Fashion Products Analysis

Anisha

2022-11-03

```
library(janitor)
library(dplyr)
library(ggplot2)
library(tibble)
library(tidyr)
library(inspectdf)
library(nnet)
library(randomForest)
library(confintr)
library(waffle)
library(PASWR2)
```

```
# Look at the dataset
head(df_hist)
```

```
##   item_no   category      main_promotion  color stars success_indicator
## 1  739157      Tunic             Catalog  Green   3.1              flop
## 2  591846     Hoodie  Category_Highlight    Red   1.5              flop
## 3  337574 Sweatshirt             Catalog    Red   4.4               top
## 4  401933 Polo-Shirt  Category_Highlight   Blue   3.1              flop
## 5  812151     Hoodie  Category_Highlight  Green   4.1               top
## 6  200284     Hoodie Display_Ad_Campaign Yellow   3.9              flop
```

```
head(df_pred)
```

```
##   item_no   category     main_promotion  color stars
## 1  405901 Sweatshirt            Catalog   Blue   3.1
## 2  644275 Polo-Shirt   Frontpage_Header Yellow   2.6
## 3  533070      Tunic            Catalog  Green   2.7
## 4  829436 Polo-Shirt            Catalog Yellow   2.6
## 5  801722      Tunic            Catalog Yellow   4.9
## 6  866263    T-Shirt Category_Highlight  Black   2.6
```

```
# Check structure of variables
str(df_hist)
```

```
## 'data.frame':    8000 obs. of  6 variables:
##  $ item_no          : int  739157 591846 337574 401933 812151 200284 974264 389059 413025 615692 ...
##  $ category         : Factor w/ 6 levels "Blouse","Hoodie",..: 6 2 4 3 2 2 4 4 5 2 ...
```

```
##  $ main_promotion   : Factor w/ 4 levels "Catalog","Category_Highlight",..: 1 2 1 2 2 3 1 2 1 2 ...
##  $ color            : Factor w/ 10 levels "Black","Blue",..: 4 8 8 2 4 10 8 8 1 9 ...
##  $ stars            : num  3.1 1.5 4.4 3.1 4.1 3.9 1.4 1.8 3.2 5 ...
##  $ success_indicator: Factor w/ 2 levels "flop","top": 1 1 2 1 2 1 1 2 2 2 ...
```

```
str(df_pred)
```

```
## 'data.frame':    2000 obs. of  5 variables:
##  $ item_no       : int  405901 644275 533070 829436 801722 866263 502221 545865 440112 930925 ...
##  $ category      : Factor w/ 6 levels "Blouse","Hoodie",..: 4 3 6 3 6 5 4 6 4 6 ...
##  $ main_promotion: Factor w/ 4 levels "Catalog","Category_Highlight",..: 1 4 1 1 1 2 1 2 3 1 ...
##  $ color         : Factor w/ 10 levels "Black","Blue",..: 2 10 4 10 10 1 8 4 2 4 ...
##  $ stars         : num  3.1 2.6 2.7 2.6 4.9 2.6 1.6 3.5 3.7 2 ...
```

```
#Check columns with na's
inspect_na(df_hist)
```

```
## # A tibble: 6 x 3
##   col_name            cnt  pcnt
##   <chr>             <int> <dbl>
## 1 item_no               0     0
## 2 category              0     0
## 3 main_promotion        0     0
## 4 color                 0     0
## 5 stars                 0     0
## 6 success_indicator     0     0
```

```
inspect_na(df_pred)
```

```
## # A tibble: 5 x 3
##   col_name         cnt  pcnt
##   <chr>          <int> <dbl>
## 1 item_no            0     0
## 2 category           0     0
## 3 main_promotion     0     0
## 4 color              0     0
## 5 stars              0     0
```

```
# See the levels of nominal variables
unique(df_hist$category)
```

```
## [1] Tunic      Hoodie     Sweatshirt Polo-Shirt T-Shirt    Blouse
## Levels: Blouse Hoodie Polo-Shirt Sweatshirt T-Shirt Tunic
```

```
unique(df_hist$color)
```

```
##  [1] Green       Red         Blue        Yellow      Black       White
##  [7] Multi-Color Brown       Pink        Orange
## Levels: Black Blue Brown Green Multi-Color Orange Pink Red White Yellow
```

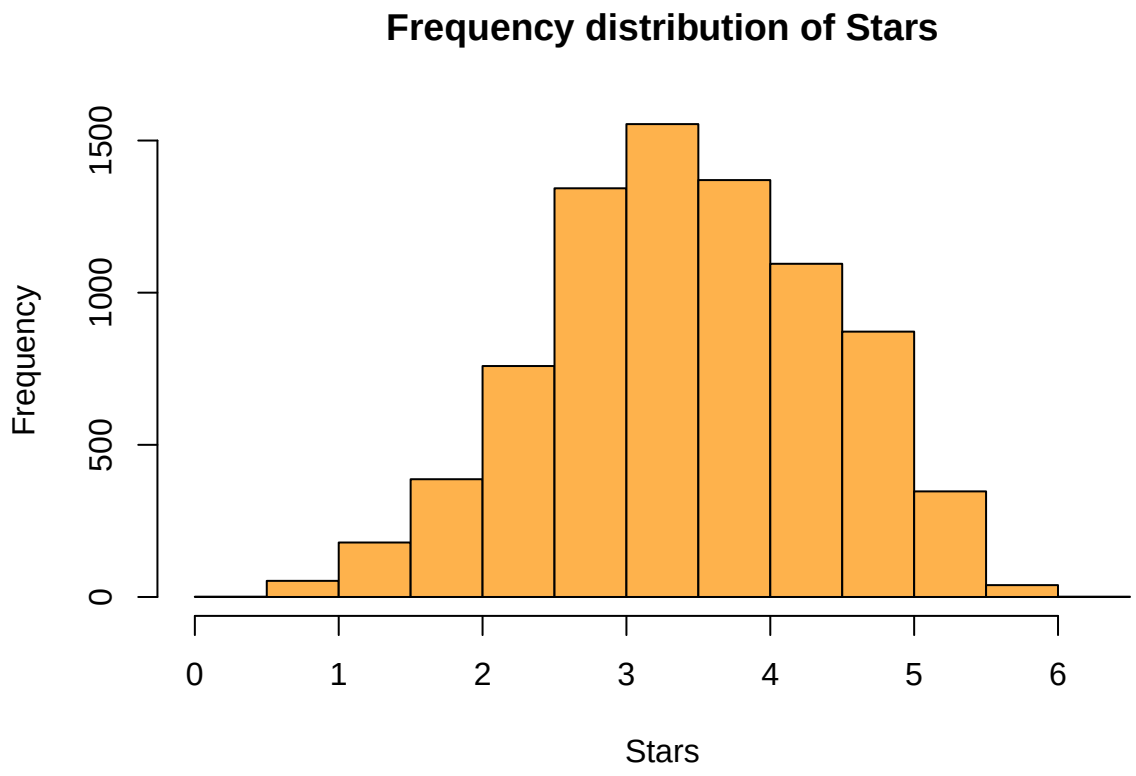
```r
unique(df_hist$main_promotion)
```

```
## [1] Catalog             Category_Highlight  Display_Ad_Campaign
## [4] Frontpage_Header
## Levels: Catalog Category_Highlight Display_Ad_Campaign Frontpage_Header
```

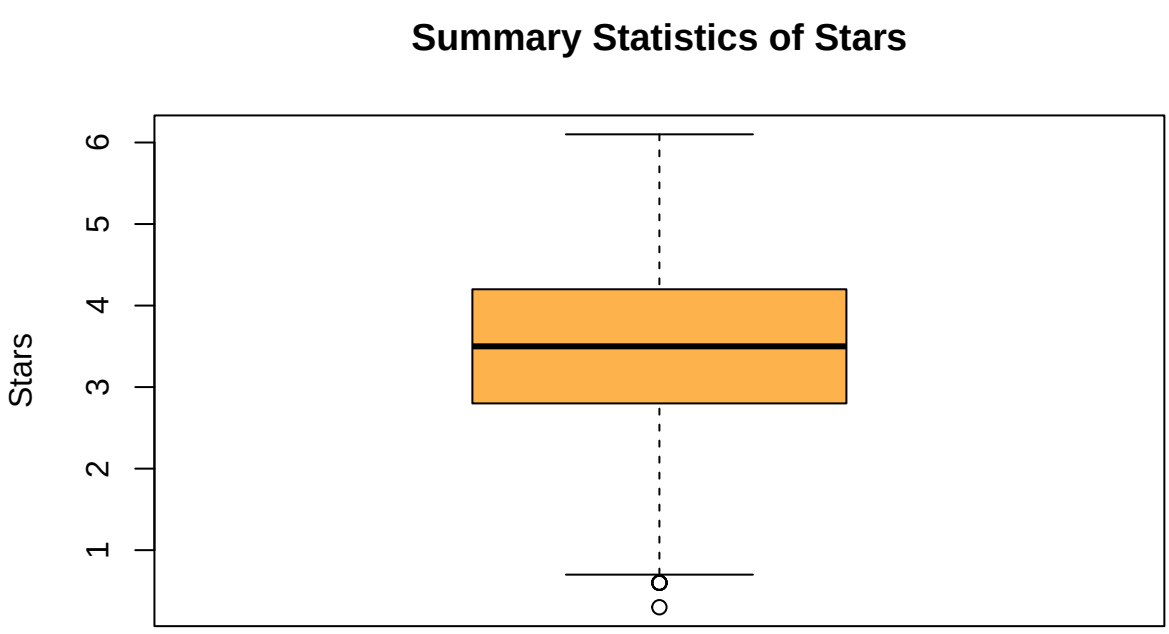
Data has no missing values. There is one numerical variable and four factor variables.

EDA

```r
# Frequency of products based on stars
hist(df_hist$stars, col = "#FEB24C", main = "Frequency distribution of Stars",
     xlab = "Stars")
```




```r
# Summary statistics of stars
boxplot(df_hist$stars, col = "#FEB24C", main = "Summary Statistics of Stars",
     ylab = "Stars")
```




```
summary(df_hist$stars)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.300   2.800   3.500   3.473   4.200   6.100
```

```
# check product with max and min stars
df_hist[which.max(df_hist$stars),]
```

```
##      item_no   category      main_promotion  color stars success_indicator
## 5189  560398 Polo-Shirt Display_Ad_Campaign Yellow   6.1               top
```

```
df_hist[which.min(df_hist$stars),]
```

```
##      item_no   category main_promotion  color stars success_indicator
## 7762  227347 Sweatshirt        Catalog Orange   0.3              flop
```

Two outliers based on stars. Frequency of stars is normally distributed.

```
# Distribution of products based on category
table(df_hist$category)
```

```
##
##     Blouse     Hoodie Polo-Shirt Sweatshirt    T-Shirt      Tunic
##       1246        739       1546       1360       1459       1650
```

```r
# Distribution of products based on promotion type
table(df_hist$main_promotion)
```

```
##
##             Catalog  Category_Highlight Display_Ad_Campaign    Frontpage_Header
##                2246                2432                1309                2013
```

```r
# Distribution of products based on color
table(df_hist$color)
```

```
##
##       Black        Blue       Brown       Green Multi-Color      Orange
##         812        1244         585         728        1443         592
##        Pink         Red       White      Yellow
##         412         776         352        1056
```

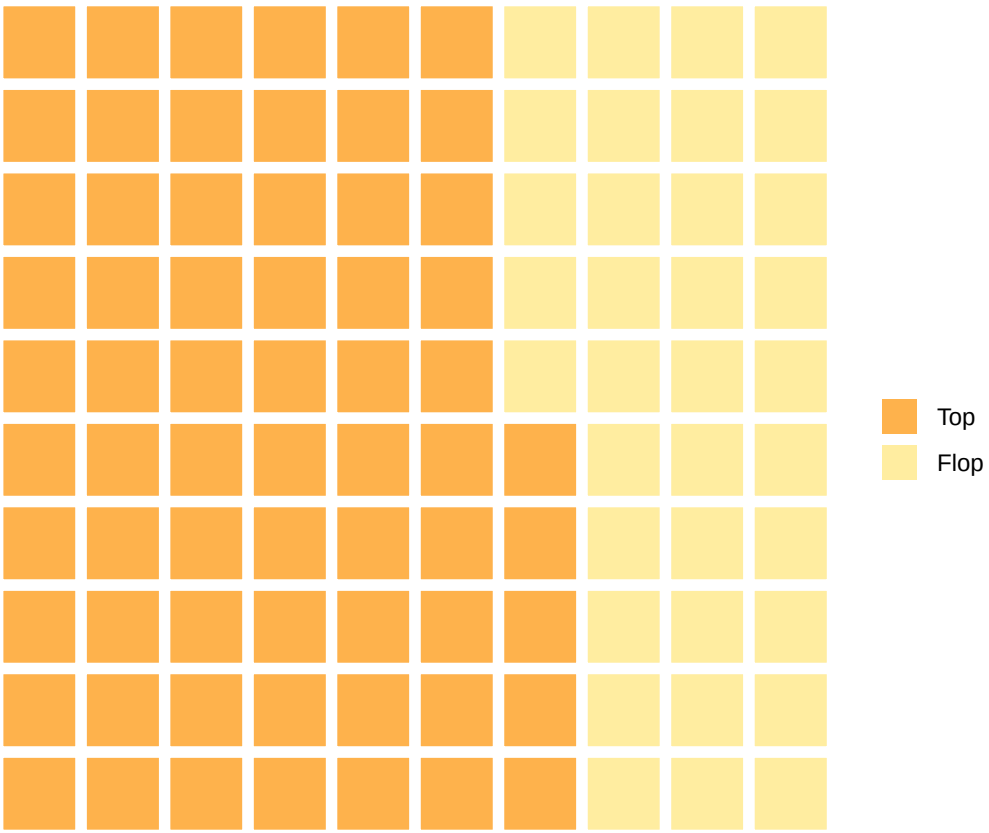
```r
# tabulate previous top/flop products
tab1 <- table(df_hist$success_indicator)
# Get percentage of previous top products
percntg_top <- (tab1[2]/sum(tab1))*100
round(percntg_top,2)
```

```
##   top
## 64.81
```

```r
# Get percentage of previous flop products
percntg_flop <- (tab1[1]/sum(tab1))*100
round(percntg_flop,2)
```

```
##  flop
## 35.19
```

```r
# Plot above result using a waffle plot
# Vector
z <- c(Top = 65, Flop = 35)

# Waffle plot
waffle(z, rows = 10,
       colors = c("#FEB24C","#FFEDA0"))
```



65% of products were successful. 35% weren't.

Cramer's v Association

```
# stars
tab_stars <- table(df_hist$success_indicator,df_hist$stars)
message(paste("Association between stars and success indicator is "),
        round(cramersv(tab_stars),2))
```

```
## Association between stars and success indicator is 0.58
```

```
# Category
tab_cat <- table(df_hist$success_indicator,df_hist$category)
message(paste("Association between category and success indicator is "),
        round(cramersv(tab_cat),2))
```

```
## Association between category and success indicator is 0.23
```

```
# promotion
tab_promo <- table(df_hist$success_indicator,df_hist$main_promotion)
message(paste("Association between promotion type and success indicator is "),
        round(cramersv(tab_promo),2))
```

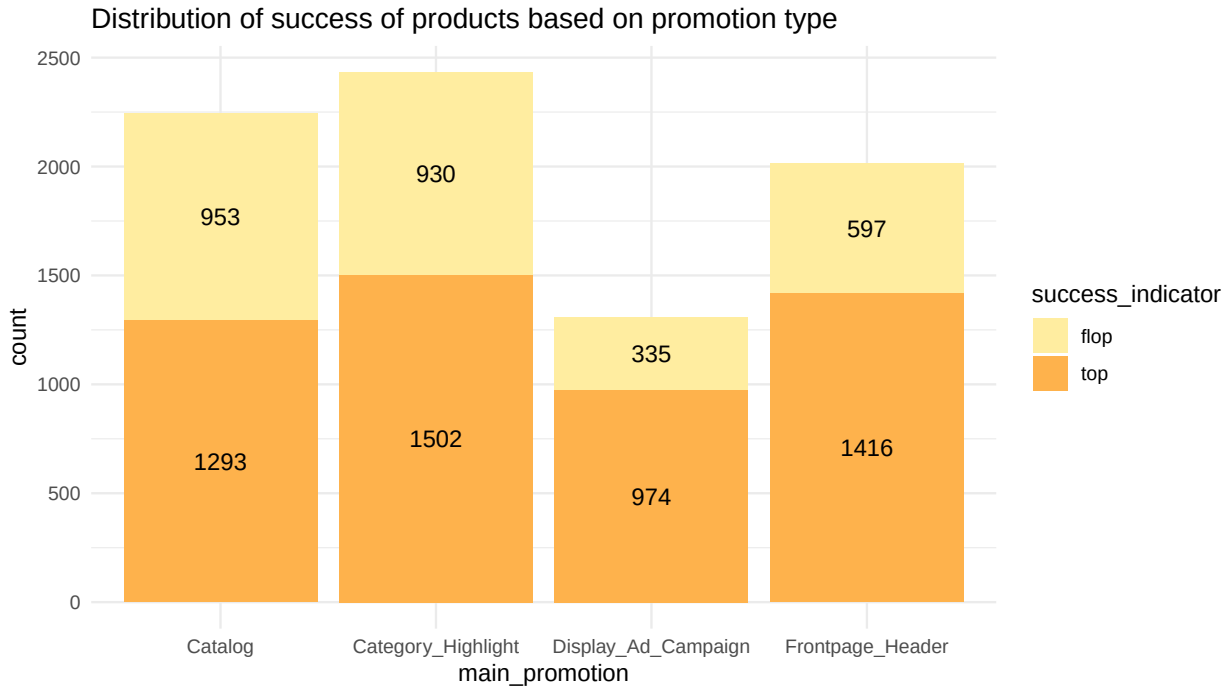
```
## Association between promotion type and success indicator is 0.13
```

```
# color
tab_col <- table(df_hist$success_indicator,df_hist$color)
message(paste("Association between color and success indicator is "),
        round(cramersv(tab_col),2))
```

```
## Association between color and success indicator is 0.22
```

Plots

```
# Change data for plots
df1 <- df_hist %>% group_by(main_promotion,success_indicator) %>% tally()
# change column names
colnames(df1)<- c("main_promotion","success_indicator","count")
# plot Distribution of success of products based on promotion type
ggplot(df1, aes(x=main_promotion, y=count, fill=success_indicator)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=count),color="black",
            position = position_stack(vjust = 0.5))+
  theme_minimal() +  scale_color_manual(values = c("#FFEDA0","#FEB24C")) +
  scale_fill_manual(values = c("#FFEDA0","#FEB24C")) +
  ggtitle("Distribution of success of products based on promotion type")
```




```
# Some percentages
promo <- unique(df1$main_promotion)
top_promo1 <- c() # -- promotion type where --% of its products were successful
top_promo2 <- c() # --% of top total is --
```

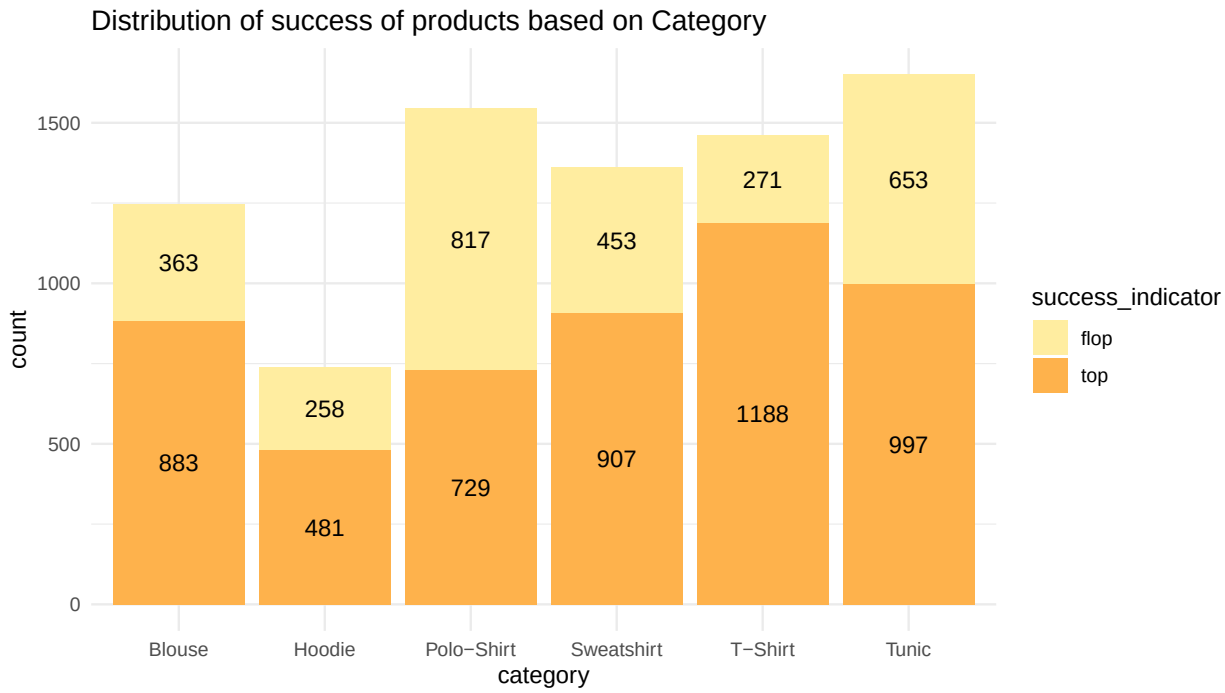
```r
for (i in promo) {
  top_promo1[i] <- (df1[which(df1$main_promotion == i &
                        df1$success_indicator == "top"),3]/
                      sum(df1[which(df1$main_promotion == i),3]))*100

  top_promo2[i] <- (df1[which(df1$main_promotion == i &
                        df1$success_indicator == "top"),3]/
                sum(df1[which(df1$success_indicator == "top"),3]))*100
}
top_promo1 # -- promotion type where --% of its products were successful
```

```
## $Catalog
## [1] 57.56901
##
## $Category_Highlight
## [1] 61.75987
##
## $Display_Ad_Campaign
## [1] 74.40794
##
## $Frontpage_Header
## [1] 70.34277
```

```r
top_promo2 # --% of top total is --
```

```
## $Catalog
## [1] 24.93732
##
## $Category_Highlight
## [1] 28.96818
##
## $Display_Ad_Campaign
## [1] 18.78496
##
## $Frontpage_Header
## [1] 27.30955
```

```r
#~~~~~~~~~~~~~~~~~~~~~~~~
# Change data for plots
df2 <- df_hist %>% group_by(category,success_indicator) %>% tally()
# change column names
colnames(df2)<- c("category","success_indicator","count")
# plot Distribution of success of products based on category
ggplot(df2, aes(x=category, y=count, fill=success_indicator)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=count),color="black",
            position = position_stack(vjust = 0.5))+
  theme_minimal() +  scale_color_manual(values = c("#FFEDA0","#FEB24C")) +
  scale_fill_manual(values = c("#FFEDA0","#FEB24C")) +
  ggtitle("Distribution of success of products based on Category")
```




```
# Some percentages
cat <- unique(df2$category)
top_cat1 <- c() # -- category where --% of its products were successful
top_cat2 <- c() # --% of top total is --
for (i in cat) {
  top_cat1[i] <- (df2[which(df2$category == i &
                        df2$success_indicator == "top"),3]/
                  sum(df2[which(df2$category == i),3]))*100

  top_cat2[i]  <- (df2[which(df2$category == i &
                        df2$success_indicator == "top"),3]/
                sum(df2[which(df2$success_indicator == "top"),3]))*100
}
top_cat1 # -- category where --% of its products were successful
```

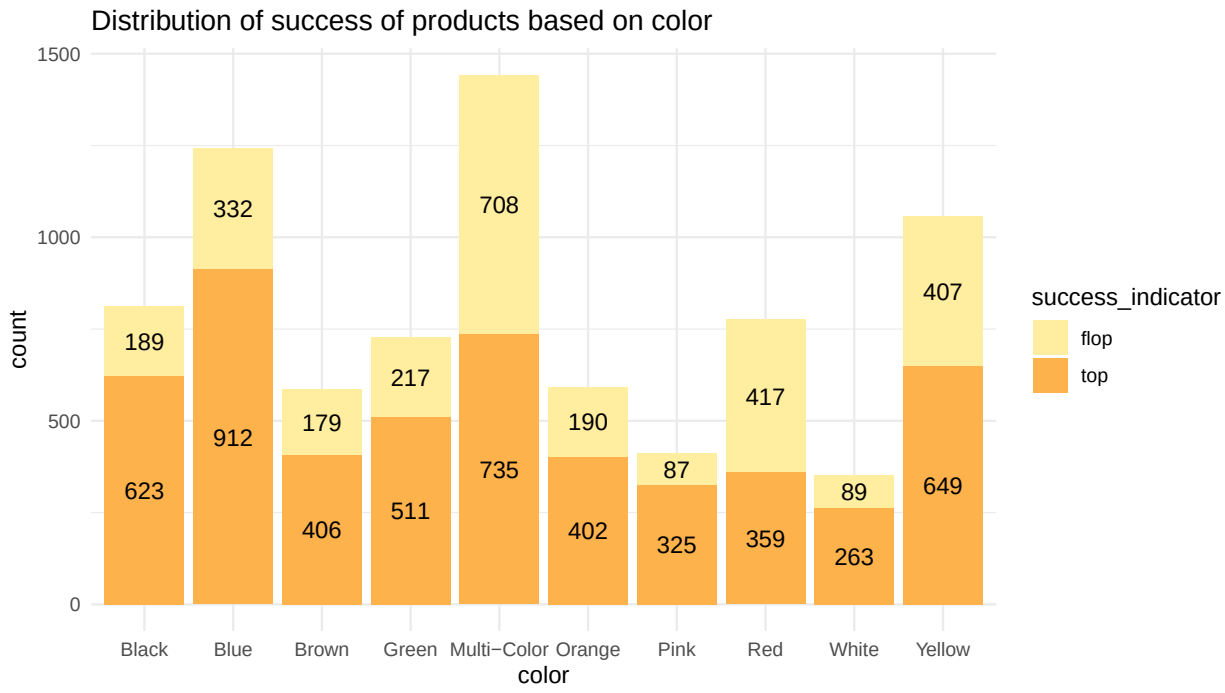
```
## $Blouse
## [1] 70.86677
##
## $Hoodie
## [1] 65.08796
##
## $`Polo-Shirt`
## [1] 47.15395
##
## $Sweatshirt
## [1] 66.69118
##
## $`T-Shirt`
## [1] 81.42563
##
## $Tunic
```

```
## [1] 60.42424
```

```
top_cat2 # --% of top total is --
```

```
## $Blouse
## [1] 17.02989
##
## $Hoodie
## [1] 9.27676
##
## $`Polo-Shirt`
## [1] 14.05979
##
## $Sweatshirt
## [1] 17.49277
##
## $`T-Shirt`
## [1] 22.91225
##
## $Tunic
## [1] 19.22854
```

```
#~~~~~~~~~~~~~~~~~~~~~~~~
# Change data for plots
df3 <- df_hist %>% group_by(color,success_indicator) %>% tally()
# change column names
colnames(df3)<- c("color","success_indicator","count")
# plot Distribution of success of products based on color
ggplot(df3, aes(x=color, y=count, fill=success_indicator)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=count),color="black",
            position = position_stack(vjust = 0.5))+
  theme_minimal() +  scale_color_manual(values = c("#FFEDA0","#FEB24C")) +
  scale_fill_manual(values = c("#FFEDA0","#FEB24C")) +
  ggtitle("Distribution of success of products based on color")
```

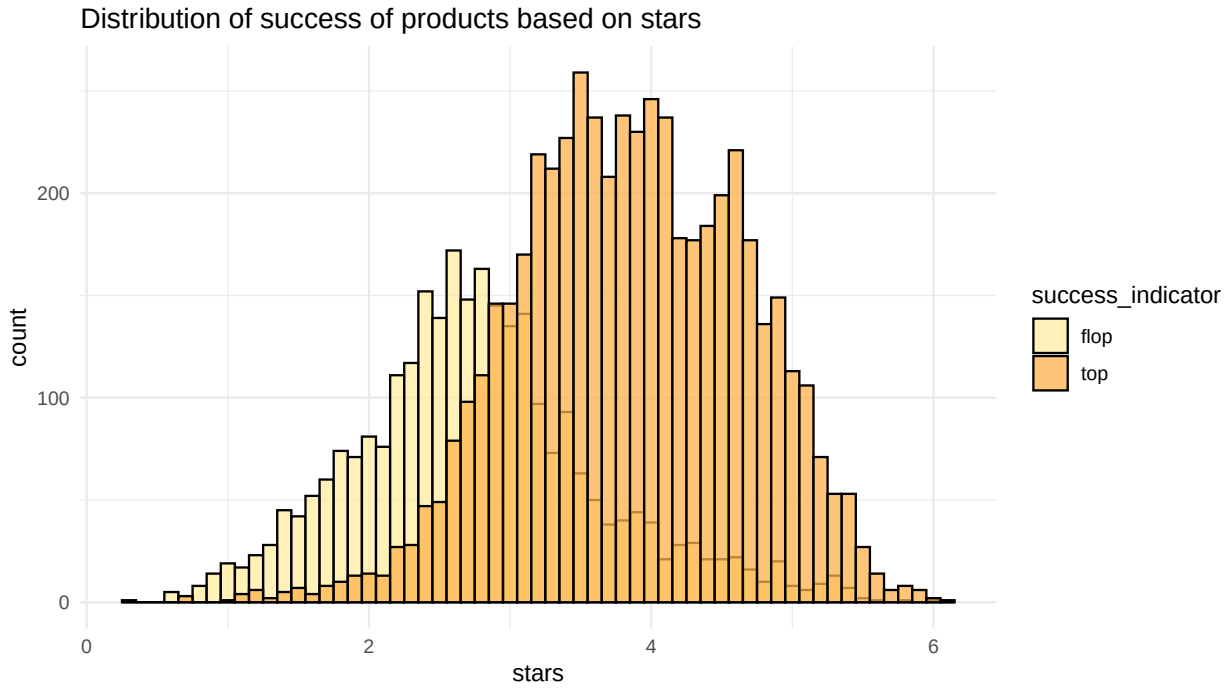



```
# Some percentages
col <- unique(df3$color)
top_col <- c() # -- color where --% of its products were successful
for (i in col) {

  top_col[i] <- (df3[which(df3$color == i &
                        df3$success_indicator == "top"),3]/
                  sum(df3[which(df3$color == i),3]))*100
}
top_col # -- color where --% of its products were successful
```

```
## $Black
## [1] 76.72414
##
## $Blue
## [1] 73.3119
##
## $Brown
## [1] 69.40171
##
## $Green
## [1] 70.19231
##
## $`Multi-Color`
## [1] 50.93555
##
## $Orange
## [1] 67.90541
##
## $Pink
## [1] 78.8835
```

```
##
## $Red
## [1] 46.26289
##
## $White
## [1] 74.71591
##
## $Yellow
## [1] 61.45833
```

```
#~~~~~~~~~~~~~~~~~~~~~~~~
# plot Distribution of success of products based on stars
ggplot(df_hist, aes(x=stars, fill=success_indicator, color=success_indicator)) +
  geom_histogram(position="identity", alpha=0.75, binwidth = 0.1, color=1)+
  theme_minimal() +  scale_color_manual(values = c("#FFEDA0","#FEB24C")) +
  scale_fill_manual(values = c("#FFEDA0","#FEB24C")) +
  ggtitle("Distribution of success of products based on stars")
```



ML model

```
# Same results
set.seed(2022)
N <- nrow(df_hist)
keep <- sample(1:N, 6000)
test <- setdiff(1:N, keep)
# Training data
dat <- df_hist[keep,]
N_train <- nrow(dat)
```

```r
# Testing data
dat_test <- df_hist[test,]
```

logistic Regression

```r
# Fit logistic regression
fit_lr <- multinom(success_indicator ~ ., data = dat[,-1],maxit = 30)
```

```
## # weights:  20 (19 variable)
## initial  value 4158.883083
## iter  10 value 2851.947831
## iter  20 value 2725.405636
## final  value 2719.266242
## converged
```

```r
# examine the results
summary(fit_lr)
```

```
## Call:
## multinom(formula = success_indicator ~ ., data = dat[, -1], maxit = 30)
##
## Coefficients:
##                                       Values  Std. Err.
## (Intercept)                       -4.77557072 0.22947377
## categoryHoodie                     0.40249809 0.14609733
## categoryPolo-Shirt                -0.30077193 0.11980555
## categorySweatshirt                 0.62186282 0.13144056
## categoryT-Shirt                    1.09953096 0.13776888
## categoryTunic                      0.09854679 0.11680030
## main_promotionCategory_Highlight   0.22934814 0.09063171
## main_promotionDisplay_Ad_Campaign -0.01515704 0.11611818
## main_promotionFrontpage_Header     0.60620824 0.09705904
## colorBlue                          0.32107612 0.15403301
## colorBrown                        -0.41217642 0.18150205
## colorGreen                        -0.02812180 0.16951491
## colorMulti-Color                  -0.46807097 0.14396303
## colorOrange                        0.28751277 0.18564195
## colorPink                          0.40326722 0.20278003
## colorRed                          -0.94622888 0.16864581
## colorWhite                        -0.12192592 0.21777949
## colorYellow                       -0.47467737 0.15529805
## stars                              1.54560473 0.04832184
##
## Residual Deviance: 5438.532
## AIC: 5476.532
```

```r
# Predict results for test set
dat_test$pred_lr <- predict(fit_lr, newdata = dat_test[,-1])
# Confusion matrix
tab2 <- table(dat_test$pred_lr,dat_test$success_indicator)
tab2
```

```
##
##        flop  top
##   flop  479  134
##   top   233 1154
```

```
# Check accuracy
acc <- (sum(diag(tab2))/sum(tab2))*100
round(acc,2)
```

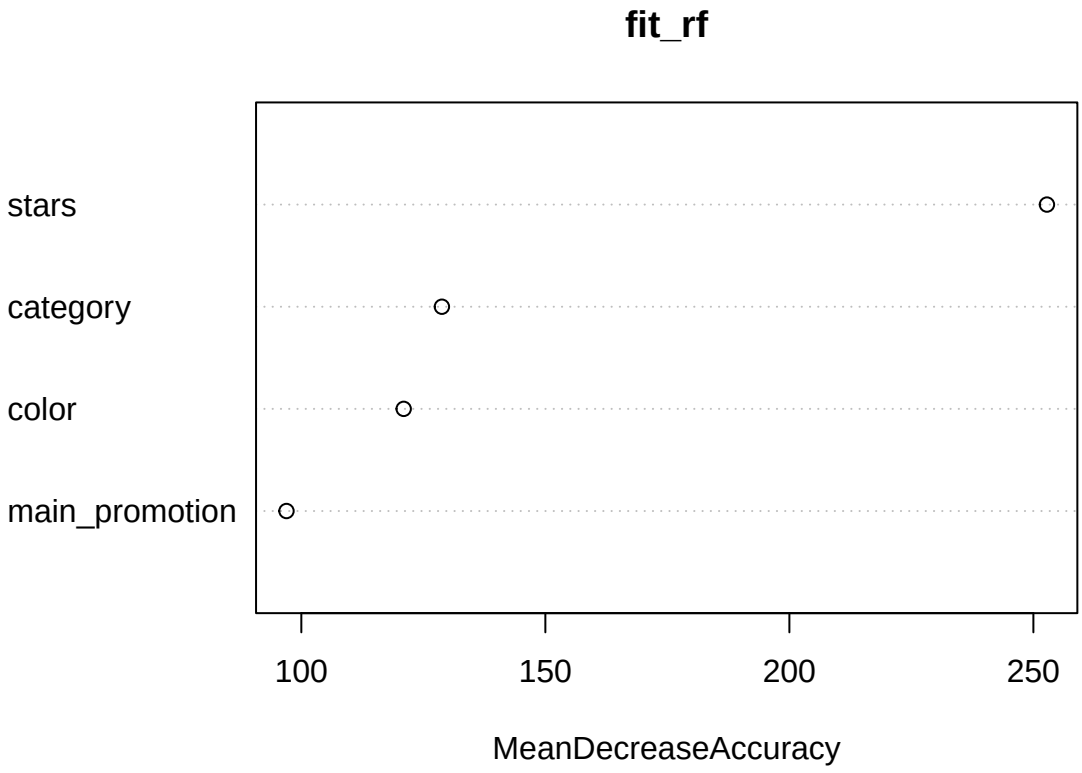
```
## [1] 81.65
```

Random forest

```
# implement the random forest algorithm
fit_rf <- randomForest(success_indicator ~ ., data = dat[,-1],
                       importance = TRUE)

# examine the results
fit_rf
```

```
##
## Call:
##  randomForest(formula = success_indicator ~ ., data = dat[, -1],      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 15.9%
## Confusion matrix:
##      flop  top class.error
## flop 1533  570  0.27104137
## top   384 3513  0.09853734
```

```
# look at variable importance
varImpPlot(fit_rf, type = 1)
```




```
# Predict results for test set
dat_test$pred_rf <- predict(fit_rf, type = "class",
                            newdata = dat_test[,-c(1,7)])
# Confusion matrix
tab3 <- table(dat_test$pred_rf,dat_test$success_indicator)
tab3
```

```
##
##        flop  top
##   flop  507  118
##   top   205 1170
```

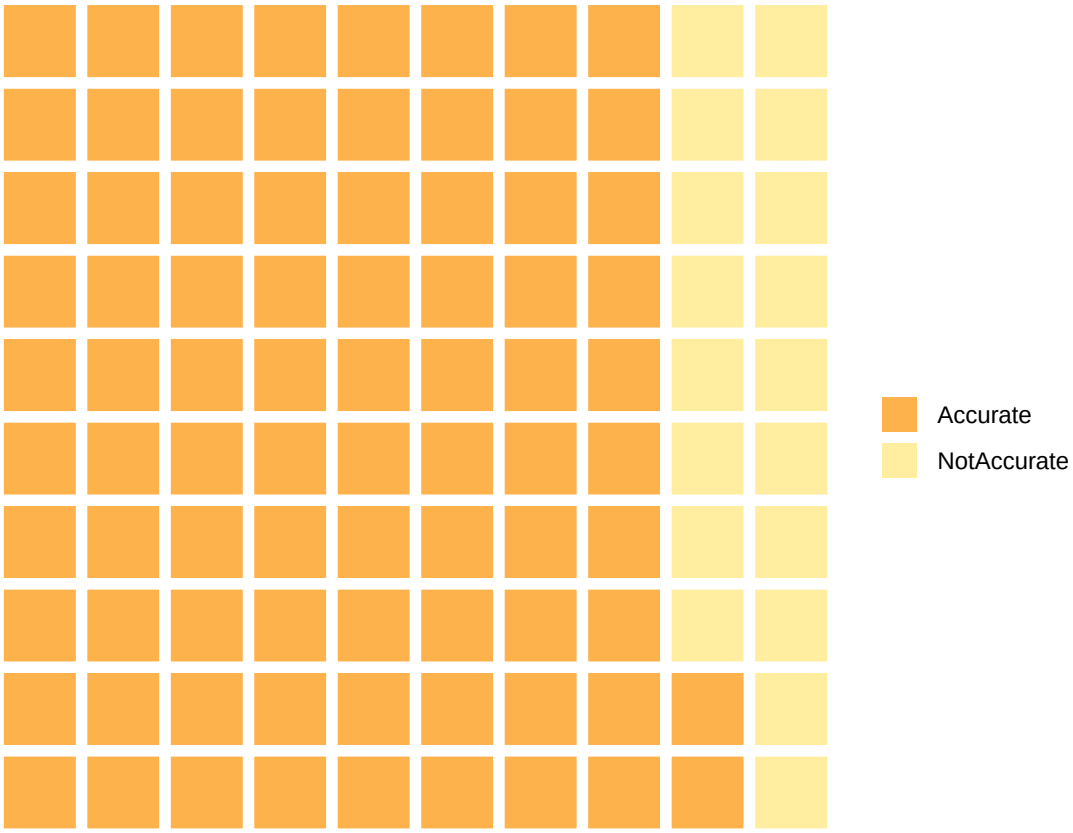
```
# Check accuracy
acc <- sum(diag(tab3))/sum(tab3)*100
round(acc,2)
```

```
## [1] 83.85
```

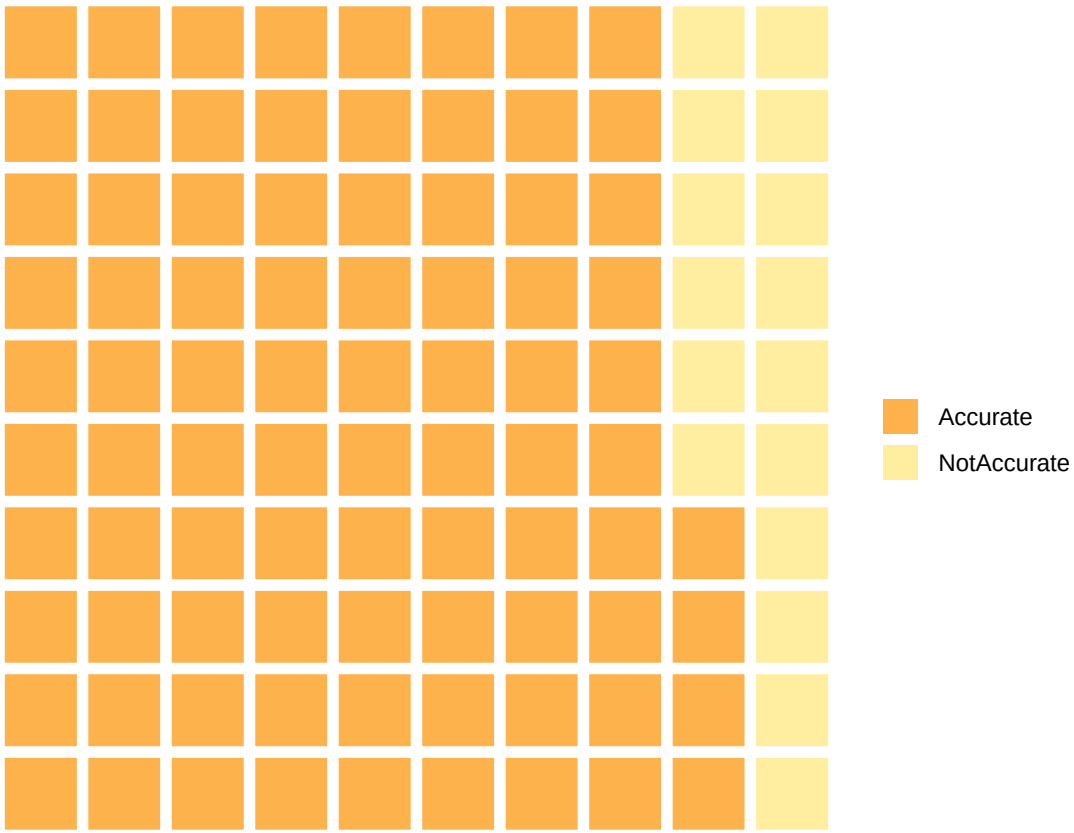
Random Forest is better since higher accuracy.

```
# Plot above result using a waffle plot
# Vector
x <- c(Accurate = 82, NotAccurate = 18) #Logistic
y <- c(Accurate = 84, NotAccurate = 16) #Random forest
```

```r
# Waffle chart
waffle(x, rows = 10,
       colors = c("#FEB24C","#FFEDA0"))
```




```r
waffle(y, rows = 10,
       colors = c("#FEB24C","#FFEDA0"))
```



Answer

```
# Use random forest to predict results for potential products
df_pred$pred_rf_new <- predict(fit_rf, type = "class", newdata = df_pred)
# See 10 results
head(df_pred,10)
```

```
##    item_no   category      main_promotion  color stars pred_rf_new
## 1   405901 Sweatshirt             Catalog   Blue   3.1         top
## 2   644275 Polo-Shirt    Frontpage_Header Yellow   2.6        flop
## 3   533070      Tunic             Catalog  Green   2.7        flop
## 4   829436 Polo-Shirt             Catalog Yellow   2.6        flop
## 5   801722      Tunic             Catalog Yellow   4.9         top
## 6   866263    T-Shirt  Category_Highlight  Black   2.6         top
## 7   502221 Sweatshirt             Catalog    Red   1.6        flop
## 8   545865      Tunic  Category_Highlight  Green   3.5         top
## 9   440112 Sweatshirt Display_Ad_Campaign   Blue   3.7         top
## 10  930925      Tunic             Catalog  Green   2.0        flop
```

```
# Tabulate the result
tab4 <- table(df_pred$pred_rf_new)
tab4
```

```
##
## flop  top
##  620 1380
```

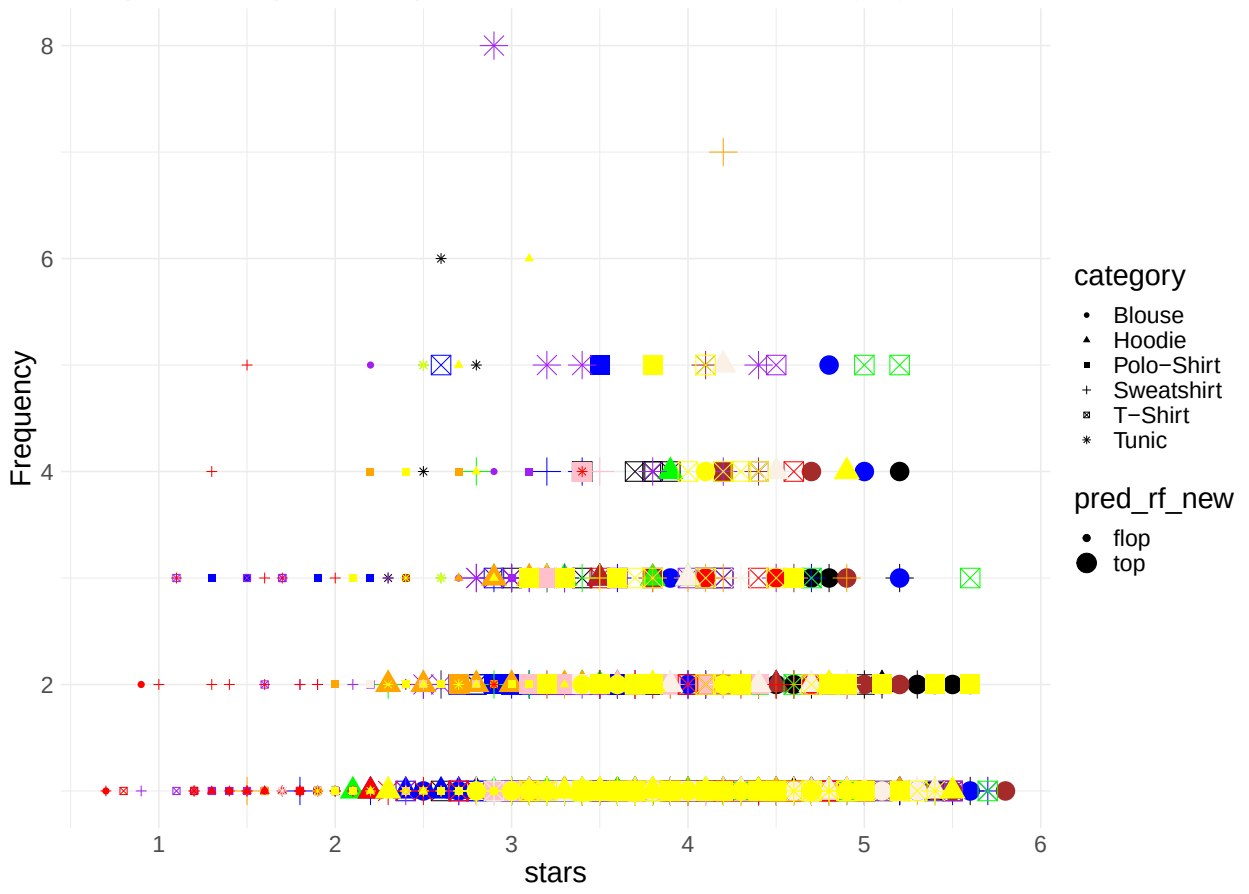
```
# Percentage top
percntg_flop <- ((sum(tab4[1]))/sum(tab4))*100
percntg_flop
```

```
## [1] 31
```

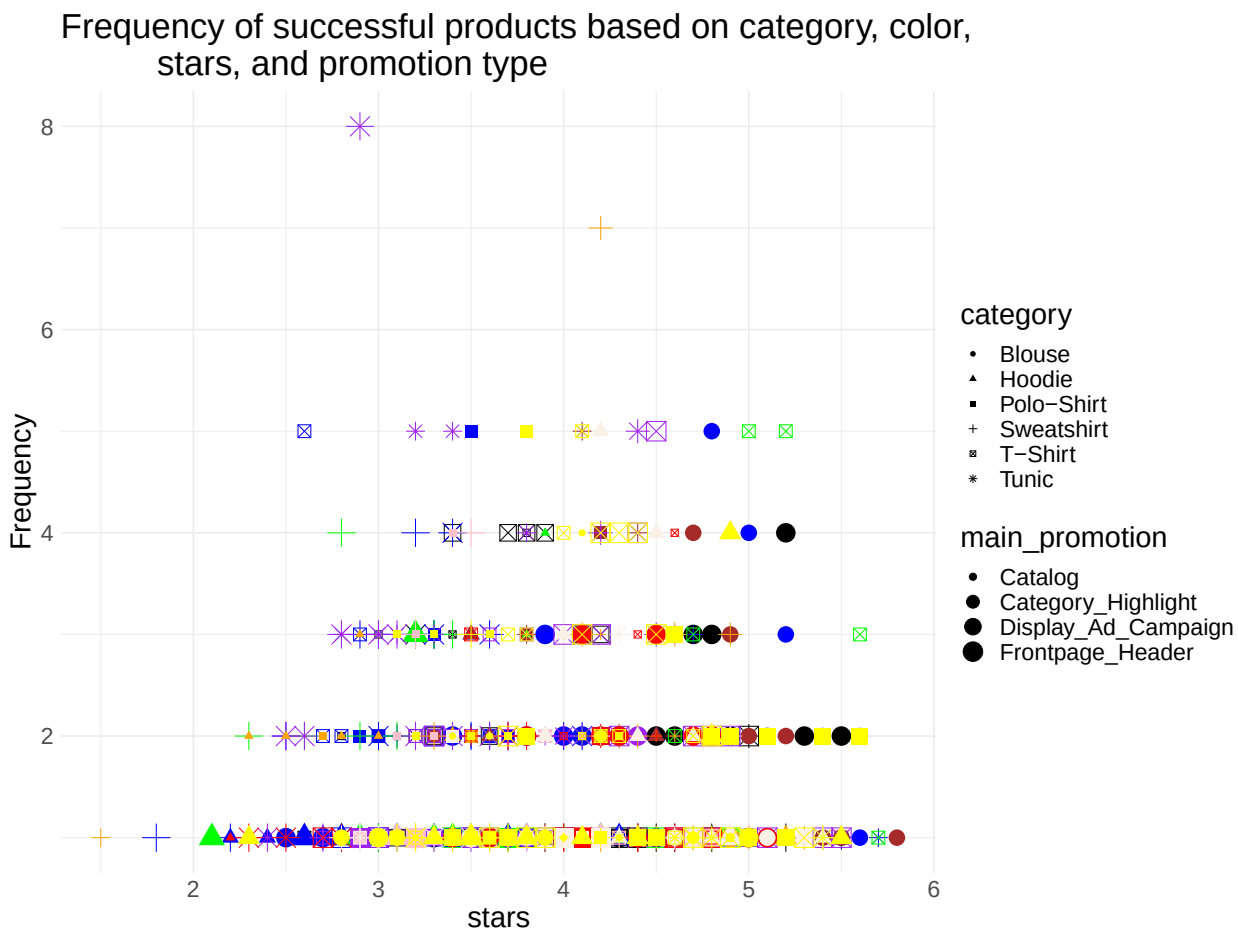
```
# Percentage flop
percntg_top <- ((sum(tab4[2]))/sum(tab4))*100
percntg_top
```

```
## [1] 69
```

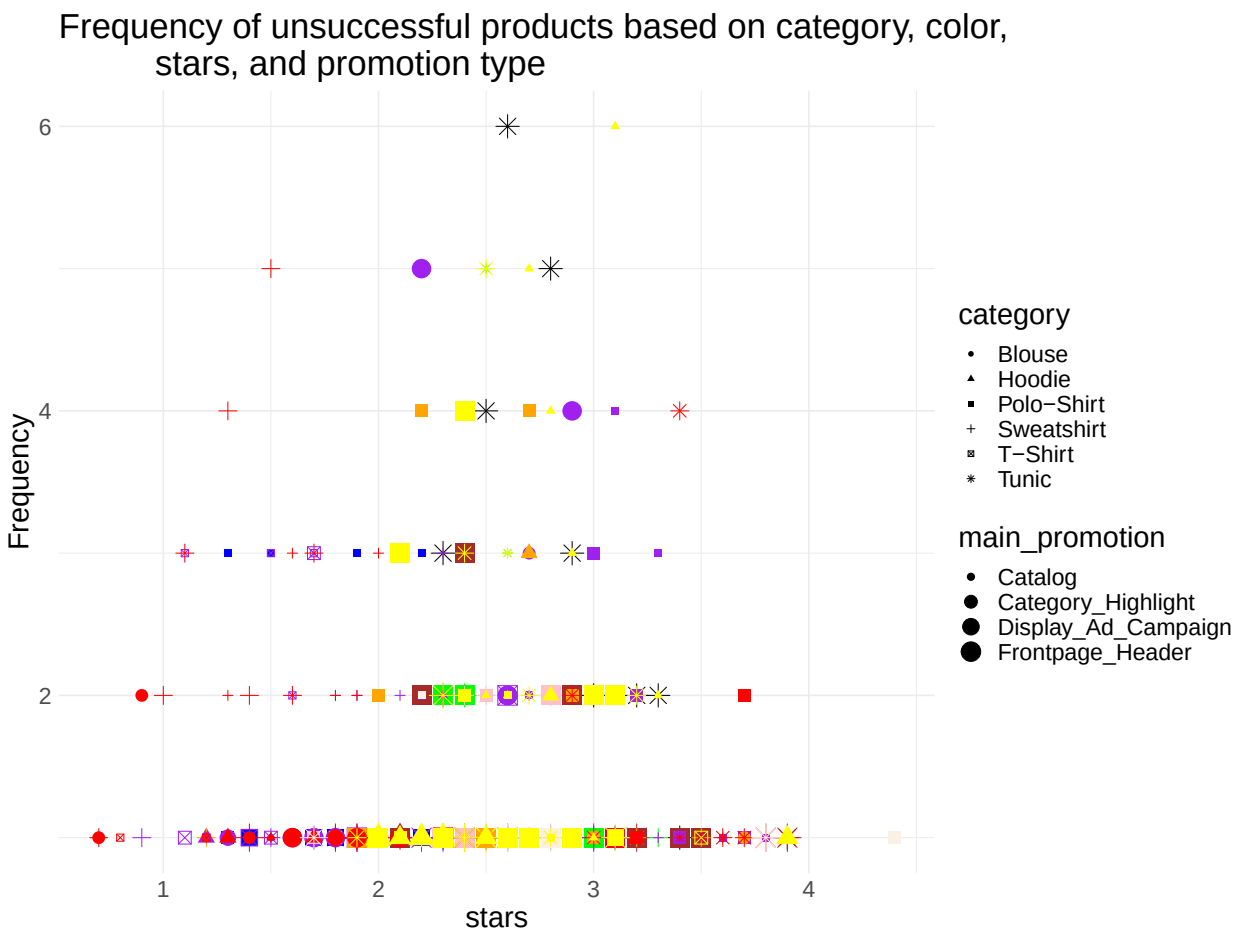
```
# Plot the results

# Change data for plots
df_pred_new <- df_pred %>% group_by(color, category, main_promotion, stars,
                                    pred_rf_new) %>% tally()
# Lowercase color naes
df_pred_new$color <- tolower(df_pred_new$color)
# change multi color and white to purple and linen for plots
df_pred_new$color <- ifelse(df_pred_new$color == "multi-color","purple",
                            df_pred_new$color)
df_pred_new$color <- ifelse(df_pred_new$color == "white","linen",
                            df_pred_new$color)
# get color columns as a new vector
col <- df_pred_new$color
# Plot Frequency of potential products sold based on category, color,and stars
ggplot(df_pred_new, aes(x=stars, y=n, shape=category,
                    size=pred_rf_new)) +
  geom_point(color=col) + ylab("Frequency") +
  ggtitle("Frequency of potential products sold based on category, color,and stars") +
  theme_minimal() +
  theme(text = element_text(size = 20))
```




```
#~~~~~~~~~~~~~~~~~~~~~~~~~
# Change data for plots
df_pred_new_top <-df_pred_new %>% subset(.,pred_rf_new == "top")
# get color columns as a new vector
col <- df_pred_new_top$color
# Plot Frequency of successful potential products based on category, color,
# stars, and promotion type
ggplot(df_pred_new_top, aes(x=stars, y=n, shape=category,
                    size=main_promotion)) +
  geom_point(color=col) + ylab("Frequency") +
  ggtitle("Frequency of successful products based on category, color,
          stars, and promotion type") + theme_minimal()+
  theme(text = element_text(size = 20))
```




```
#~~~~~~~~~~~~~~~~~~~~~~~~~
# Change data for plots
df_pred_new_flop <-df_pred_new %>% subset(.,pred_rf_new == "flop")
# get color columns as a new vector
col <- df_pred_new_flop$color
# Plot Frequency of unsuccessful potential products based on category, color,
# stars, and promotion type
ggplot(df_pred_new_flop, aes(x=stars, y=n, shape=category,
                             size = main_promotion)) +
  geom_point(color=col) + ylab("Frequency") +
  ggtitle("Frequency of unsuccessful products based on category, color,
          stars, and promotion type") + theme_minimal()+
  theme(text = element_text(size = 20))
```


Frequency of unsuccessful products based on category, color, stars, and promotion type
Frequency
6
4
2
1
2
3
4
stars
category
Blouse
Hoodie
Polo−Shirt
Sweatshirt
T−Shirt
Tunic
main_promotion
Catalog
Category_Highlight
Display_Ad_Campaign
Frontpage_Header