DATA 643 Proj 4

Dan Fanelli

June 27, 2017

Accuracy and Beyond

- This data was taken from the Kaggle "Instacart Market Basket Analysis" Competition
- https://www.kaggle.com/c/instacart-market-basket-analysis/data

Business Goal:

We would like to give newer products room to grow. Therefore, we will not view the order counts, we will simply observe whether a product has been purchsed previously or not. This will allow for the "underdog" products to get a greater opportunity than their numbers may allow for otherwise.

A Sample of the (Joined) Kaggle Data Set:

```
##          eval_set order_number order_dow order_hour_of_day
## 5450342     prior           29         3                 7
## 26191382    prior           76         5                12
## 12485409    prior           24         1                22
## 10629896    prior           12         1                20
## 19528828    prior            2         6                 8
## 19603213    prior            9         3                20
##          days_since_prior_order user_id order_id product_id BOUGHT
## 5450342                       5   34633   352010      33294      1
## 26191382                      6  166451   405607      16479      1
## 12485409                      0   78993   557942      14335      1
## 10629896                     11   67445   268495      35221      1
## 19528828                      7  123780   729466      12206      1
## 19603213                      4  124257   651828      19173      1
##          reordered                          product_name aisle_id
## 5450342          1 Tortillas, Wheat Free, Sprouted Corn      128
## 26191382         0     Fruit Spread, Deluxe, Strawberry       88
## 12485409         1               Phish FoodÂ® Ice Cream       37
## 10629896         1                 Lime Sparkling Water      115
## 19528828         0                         Basil Pesto        9
## 19603213         1 Orange Calcium & Vitamin D Pulp Free       98
##          department_id      department                          aisle
## 5450342              3          bakery          tortillas flat bread
## 26191382            13          pantry                       spreads
## 12485409             1          frozen                ice cream ice
## 10629896             7       beverages water seltzer sparkling water
## 19528828             9 dry goods pasta                   pasta sauce
## 19603213             7       beverages                juice nectars
```

A Sample of the 3 Fields that will be focused upon:

```
##          user_id product_id BOUGHT
## 5450342    34633      33294      1
## 26191382  166451      16479      1
## 12485409   78993      14335      1
## 10629896   67445      35221      1
## 19528828  123780      12206      1
## 19603213  124257      19173      1
```

UBCF Recommendations

```
## Available parameter (with default values):
## method    =  cosine
## nn    =  25
## sample    =  FALSE
## normalize     =  center
## verbose   =  FALSE
```

User	Rec_1	Rec_2	Rec_3	Rec_4	Rec_5
1	NA	NA	NA	NA	NA
2	NA	NA	NA	NA	NA
3	NA	NA	NA	NA	NA
4	NA	NA	NA	NA	NA
5	NA	NA	NA	NA	NA
6	NA	NA	NA	NA	NA

IBCF Recommendations

```
## Available parameter (with default values):
## k     =  30
## method    =  Cosine
## normalize     =  center
## normalize_sim_matrix  =  FALSE
## alpha     =  0.5
## na_as_zero    =  FALSE
## verbose   =  FALSE
```

User	Rec_1	Rec_2	Rec_3	Rec_4	Rec_5
1	NA	NA	NA	NA	NA
2	NA	NA	NA	NA	NA
3	NA	NA	NA	NA	NA
4	NA	NA	NA	NA	NA
5	NA	NA	NA	NA	NA
6	NA	NA	NA	NA	NA

RANDOM Recommendations

User	Rec_1	Rec_2	Rec_3	Rec_4	Rec_5
1	NA	NA	NA	NA	NA
2	NA	NA	NA	NA	NA
3	NA	NA	NA	NA	NA
4	NA	NA	NA	NA	NA
5	NA	NA	NA	NA	NA
6	NA	NA	NA	NA	NA

POPULAR Recommendations

```
## Available parameter (with default values):
## normalize     =  center
## aggregationRatings    =  function (x, na.rm = FALSE, dims = 1, ...)  standardGeneric("colMeans")
## aggregationPopularity     =  function (x, na.rm = FALSE, dims = 1, ...)  standardGeneric("colSums")
## verbose   =  FALSE
```

User	Rec_1	Rec_2	Rec_3	Rec_4	Rec_5
1	NA	NA	NA	NA	NA
2	NA	NA	NA	NA	NA
3	NA	NA	NA	NA	NA
4	NA	NA	NA	NA	NA
5	NA	NA	NA	NA	NA
6	NA	NA	NA	NA	NA

SVD Recommendations

User	Rec_1	Rec_2	Rec_3	Rec_4	Rec_5
1	NA	NA	NA	NA	NA
2	NA	NA	NA	NA	NA
3	NA	NA	NA	NA	NA
4	NA	NA	NA	NA	NA
5	NA	NA	NA	NA	NA
6	NA	NA	NA	NA	NA

ACCURACY: METRICS

```
##                RMSE          MSE         MAE
## UBCF     2233.9518    4990540.5  2228.36225
## IBCF    33117.6344 1096777710.9 32789.09779
## RANDOM   1176.7118    1384650.6   104.43972
## POPULAR   390.8166     152737.6    55.54977
## SVD       594.2836     353173.0    52.79278
```

If only online evaluation was possible:

Changes in recent events would not be possible with this offline scenario. Examples:

- A hurricane is on its way, people are stocking up on something early - what would that be?

- A terrorist attack has recently happened in a specific location, what products are newly in demand right now because of this.
- A recent twitter trending topic has arisen, and the seller would like to be able to react to this.

Figure 1: Changes

Conclusions

- Once again, the hours of loading into the base RDA showed more than in the past. In the past, it was merely the size of the data. In this case, I believe it showed just how expensive row relations can be, especially in case like this, where the text files have no notion of ***foreign keys***
- These packages do not seem stable to me. I run a job with good output, and then with no changes, click again and R crashes with no error output, just a line number. Java would never leave you hanging like that.
- It turned out to be a little bit of bad data and my not setting ***set.seed()*** in the sampling