Black Fridat Data Set

manu

January 26, 2019

### Introduction

A study of sales trough consumer behaviours.

The following data analysis explores the “black friday” dataset of 550 000 observations about the black Friday in a retail store, it contains different kinds of variables either numerical or categorical. It contains missing values.

In the present document, after some exploration and visualisations some models are trained so to predict the amount of money a customer will spend.

The data can be found in the following link <https://www.kaggle.com/mehdidag/black-friday> from Kaggle.

### Data import

After reading the data and importing the necessary packages we visualise the first few rows.

## User\_ID Product\_ID Gender Age Occupation City\_Category  
## 1 1000001 P00069042 F 0-17 10 A  
## 2 1000001 P00248942 F 0-17 10 A  
## 3 1000001 P00087842 F 0-17 10 A  
## 4 1000001 P00085442 F 0-17 10 A  
## 5 1000002 P00285442 M 55+ 16 C  
## Stay\_In\_Current\_City\_Years Marital\_Status Product\_Category\_1  
## 1 2 0 3  
## 2 2 0 1  
## 3 2 0 12  
## 4 2 0 12  
## 5 4+ 0 8  
## Product\_Category\_2 Product\_Category\_3 Purchase  
## 1 NA NA 8370  
## 2 6 14 15200  
## 3 NA NA 1422  
## 4 14 NA 1057  
## 5 NA NA 7969

### Feature exploration

In this segment we will explore the variables and prepare them so to have a better structure. By better structure here we mean a data set which follows what we call the tidyverse philosophy, that is, for the columns we have the variables and in the rows each variable. In their intersect we’ll have the value of the column for that observation.

We have noticed that each row ain’t a user but a product bought by a user. This ain’t incorrect, it depends on the purpose of each analyst. In our case (at least as a prime approximation), we’d rather prefer to have a user in each row. Hence, we modified our dataframe accordingly. As information will be lost, in order to lose fewer amount of information we count the number of products bought (“Product\_ID”), as well as the categories if any (“Product\_Category\_1”, “Product\_Category\_2”, “Product\_Category\_2”) and we sum the purchase quantity of the products, we believe that by doing this we will be losing less information. We rename some columns as well.

For doing this we embed pure SQL with the right package (check the Rmd for more details). Our result is as follows:

## IDUSER IDPROD GENDER AGE OCCUPATION CITY\_CAT YEARS\_IN\_CITY IS\_MARRIED  
## 1 1000001 34 F 0-17 10 A 2 0  
## 2 1000002 76 M 55+ 16 C 4+ 0  
## 3 1000003 29 M 26-35 15 A 3 0  
## PROD\_CAT1 PROD\_CAT2 PROD\_CAT3 PURCHASE  
## 1 34 21 14 333481  
## 2 76 54 26 810353  
## 3 29 23 13 341635

Let’s check the data type of the columns. That is important so to be able to analyse the data properly. For doing this we run a funtion created by us (check the Rmd for more details).

## Column type  
## IDUSER integer  
## IDPROD integer  
## GENDER character  
## AGE character  
## OCCUPATION integer  
## CITY\_CAT character  
## YEARS\_IN\_CITY character  
## IS\_MARRIED integer  
## PROD\_CAT1 integer  
## PROD\_CAT2 integer  
## PROD\_CAT3 integer  
## PURCHASE integer

Some of the columns (Occupation, Marital\_Status, Product\_Category\_1, Product\_Category\_2 and Product\_Category\_3) do not have the desired so we modify them accordingly. Hence we convert them and now are treated as factors. On the other hand, ‘User\_ID’ has been substracted one million losing ain’t information. Initial range of ‘User\_ID’ was 1000001, 1006040. The values of the “Purchase” field we consider them very high, we thought on dividing in per 1000 but finally decided to leaeve it as it is.

Once we have the features prepared, we can get a summary of them due to have an idea about them. Of course this makes more sense in the numerical fields. Here the summary of “Purchase”.

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 44108 234914 512612 851752 1099005 10536783

Our initial dataset did have missing values, but due to our modification we’ve got none now.

We can run our funtion created before that finds columns with null values and the percentage of them for the initial dataset and for the new dataset.

Columns with missing values in the initial dataset:

## Product\_Category\_2 Product\_Category\_3   
## 0.3106271 0.6944103

Columns with missing values in the new dataset:

## named numeric(0)

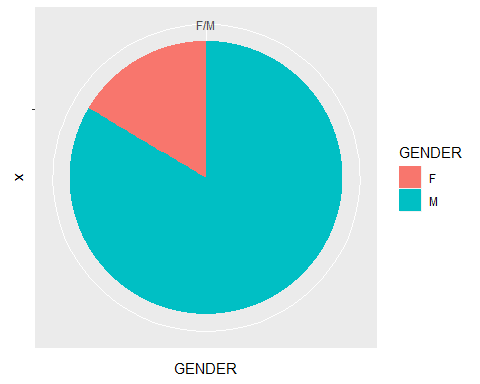
We got no null values now.

In case one desides to continue with the initial dataset that have products and no users as each observation, then in case of need to remove null values we recommend to erase the “Product\_Category\_3” column as it has got almost 70 % of missing values and then erase each observation with NAs for “Product\_Category\_2”. Of course information is lost but is one of the many paths that can be followed.

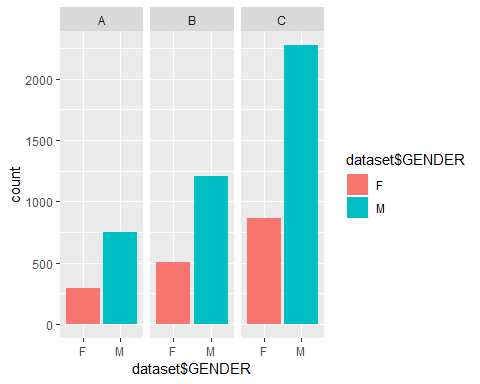
### On the visualisation of our data set

A continuation, we’ll proceed to perform some visualisations. The way we have feature engineered and modified our initial dataset now we’ve got in each row a user and in each column a value. Their intersect is a single value as explained before.

#### Gender

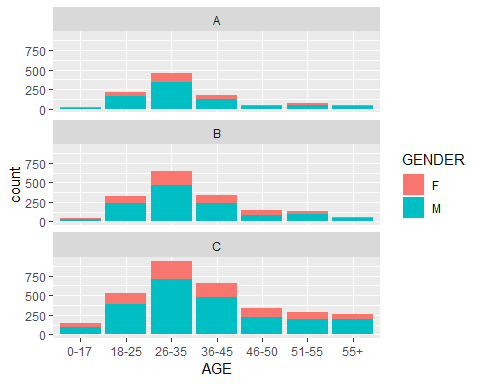


In this figure we can observe that more that 75% of our users are male.

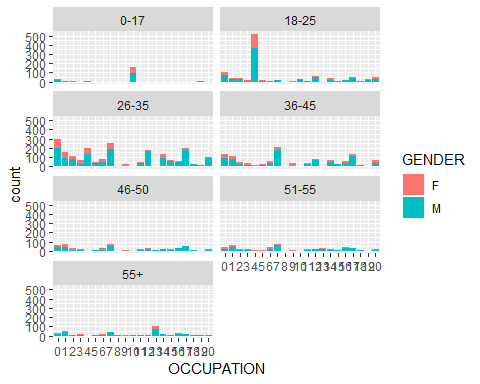


In the above plot we can see the amount of people per gender and per city. It kind of seems that many people of city category C bought in this retail shop in both genders. The same for city category B respect to A. Could be city category C is plenty of richer people

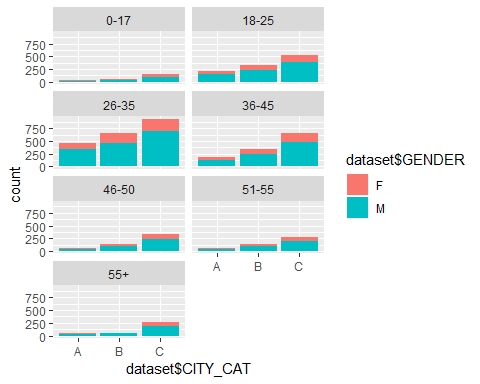
#### AGE



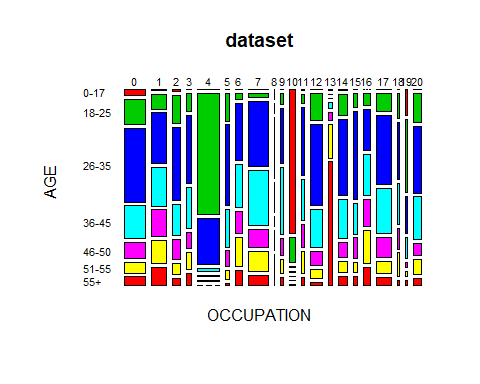
#### Occupation



Here we can observe the distribution of the occupation through the age ranges. It is interesting to observe how there are many young people working in the occupation number 5. Maybe this is the status of student or trainee. Apparently most are men but this is no surprise nor interesting for the fact we already observed that our data contained in its total a high number of male gender. If we would know the city name maybe we could check the proportion of the gender distribution in the goverment statistics and try to compare.

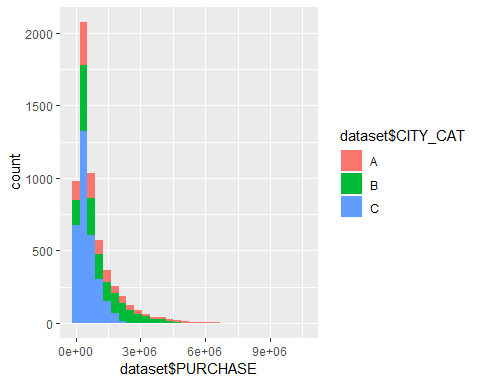


Mosaic plot for Occupation vs age.

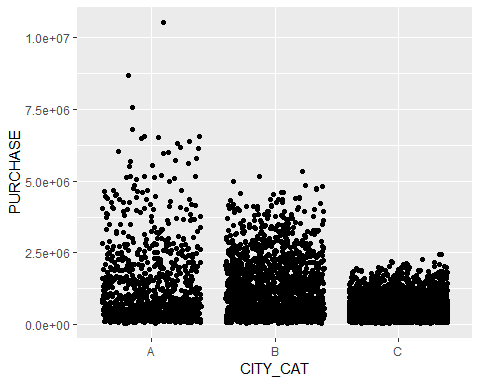


#### PURCHASE

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

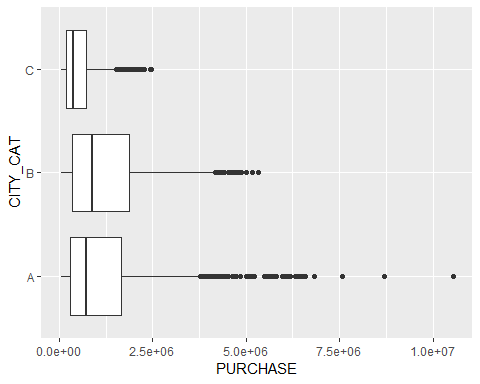


Here we can’t say much about the histogram of “purchase”, but we observe that the stayment in the current city is not quite unbalanced.



This makes no much sense but we wanted to include some scatterplot.

The above representation makes more sense if printed through a box plot as bellow.



We can observe some outliers mostly in city class A.

### On the model fitting of the black fridat dataset modified

We are going to predict the amout spent.

As our goal is not the to predict accurately we are going to fit a linear regression and xgboost models just to show some syntax here.

We separate the data in training set (75%) and test set(25%.

As an evaluation method we will use the RMSE (Root Mean Square Error).

We erase column “USERID” as is not useful for prediction. We either include product variables as are known post purchase.

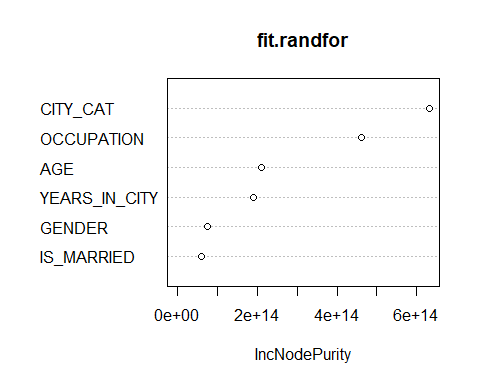
##   
## Call:  
## lm(formula = train.dataset$PURCHASE ~ ., data = train.dataset)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1407388 -501531 -160490 328195 7217702   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1077093.2 121227.2 8.885 < 2e-16 \*\*\*  
## GENDERM 247250.8 30151.6 8.200 3.12e-16 \*\*\*  
## AGE18-25 175.2 111739.7 0.002 0.998749   
## AGE26-35 142154.9 112069.3 1.268 0.204703   
## AGE36-45 108006.9 114199.5 0.946 0.344316   
## AGE46-50 79653.4 119276.1 0.668 0.504292   
## AGE51-55 -2285.3 121181.4 -0.019 0.984955   
## AGE55+ -138223.4 125378.0 -1.102 0.270325   
## OCCUPATION1 -35585.1 57821.9 -0.615 0.538305   
## OCCUPATION2 -54534.6 71559.4 -0.762 0.446048   
## OCCUPATION3 92623.6 83951.7 1.103 0.269959   
## OCCUPATION4 -53498.8 57514.3 -0.930 0.352328   
## OCCUPATION5 109565.3 105575.1 1.038 0.299423   
## OCCUPATION6 21653.9 75779.8 0.286 0.775085   
## OCCUPATION7 -123330.3 54200.8 -2.275 0.022928 \*   
## OCCUPATION8 132151.6 224868.5 0.588 0.556775   
## OCCUPATION9 -55711.1 111353.6 -0.500 0.616883   
## OCCUPATION10 -154712.5 119082.4 -1.299 0.193942   
## OCCUPATION11 -240943.2 94417.9 -2.552 0.010748 \*   
## OCCUPATION12 -240246.5 65080.9 -3.692 0.000226 \*\*\*  
## OCCUPATION13 -118787.1 103252.7 -1.150 0.250021   
## OCCUPATION14 -87776.7 70064.0 -1.253 0.210343   
## OCCUPATION15 -137399.6 89390.2 -1.537 0.124347   
## OCCUPATION16 42134.4 76447.3 0.551 0.581555   
## OCCUPATION17 -158327.7 59640.3 -2.655 0.007966 \*\*   
## OCCUPATION18 71328.7 125145.8 0.570 0.568731   
## OCCUPATION19 149625.8 122526.0 1.221 0.222085   
## OCCUPATION20 -13974.7 70873.5 -0.197 0.843698   
## CITY\_CATB -592.0 39072.4 -0.015 0.987912   
## CITY\_CATC -702261.6 35979.8 -19.518 < 2e-16 \*\*\*  
## YEARS\_IN\_CITY1 -58949.9 41967.3 -1.405 0.160192   
## YEARS\_IN\_CITY2 -56749.3 46503.4 -1.220 0.222407   
## YEARS\_IN\_CITY3 25090.3 48110.0 0.522 0.602032   
## YEARS\_IN\_CITY4+ -15051.7 48713.8 -0.309 0.757350   
## IS\_MARRIED1 3915.4 28060.3 0.140 0.889033   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 856400 on 4384 degrees of freedom  
## Multiple R-squared: 0.1743, Adjusted R-squared: 0.1679   
## F-statistic: 27.22 on 34 and 4384 DF, p-value: < 2.2e-16

Our goal ain’t inference but we’ll comment a bit the results.

In the results we can observe how apparently the gender plays an ‘important’ role, as well as city category c.

After testing our data we find out the RMSE for the linear model is 8.482848110^{5}. This is useful for comparing models.

Now we are going to implement, train and test a random forest model so to compete with the linear regretion fitted before. For more specification about this technique check <https://www.kdnuggets.com/2017/10/random-forests-explained.html>.



Above we print the variable importance according to random forest methodology. As we can observe the most important variable is city category, followed by occupation.

The RMSE result of training a random forest with a cross validation of 5 has been 8.625451110^{5}, which is higher than the one from the linear model. Hence, if we qould have to choose between them we would go for the linear model (strange result but whatever…).

#### A brief scent on recommenders system

With the initial dataset we’ll create a basic recommender system.

# dataset.recomm$Product\_Category\_1 <- NULL  
# dataset.recomm$Product\_Category\_2 <- NULL  
# dataset.recomm$Product\_Category\_3 <- NULL  
# dataset.recomm$Marital\_Status <- as.factor(dataset.recomm$Marital\_Status)  
#   
# dataset.recomm = dataset.recomm %>% mutate\_if(is.character, as.factor)  
# dataset.recomm