Black Friday Predictions by Categorical Dimensionality Reduction & Clustering

Jester Ugalde

```r
require(FactoMineR)
```

```
## Loading required package: FactoMineR
```

```r
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```r
library(readr)
library(MASS)
library(leaps)
```

```r
BlackFriday1 <- read_csv("BlackFriday1.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

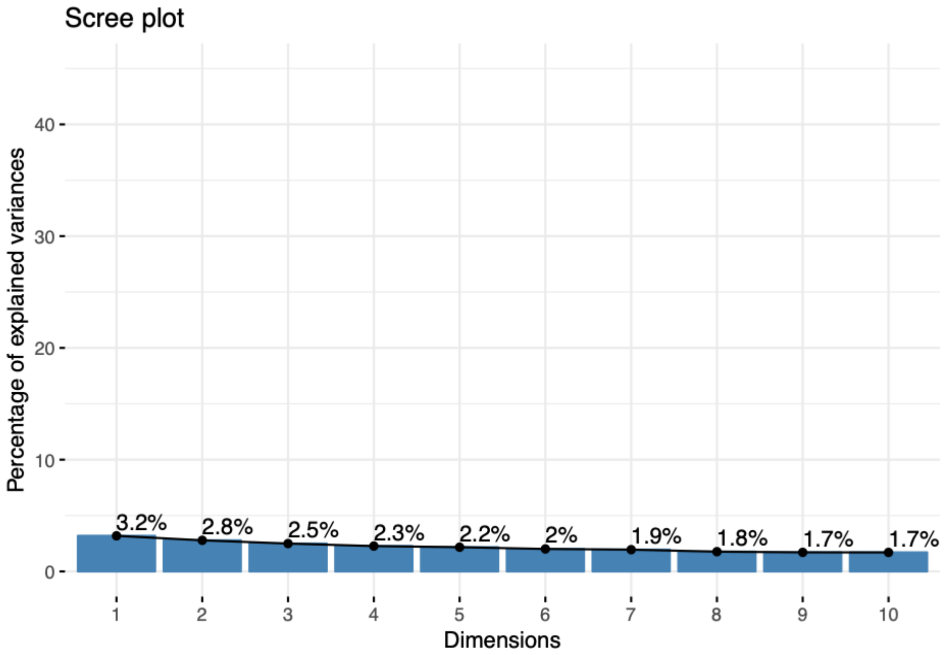
```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   Product_ID = col_character(),
##   Gender = col_double(),
##   Age = col_character(),
##   Occupation = col_double(),
##   City_Category = col_double(),
##   Stay_In_Current_City_Years = col_double(),
##   Marital_Status = col_double(),
##   Product_Category_1 = col_double(),
##   Product_Category_2 = col_double(),
##   Product_Category_3 = col_double(),
##   Purchase = col_double()
## )
```

```r
newbf = BlackFriday1[, c("Occupation", "City_Category", "Stay_In_Current_City_Years", "Marital_Status",

cats = apply(newbf, 2, function(x) nlevels(as.factor(x)))

newbf <- as.data.frame(sapply(newbf, as.factor))
```

```r
mca1 = MCA(newbf, ncp = 3, graph = FALSE)

library("factoextra")
```

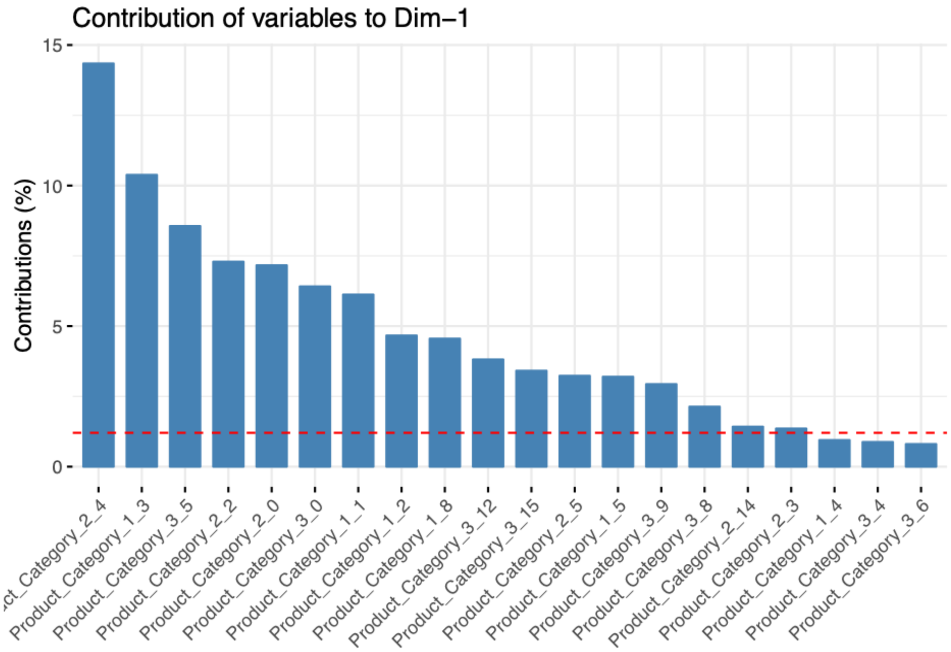
```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```r
#Scree plot to see percentage of variance explained
fviz_screeplot(mca1, addlabels = TRUE, ylim = c(0, 45))
```

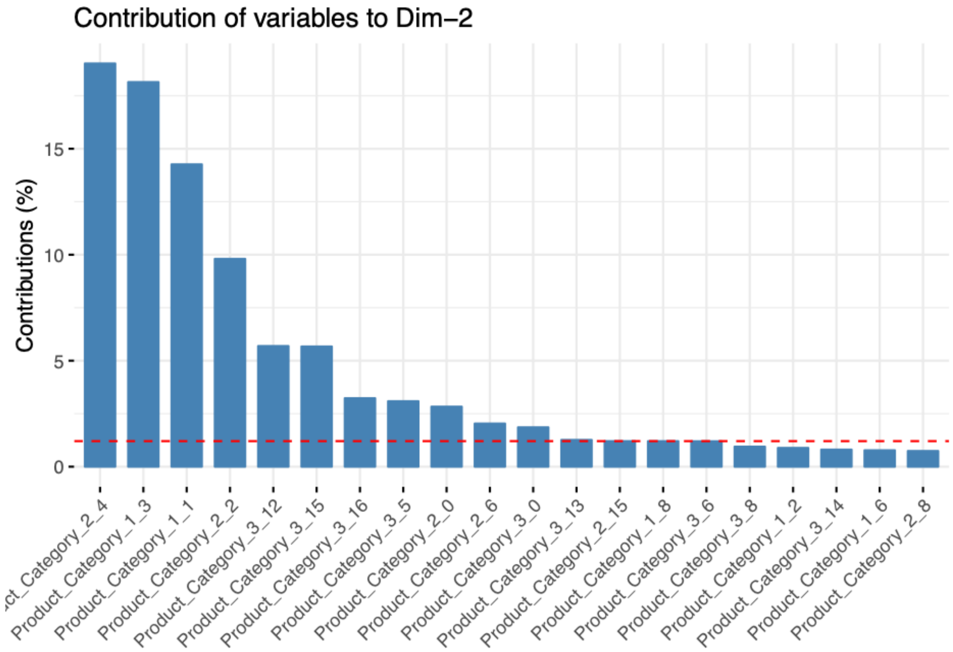


```
#Correlation of variables and first two dimensions
#fviz_mca_var(mca1, choice = "mca.cor", repel = TRUE, ggtheme = theme_minimal())
```

```
#Contribution from variables onto MCA'd data: First two dimensions
fviz_contrib(mca1, choice = "var", axes = 1, top = 20)
```

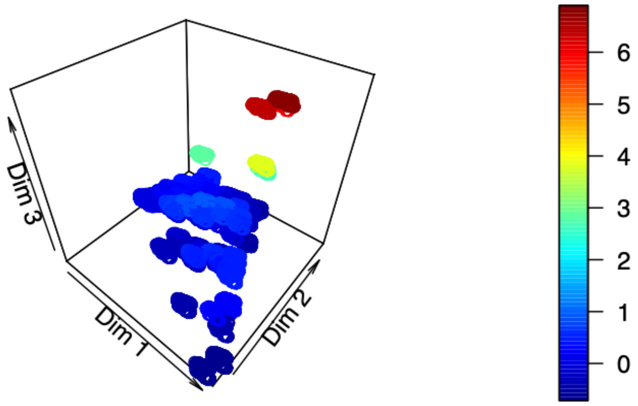


```r
fviz_contrib(mca1, choice = "var", axes = 2, top = 20)
```

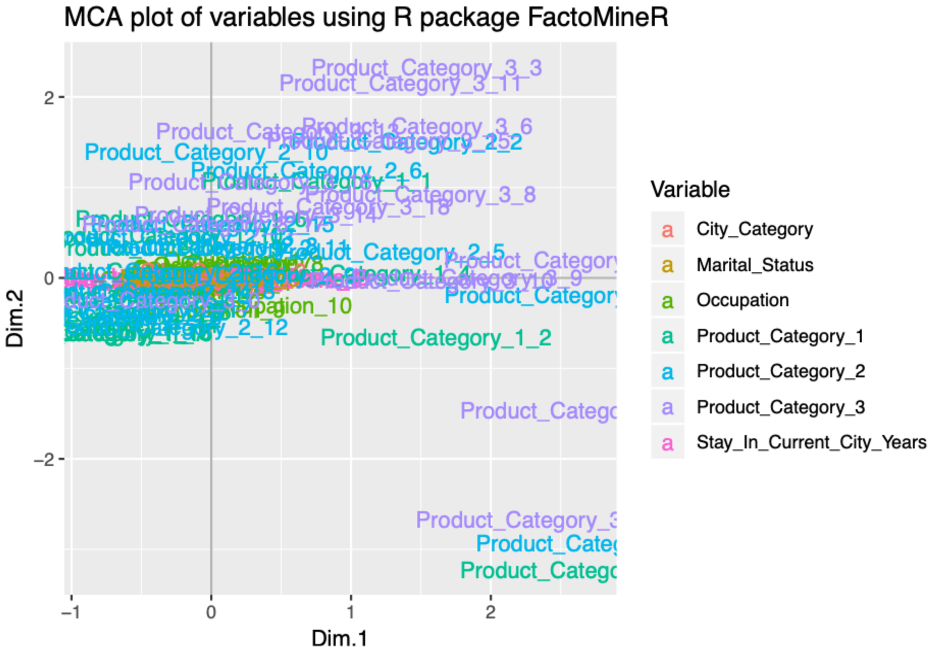


```
#Variable category placed on new dimensions with gradient color to show quality
#fviz_mca_var(mca1, col.var = "cos2",
             #gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             #repel = TRUE,
             #ggtheme = theme_minimal())

mca1_vars_df = data.frame(mca1$var$coord, Variable = rep(names(cats), cats))

mca1_obs_df = data.frame(mca1$ind$coord)
mca1_obs_df$Purchase <- BlackFriday1$Purchase
```

```
library(plot3D)
scatter3D(mca1_obs_df$Dim.1, xlab = "Dim 1", mca1_obs_df$Dim.2, ylab = "Dim 2",
          mca1_obs_df$Dim.3, zlab = "Dim 3", size = 10)
```



```r
ggplot(data=mca1_vars_df,
       aes(x = Dim.1, y = Dim.2, label = rownames(mca1_vars_df))) +
  geom_hline(yintercept = 0, colour = "gray70") +
  geom_vline(xintercept = 0, colour = "gray70") +
  geom_text(aes(colour=Variable)) +
  ggtitle("MCA plot of variables using R package FactoMineR")
```



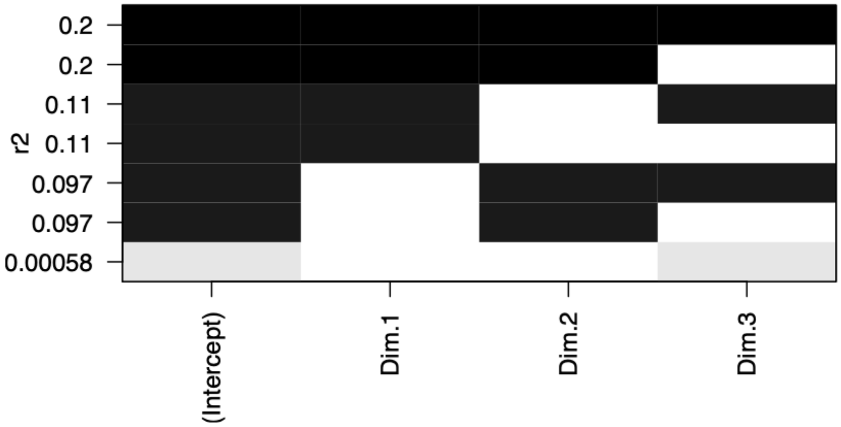
```r
#Splitting test and train data
smp_size <- floor(0.80 * nrow(mca1_obs_df))
set.seed(123)
train_ind <- sample(seq_len(nrow(mca1_obs_df)), size = smp_size)
```

```r
train <- mca1_obs_df[train_ind, ]
test <- mca1_obs_df[-train_ind, ]
```

```r
#Checking residuals squared to find variance to know which regression model to use
leaps<-regsubsets(Purchase ~ Dim.1 + Dim.2 + Dim.3, data = train, nbest=10)
summary(leaps)
```

```
## Subset selection object
## Call: regsubsets.formula(Purchase ~ Dim.1 + Dim.2 + Dim.3, data = train,
##     nbest = 10)
## 3 Variables  (and intercept)
##       Forced in Forced out
## Dim.1     FALSE      FALSE
## Dim.2     FALSE      FALSE
## Dim.3     FALSE      FALSE
## 10 subsets of each size up to 3
## Selection Algorithm: exhaustive
##          Dim.1 Dim.2 Dim.3
## 1  ( 1 ) "*"   " "   " "
## 1  ( 2 ) " "   "*"   " "
## 1  ( 3 ) " "   " "   "*"
## 2  ( 1 ) "*"   "*"   " "
## 2  ( 2 ) "*"   " "   "*"
## 2  ( 3 ) " "   "*"   "*"
## 3  ( 1 ) "*"   "*"   "*"
```

```r
plot(leaps, scale = "r2")
```



```r
#Linear Regression Model 2 Dimensions
LinReg2d <- lm(Purchase ~ Dim.1 + Dim.2, data = train)
predic2d <- predict.lm(LinReg2d, test)
#Accuracy
actuals_preds2d <- data.frame(cbind(actuals=test$Purchase, predicteds=predic2d))
correlation_accuracy2d <- cor(actuals_preds2d)
min_max_accuracy2d <- mean(apply(actuals_preds2d, 1, min) / apply(actuals_preds2d, 1, max))
mape2d <- mean(abs((actuals_preds2d$predicteds - actuals_preds2d$actuals))/actuals_preds2d$actuals)
```

```
sprintf("The min/max accuracy is %f ", min_max_accuracy2d*100)
```

```
## [1] "The min/max accuracy is 70.575020 "
```

```
sprintf("The mean absolute perc error is %f ", mape2d*100)
```

```
## [1] "The mean absolute perc error is 70.565048 "
```

```
#scatter2D(mca1_obs_df$Dim.1, xlab = "Dim 1", mca1_obs_df$Dim.2, ylab = "Dim 2")
```

```
#We will use 2dimensions since they 2D and 3D yield roughly the same accuracy so 3dimension linear
#regression will be blocked off

#Linear Regression model 3 dimensions
LinReg <- lm(Purchase ~ Dim.1 + Dim.2 + Dim.3, data = train)
predic <- predict.lm(LinReg, test)

#Accuracy for 3 Dimensions
actuals_preds <- data.frame(cbind(actuals=test$Purchase, predicteds=predic))
correlation_accuracy <- cor(actuals_preds)
min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actuals_preds, 1, max))
mape <- mean(abs((actuals_preds$predicteds - actuals_preds$actuals))/actuals_preds$actuals)

sprintf("The min/max accuracy is %f ", min_max_accuracy*100)
```

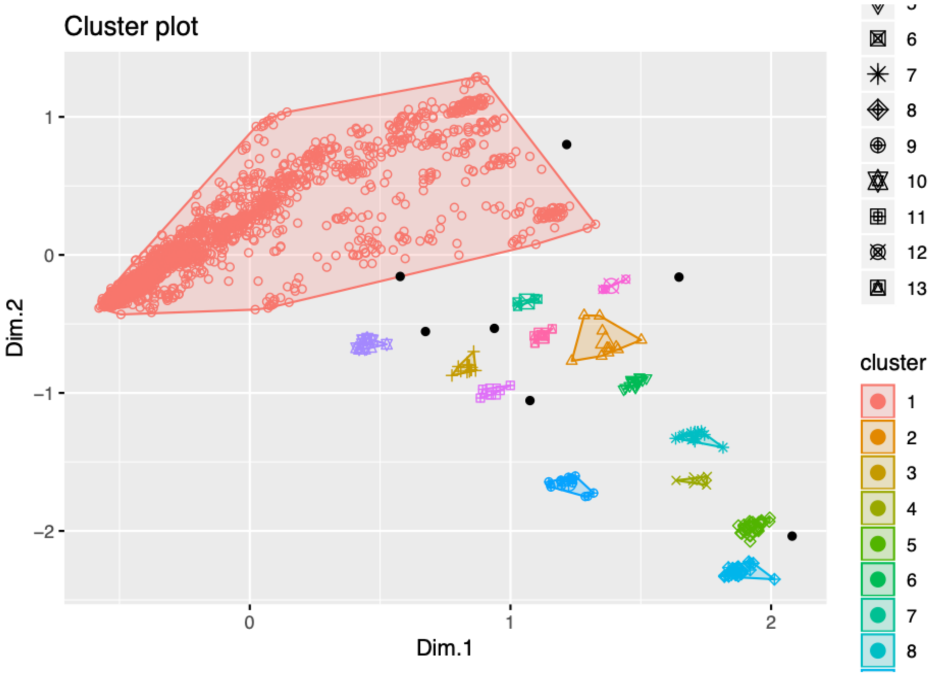
```
## [1] "The min/max accuracy is 70.599507 "
```

```
sprintf("The mean absolute perc error is %f ", mape*100)
```

```
## [1] "The mean absolute perc error is 70.563754 "
```

```
#Using a clustering algorithm to check if there are any underlying trends in the data
#That could have been missed
#Clustering <5% of the data since 500k+ samples would take a long time
d <- train[1:2500, 1:2]
library("fpc")
set.seed(123)
db <- fpc::dbscan(d, eps = 0.13, MinPts = 3)

library("factoextra")
fviz_cluster(db, d, stand = FALSE, ellipse = TRUE, frame = FALSE, geom = "point")
```

```
## Warning: argument frame is deprecated; please use ellipse instead.
```


Cluster plot
Dim.2
Dim.1
1
0
−1
−2
0
1
2
6
7
8
9
10
11
12
13
cluster
1
2
3
4
5
6
7
8