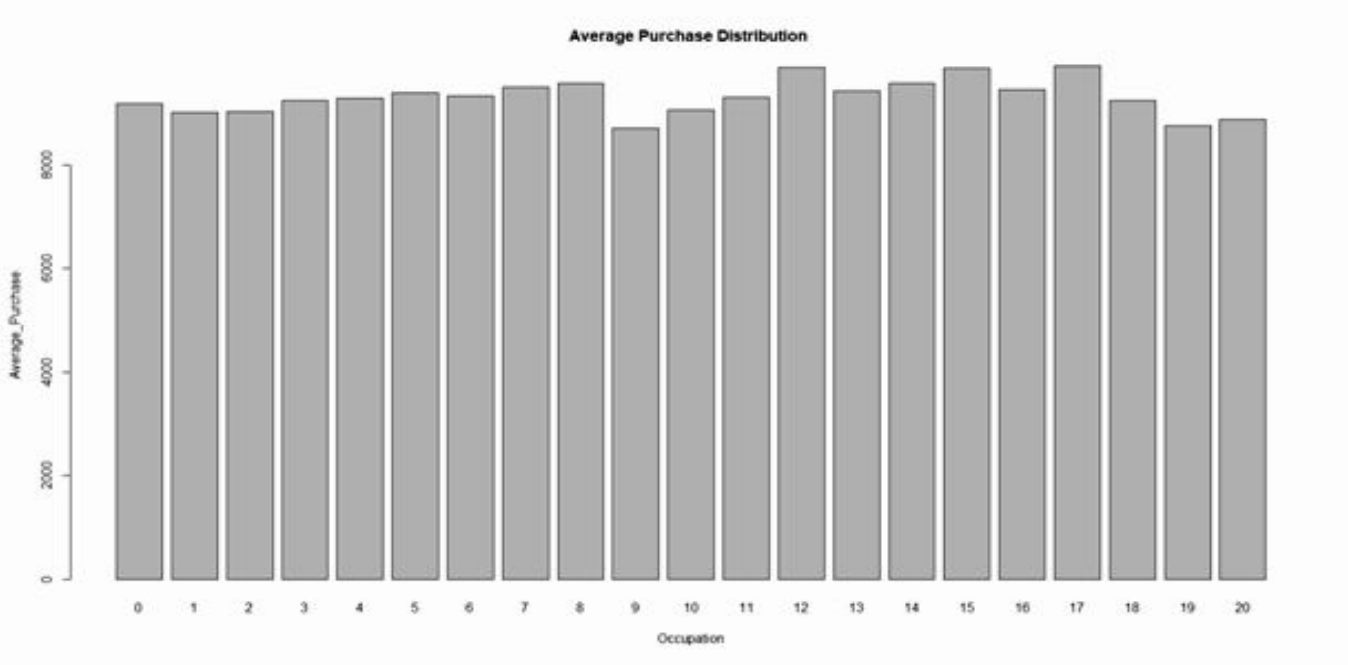
PURCHASE DISTRIBUTION



DATA EXPLORATION (Contd..)

- Analysis of variable - **OCCUPATION**

PURCHASE DISTRIBUTION

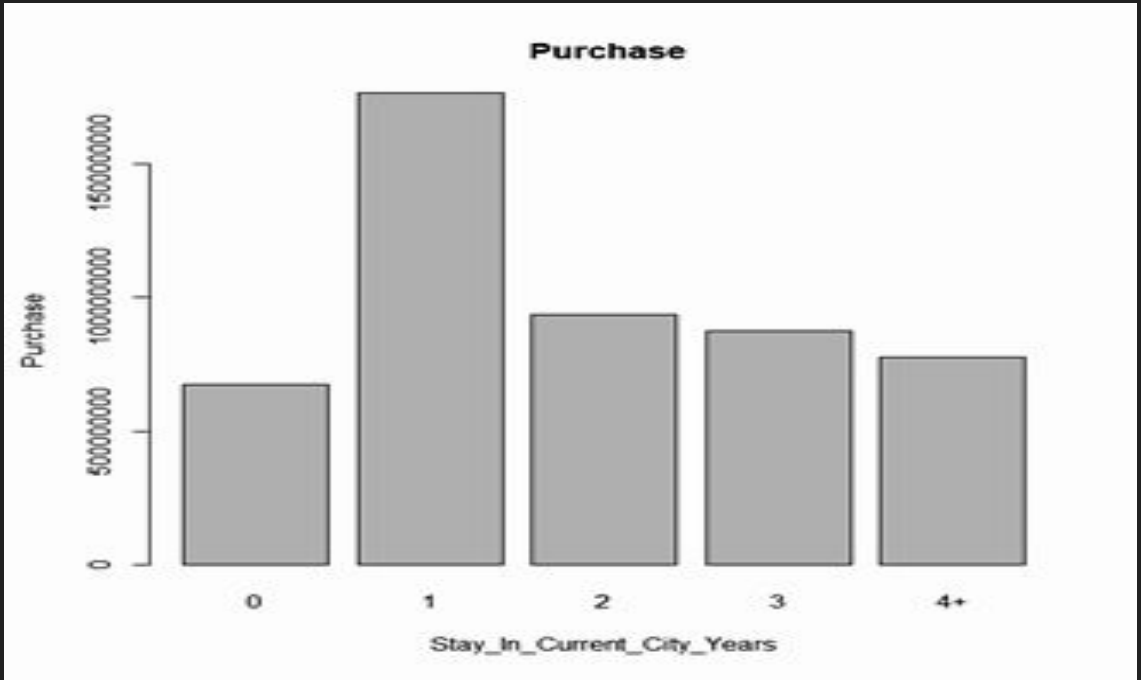


- Average is more / less same for all occupations.
- Masked Data

DATA EXPLORATION (Contd..)

- Analysis of variable - **STAY IN CURRENT CITY YEARS**

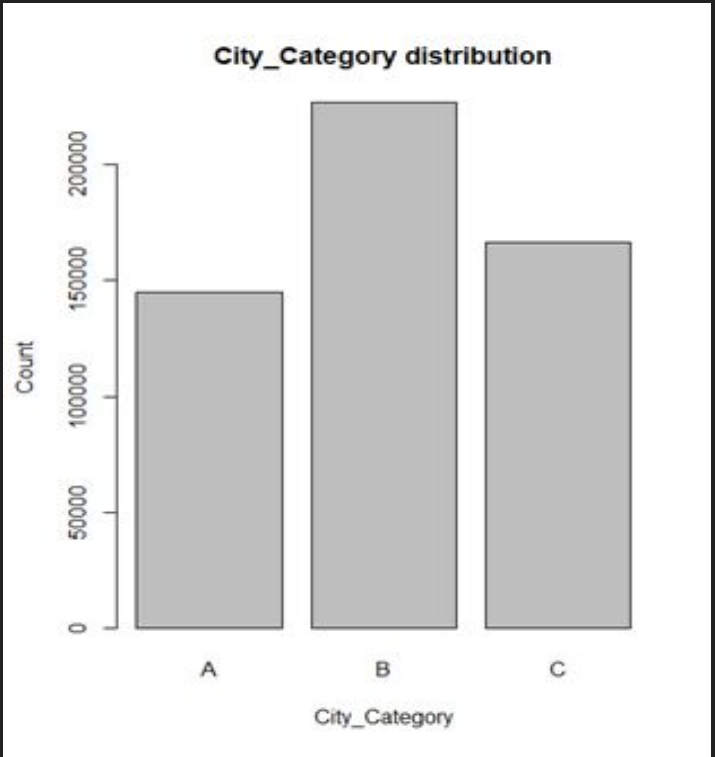
PURCHASE DISTRIBUTION



- New residents will take advantage of the low prices or offers on Black Friday to purchase all things needed.

DATA EXPLORATION (Contd..)

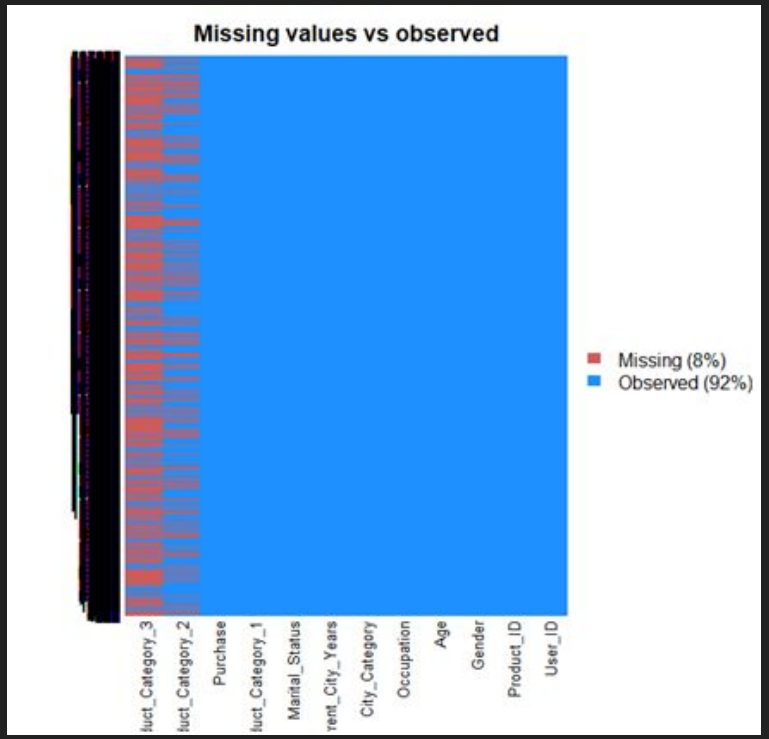
- Analysis of variable - **CITY CATEGORY**



- More shoppers from City B

DATA CLEANING & PRE-PROCESSING

- IMPUTING MISSING VALUES (Product Category 2 & Product Category 3)




```
> sapply(BlackF.raw.df,function(x) sum(is.na(x)))
                  User_ID                Product_ID                    Gender
                        0                         0                         0
                      Age                Occupation             City_Category
                        0                         0                         0
Stay_In_Current_City_Years            Marital_Status          Product_Category_1
                        0                         0                         0
        Product_Category_2          Product_Category_3                  Purchase
                        0                         0                         0
```

DATA CLEANING & PRE-PROCESSING(Cont..)

- REMOVING CUSTOMER ID AND PRODUCT ID

  - Unique ID's.
  - Not a customer purchase characteristic.

- REMOVING OCCUPATION

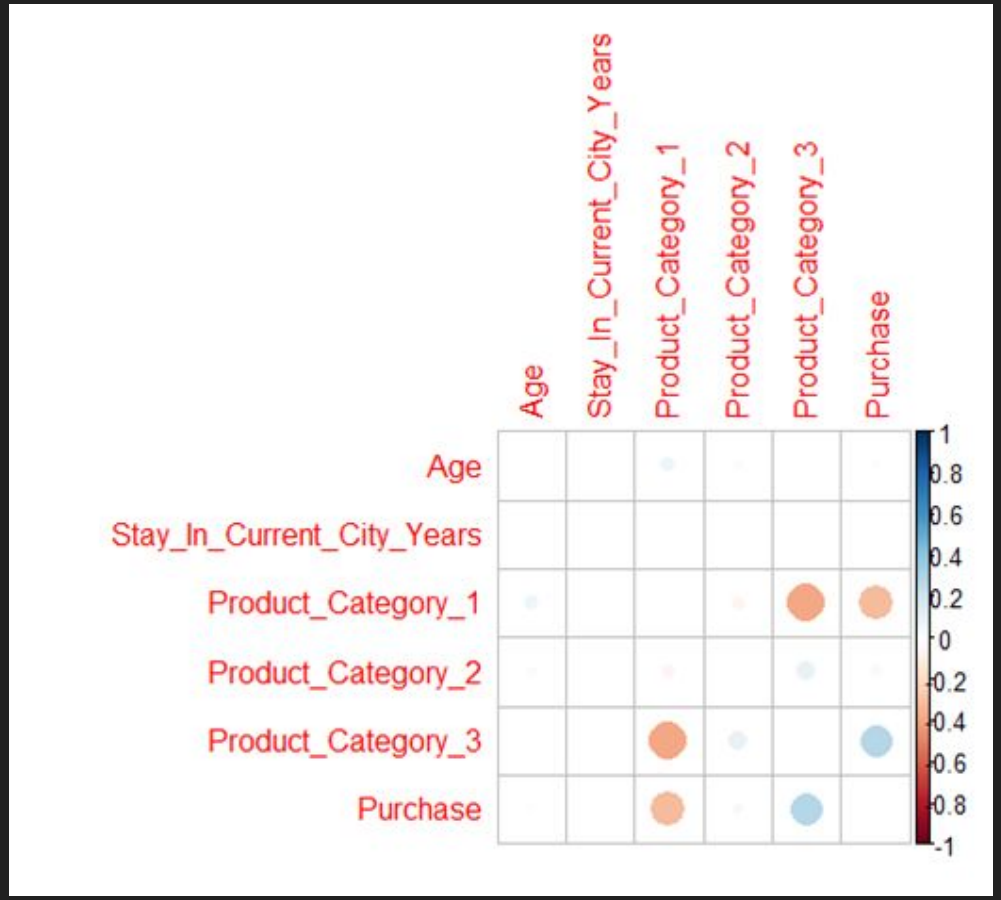
  - Average is more / less same for all occupations.
  - Masked Data
  - Doesn’t provide us much information regarding purchase amount

DATA CLEANING & PRE-PROCESSING(Cont..)

- ALTERING TYPE OF FEATURES

Age to Continuous	Marital status to factor	Stay in current city to continuous
- categorical variable with 7 factor levels. - Creating dummies we lose information - Mean of the range provided is taken. For example, for '0-17', we impute 8.5.	- categorical feature is identified as an integer value type (0/1). - Make it factor	- This feature can very well be continuous. (0,1,2,3,4+) - Encoded 4 in the place of 4+, interpret 4 as 4 or more years. (No negative impact)

RELATIONSHIP AMONG FEATURES



- Purchase highly correlated with :
  Product_Category_1
  Product_Category_3

- No multicollinearity

MODEL BUILDING

SUPERVISED LEARNING

MULTIPLE LINEAR REGRESSION MODEL

How can we predict the amount of purchase of a customer by analysing customer purchase behaviour in order to provide promotional offers?

MULTIPLE LINEAR REGRESSION MODEL :

- MODEL SPECIFIC PRE-PROCESSING :

  - Dummy Creation for categorical variables: Gender, City_Category , Marital Status
  - Data Partitioning (Train/Valid/Test)

- SELECTING BEST SUBSET OF PREDICTORS:

  - Used Exhaustive Search.
  - select the best subset of predictors based on adj R2 : 5,6,7 Predictors

```
> sum$adjr2
[1] 0.09870743 0.12954834 0.13147508 0.13307515 0.13501581 0.13584238 0.13593739 0.13593955 0.13593937
```

MULTIPLE LINEAR REGRESSION MODEL (Contd..):

- MODEL SELECTION (Based on Valid Data) :

- **MODEL 1**:

```
lm(formula = Purchase ~ Gender.F + City_Category.A + City_Category.B +
    Product_Category_1 + Product_Category_3, data = train.data)
```

RMSE: 4639.693

- **MODEL 2**:

```
lm(formula = Purchase ~ Gender.F + Age + City_Category.A + City_Category.B +
    Product_Category_1 + Product_Category_3, data = train.data)
```

RMSE: 4637.758

- **MODEL 3**:

```
lm(formula = Purchase ~ Gender.F + Age + City_Category.A + City_Category.B +
    Product_Category_1 + Product_Category_2 + Product_Category_3,
    data = train.data)
```

RMSE: 4637.501

MULTIPLE LINEAR REGRESSION MODEL (Contd..):

- BEST MODEL : **Model 3**
- MODEL APPLICATION ON TEST DATA (Unseen Data):

```
> accuracy(pred_test, test.data$Purchase)
                ME     RMSE      MAE       MPE     MAPE
Test set -23.52819 4630.611 3545.017 -47.44084 69.58209
```

- It turns out that our prediction accuracy, is even better than on validation data.
- So, we can conclude that our model will predict Consumer purchase amount for any new customer accurately.

MODEL BUILDING

UNSUPERVISED LEARNING

CLUSTERING USING K-PROTOTYPES

Can we develop a customer segmentation to define marketing strategy based on their spending habits?

WHY K-PROTOTYPE:

- **Hierarchical clustering** ❌
  - Gower ✓
  - Ward/Complete Linkage…❌ : Solution :- K-Medoid ⚠ (Memory Overflow)

- **Non- Hierarchical clustering** ✓
  - K- Means ❌ (Because of Euclidean distance)
  -  Solution -> **K- Prototype** :

    Mixed[Mean(Numerical features) + Mode(Categorical Features)]

CLUSTERING USING K PROTOTYPE:

- ENCODING THE FACTOR VARIABLES:

  - Gender into binary: 0- Female, 1-Male
  - City_Category as : 1 - City A, 2- City B, 3- City C
  - For easy calculation and interpretation

- SCALING DATA:

  - Normalizing Numerical data (Range of 0 to 1).
  - For distance calculation

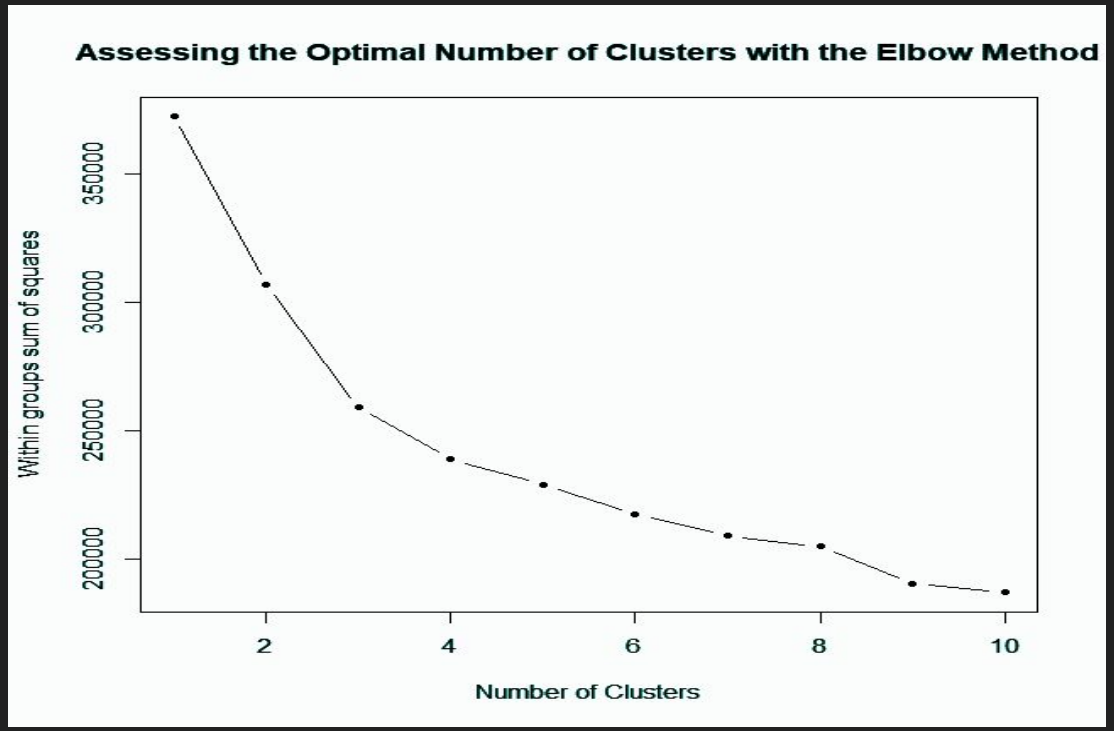
CLUSTERING USING K PROTOTYPE:

❑ APPLICATION OF K-PROTOTYPE :

- Library Required – clustMixType
- Check for the optimal number of clusters: Ran kproto for 10 clusters and stored their respective "withinss" to plot and identify the optimal number of clusters.
- Elbow method: We plotted the No. of clusters (1-10) against their respective withinss (within cluster distance) to identify the number of clusters where the withinss is minimum.
- Optimal k-Selection: Based on the above plot we can identify clearly that if we choose 4 clusters our distance within the clusters is minimum. So, our optimal k (no. of clusters) is 4.
- Creating Customer Clusters: applied k prototypes with optimal k (number of clusters) and assigned each transaction to a cluster.

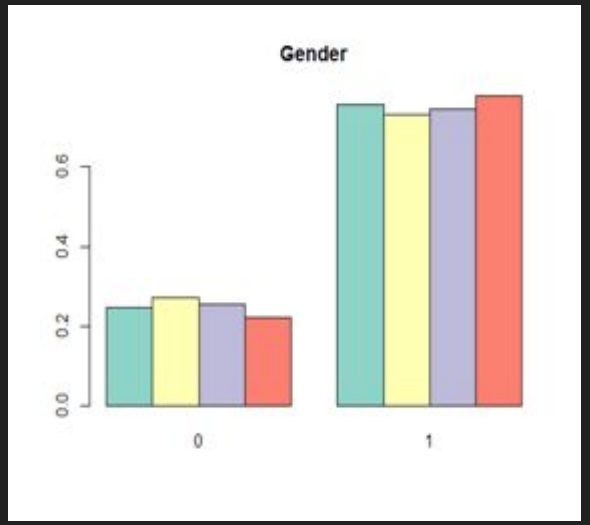
CLUSTERING USING K PROTOTYPE:

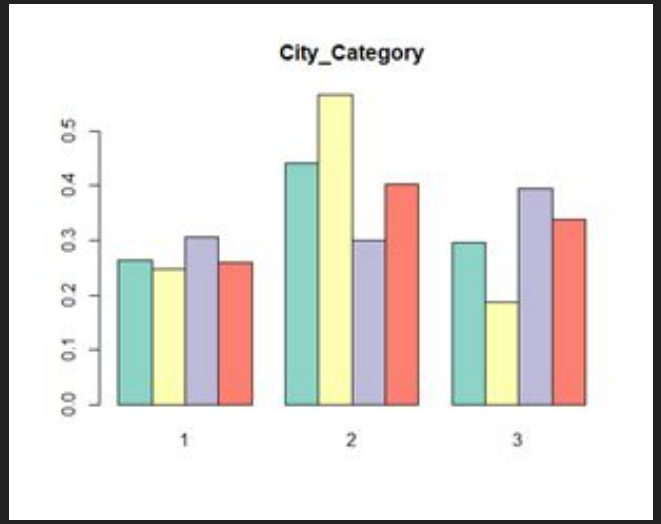
- PLOT :

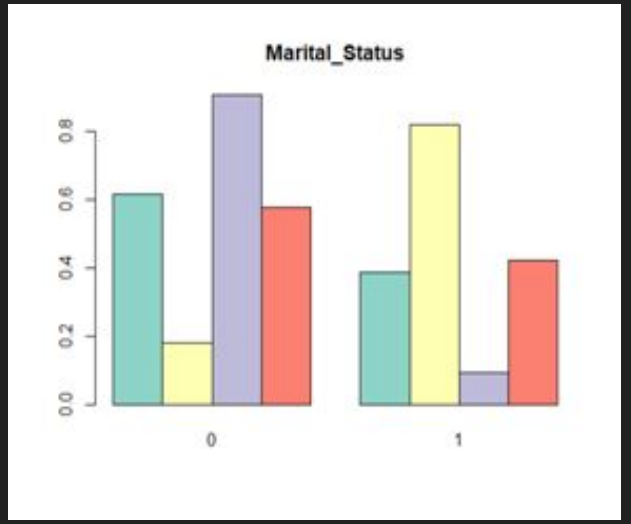


CLUSTERING USING K PROTOTYPE:

- RESULTS :
  - Visualizing cluster distribution of **categorical data.**

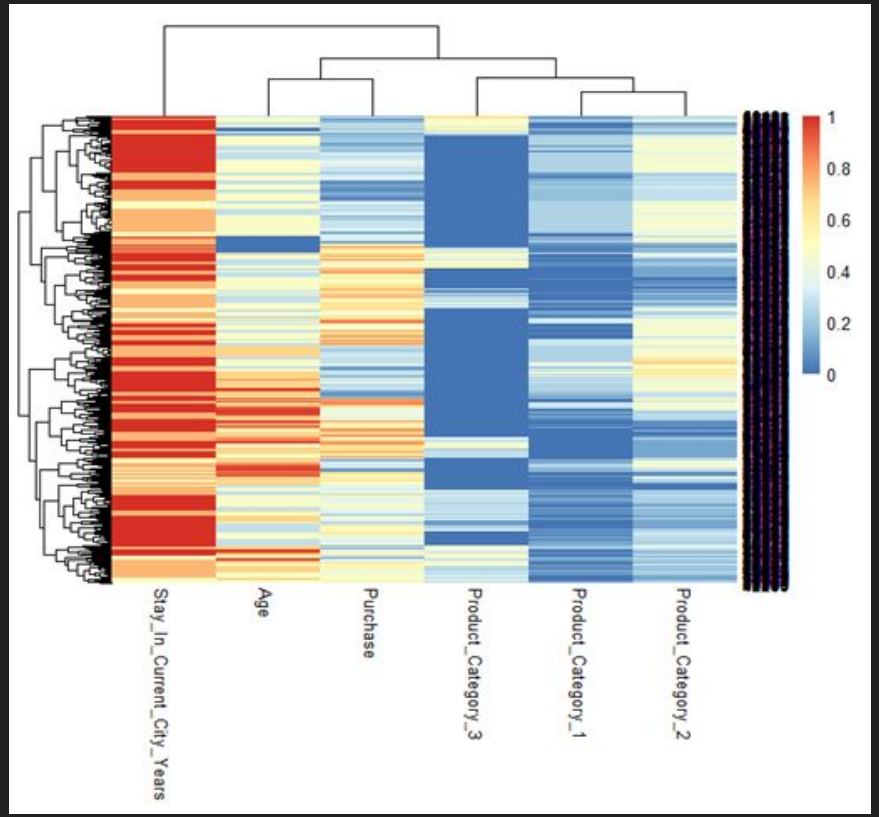


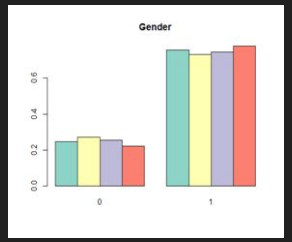


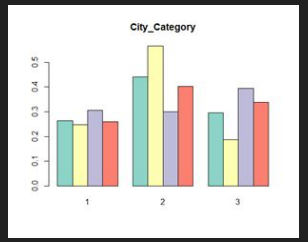


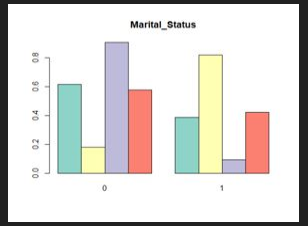
CLUSTERING USING K PROTOTYPE:

- Visualizing cluster distribution of **Numerical data.**





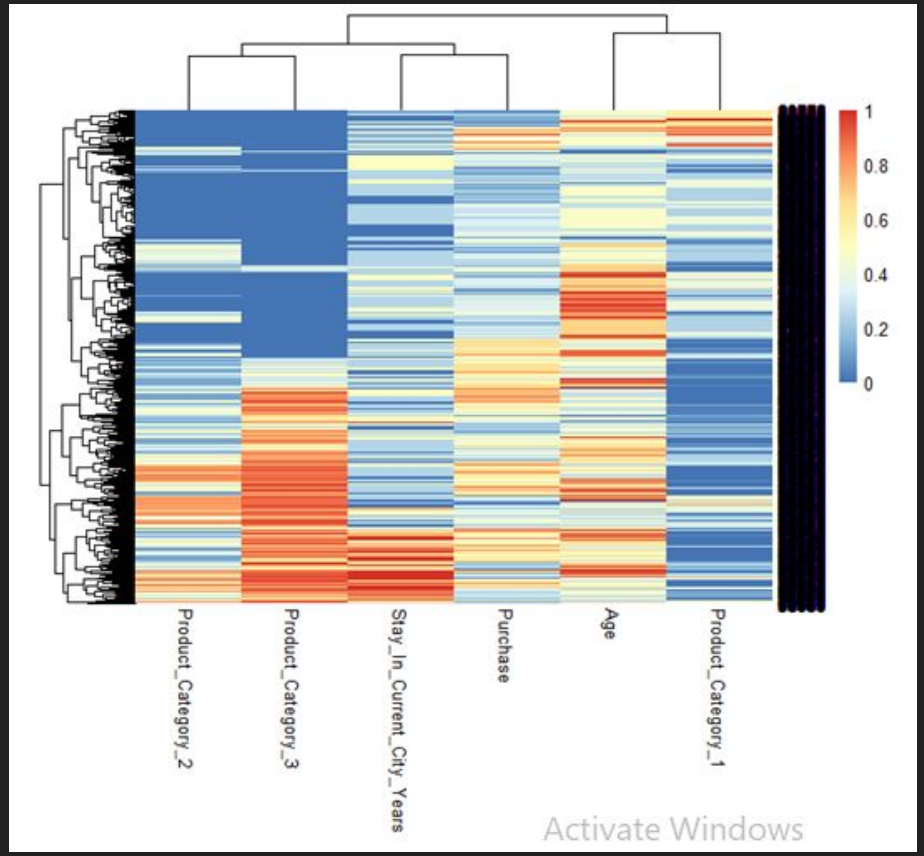


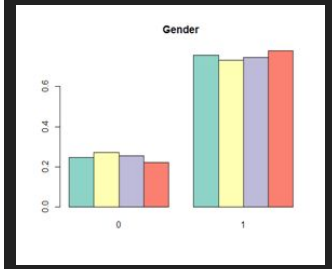


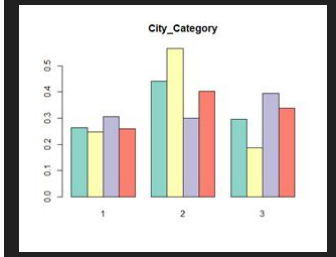
CLUSTER- 1

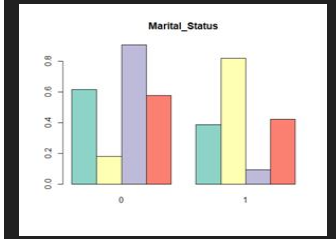
- Current City from long time.
- City A
- More Electronic Items purchase.
- More Unmarried People.
- More Female customers.

CLUSTERING USING K PROTOTYPE:





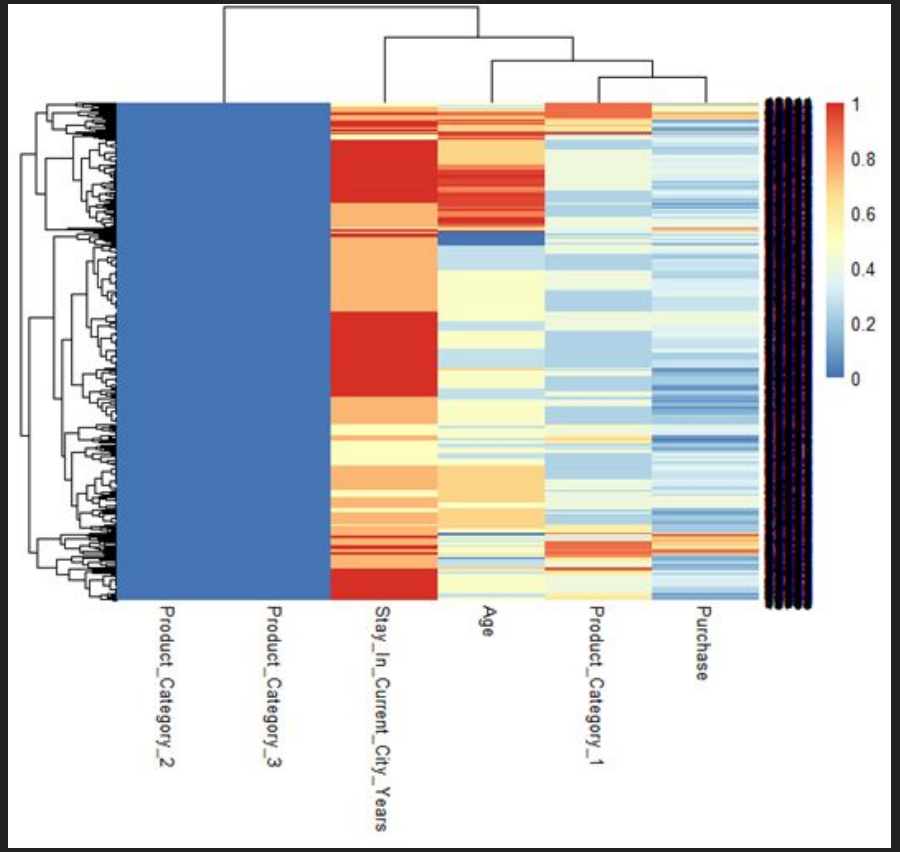


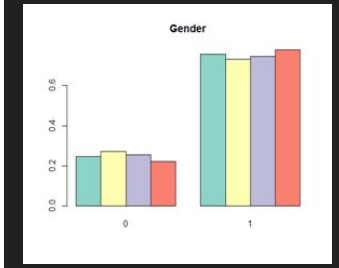


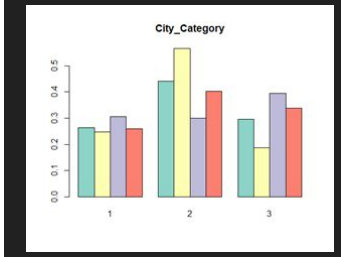
CLUSTER -2

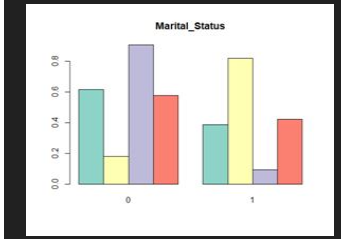
- Recently moved to their Current city.
  (OR) do not live in Current city.
- Purchase home goods.
- More married People.
- More male customers.

CLUSTERING USING K PROTOTYPE:





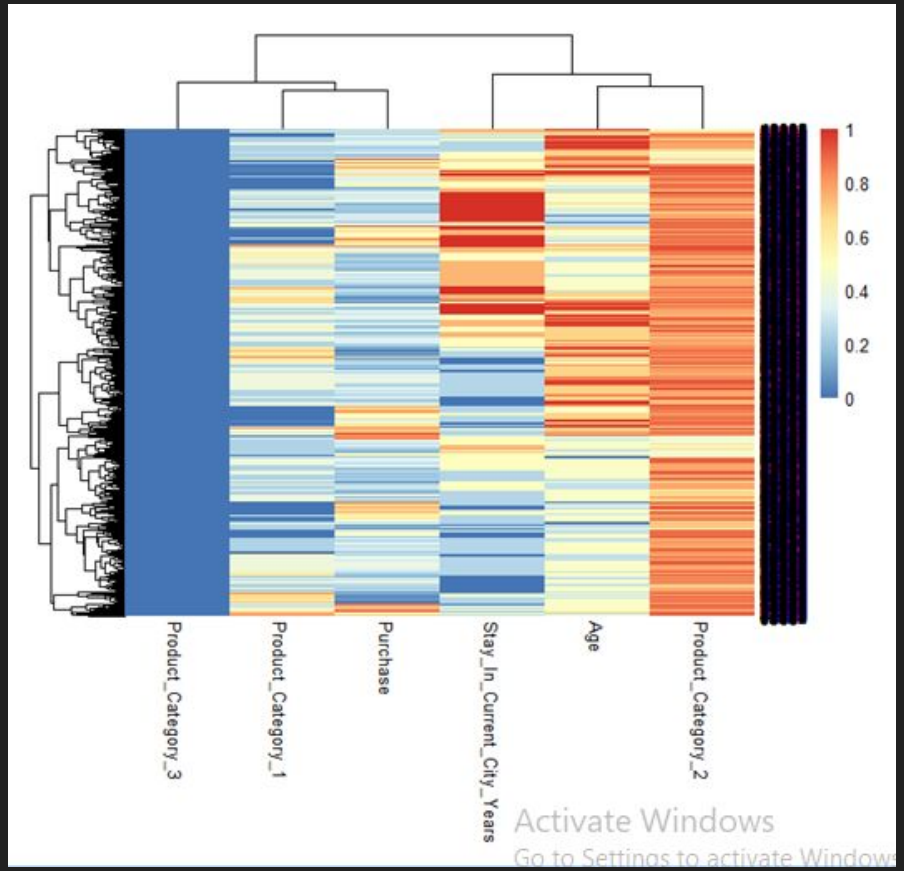


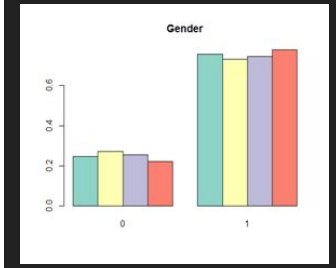


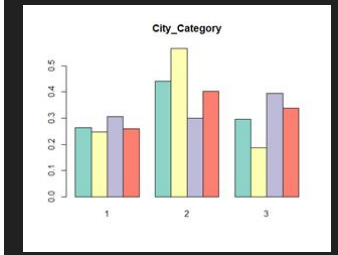
CLUSTER -3

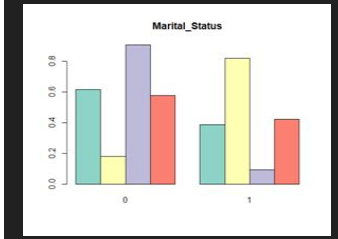
- Current City from long time.
- City C
- More Clothing Items purchase.
- More married People.
- More male customers.

CLUSTERING USING K PROTOTYPE:









CLUSTER -4

- live in Current city for quite few years.
- City B and C
- More purchase of Electronic Items along with Clothing.
- More Unmarried People.

MODEL BUILDING

UNSUPERVISED LEARNING

ASSOCIATION RULES (Apriori Algorithm)

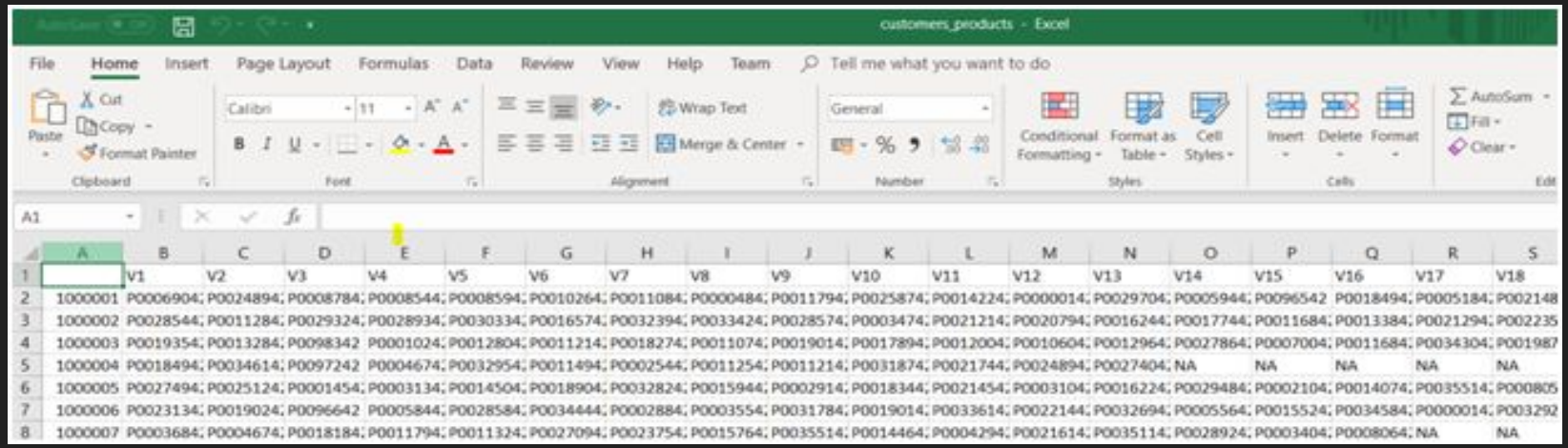
Can we attempt to associate products in order to perform product recommendation and organize store layout?

APRIORI ALGORITHM :

- MODEL SPECIFIC PRE-PROCESSING :

  - Changing the data format : *subset of the original data (User_ID and Product_ID)*

  - Customer product table creation. (Spread())



customers_products - Excel

File Home Insert Page Layout Formulas Data Review View Help Team Tell me what you want to do

Cut Copy Format Painter Paste Clipboard

Calibri 11 Font

Wrap Text Merge & Center Alignment

General Number

Conditional Formatting Format as Table Cell Styles Styles

Insert Delete Format Cells

AutoSum Fill Clear Edit

A1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18
2	1000001	P0006904	P0024894	P0008784	P0008544	P0008594	P0010264	P0011084	P0000484	P0011794	P0025874	P0014224	P0000014	P0029704	P0005944	P0096542	P0018494	P0005184	P002148
3	1000002	P0028544	P0011284	P0029324	P0028934	P0030334	P0016574	P0032394	P0033424	P0028574	P0003474	P0021214	P0020794	P0016244	P0017744	P0011684	P0013384	P0021294	P002235
4	1000003	P0019354	P0013284	P0098342	P0001024	P0012804	P0011214	P0018274	P0011074	P0019014	P0017894	P0012004	P0010604	P0012964	P0027864	P0007004	P0011684	P0034304	P001987
5	1000004	P0018494	P0034614	P0097242	P0004674	P0032954	P0011494	P0002544	P0011254	P0011214	P0031874	P0021744	P0024894	P0027404	NA	NA	NA	NA	NA
6	1000005	P0027494	P0025124	P0001454	P0003134	P0014504	P0018904	P0032824	P0015944	P0002914	P0018344	P0021454	P0003104	P0016224	P0029484	P0002104	P0014074	P0035514	P000805
7	1000006	P0023134	P0019024	P0096642	P0005844	P0028584	P0034444	P0002884	P0003554	P0031784	P0019014	P0033614	P0022144	P0032694	P0005564	P0015524	P0034584	P0000014	P003292
8	1000007	P0003684	P0004674	P0018184	P0011794	P0011324	P0027094	P0023754	P0015764	P0035514	P0014464	P0004294	P0021614	P0035114	P0028924	P0003404	P0008064	NA	NA

APRIORI ALGORITHM :

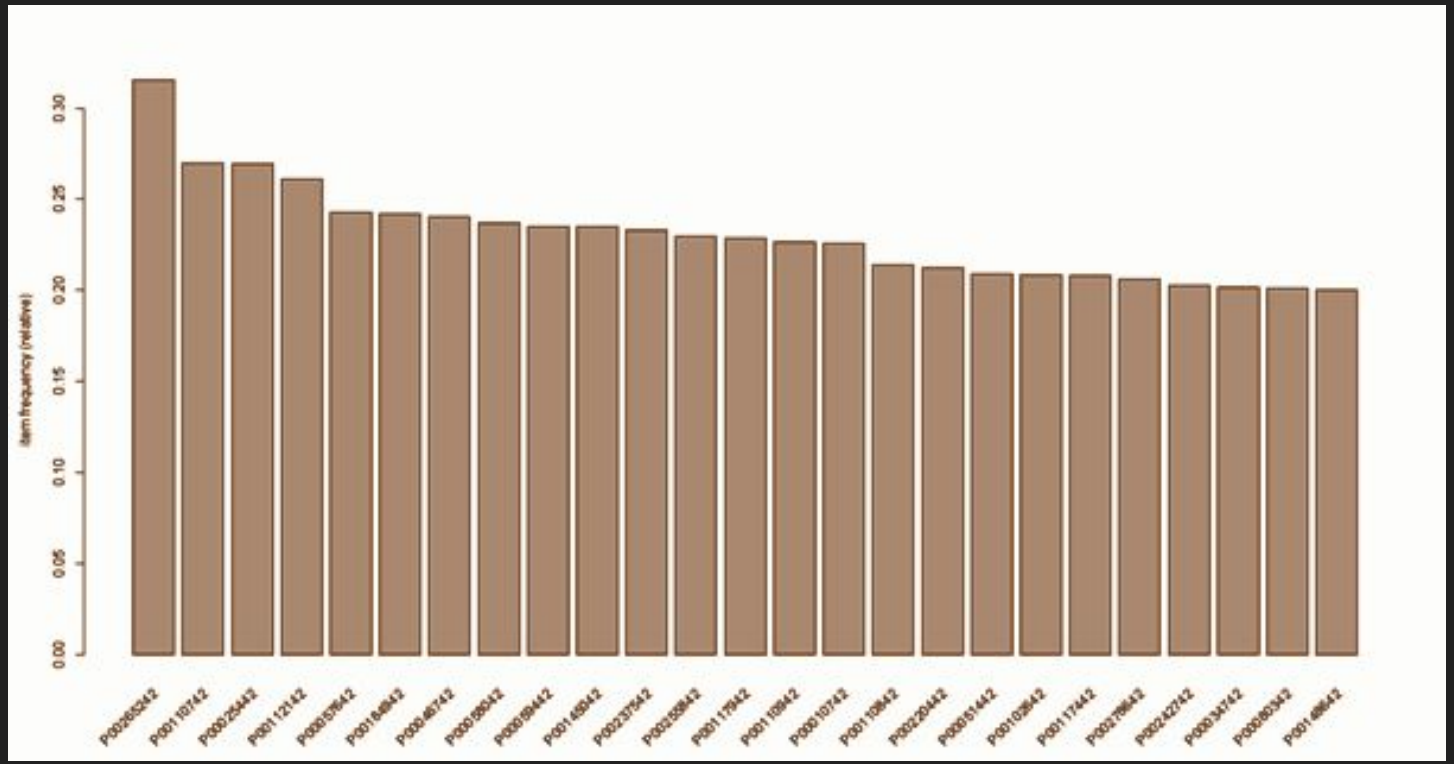
- Sparse Matrix Creation :
  - Apriori doesn't take strings or text as input, but rather 1 + 0. (Binary Format)
  - Allocate a column for each product(1-User bought product / 0 - User didn't)
  - arules library : read.transactions() -> Sparse Matrix
    rm.duplicates()-> Remove duplicates

```
> summary(customersProducts)
transactions as itemMatrix in sparse format with
 5892 rows (elements/itemsets/transactions) and
 10539 columns (items) and a density of 0.008768598

most frequent items:
P00265242 P00110742 P00025442 P00112142 P00057642   (Other)
     1858      1591      1586      1539      1430    536489

element (itemset/transaction) length distribution:
sizes
   6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27
   1    5    7   20   37   55   77   78  120  113  121  104  122  118   94   79   93   85   77   66   74   64
  28   29   30   31   32   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48   49
  72   72   74   77   58   50   58   39   63   56   53   40   55   57   51   44   49   37   42   41   43   38
  50   51   52   53   54   55   56   57   58   59   60   61   62   63   64   65   66   67   68   69   70   71
  36   32   36   41   40   49   51   30   43   35   26   30   24   27   27   48   32   30   29   16   31   30
  72   73   74   75   76   77   78   79   80   81   82   83   84   85   86   87   88   89   90   91   92   93
  40   25   27   24   30   31   30   19   25   20   21   23   23   30   22   14   20   20   20   14   18   28
  94   95   96   97   98   99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115
  20   22   27   17   15   13   16   20   20   12   13   13   11   12   17   21   18   12   15   19    8   23
 116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137
  18   17   10   15   19    9   10   14    5   18   11    7    9   14    7   21    6   13   11   10   13   15
 138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159
```

No. of items in a customer basket.

APRIORI ALGORITHM :

- ITEM FREQUENCY PLOT :



APRIORI ALGORITHM :

❑ TRAINING ASSOCIATION RULE MODEL :

- Application of apriori algorithm to train and generate rules.

- Main Parameters :
    - **Support** : min no. of transactions / Total no. of transactions.
      Why? Setting min support value for our rules to take effect

    - **Confidence** : How often rule is to be found True
      Why? To specify min strength of rule (default = 0.80)

APRIORI ALGORITHM :

- Apriori Result:

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.8    0.1    1 none FALSE            TRUE       0   0.008      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 47

set item appearances ...[0 item(s)] done [0.08s].
set transactions ...[10539 item(s), 5892 transaction(s)] done [4.47s].
sorting and recoding items ... [2099 item(s)] done [0.03s].
creating transaction tree ... done [0.45s].
checking subsets of size 1 2 3 4 5 6 done [21.11s].
writing ... [7 rule(s)] done [0.45s].
creating S4 object  ... done [0.45s].
```

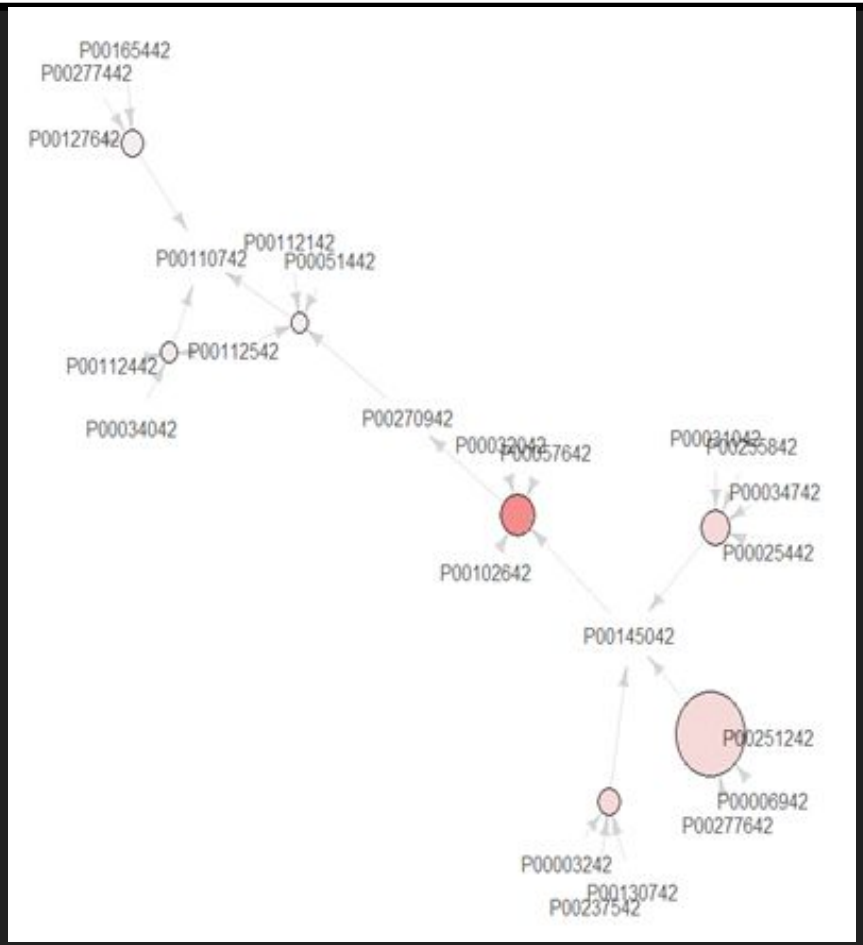
APRIORI ALGORITHM :

- Rules generated:

```
> inspect(sort(rules, by = 'lift'))
    lhs                                                rhs          support     confidence lift     count
[1] {P00032042,P00057642,P00102642,P00145042} => {P00270942} 0.008655804 0.8793103  4.540663 51
[2] {P00025442,P00031042,P00034742,P00255842} => {P00145042} 0.008486083 0.8064516  3.433246 50
[3] {P00003242,P00130742,P00237542}           => {P00145042} 0.008316361 0.8032787  3.419738 49
[4] {P00006942,P00251242,P00277642}           => {P00145042} 0.009674134 0.8028169  3.417773 57
[5] {P00034042,P00112442,P00112542}           => {P00110742} 0.008146640 0.8135593  3.012880 48
[6] {P00127642,P00165442,P00277442}           => {P00110742} 0.008316361 0.8032787  2.974807 49
[7] {P00051442,P00112142,P00112542,P00270942} => {P00110742} 0.008146640 0.8000000  2.962665 48
```

- **Confidence/ Benchmark Confidence = Lift > 1** , indicates a rule that is useful in finding consequent item sets.

APRIORI ALGORITHM :



Visual Representation of Association Rules:

-> Graph for 7 rules

Size : Support
Color: Lift

APRIORI ALGORITHM :

- INSIGHTS :
  - Product Placement optimization
  - Promotional Campaigns
  - Online Recommendation System


Q
A

THANK YOU !!