Facebook Ad-Sales Analysis

Introduction

This report delves into the effectiveness of Facebook ad campaigns, aiming to identify the critical factors that drive ad conversions. By analyzing ad features such as demographic information, user interests, and engagement metrics, we aim to develop predictive models that classify ads based on their conversion rates. The insights garnered from this analysis will assist marketers in optimizing their ad campaigns for better performance and return on investment.

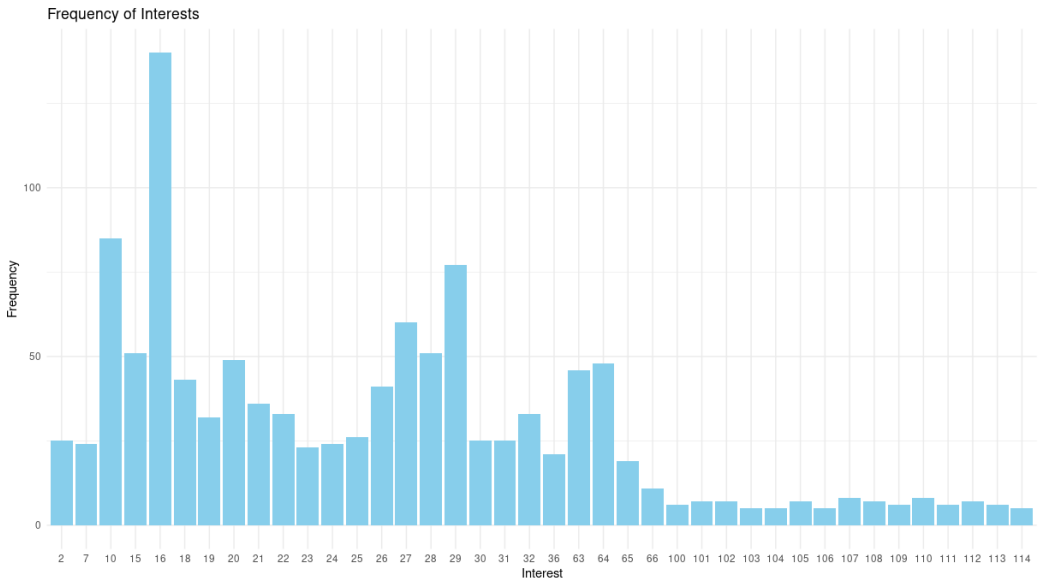
Dataset

The dataset consists of 1,143 records detailing various attributes of Facebook ads. These attributes include:

- **Demographic Information:** Age and gender of the target audience.

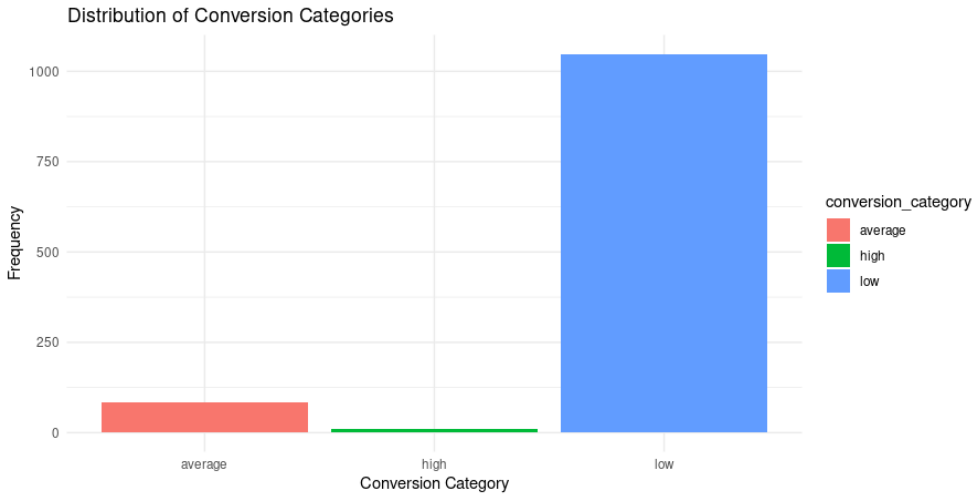
	F	M
30-34	0.17235346	0.20034996
35-39	0.09536308	0.12160980
40-44	0.09361330	0.09011374
45-49	0.12073491	0.10586177

- **User Interests:** Categorized interests that align with the target audience.



- **Engagement Metrics:** Includes Impressions (number of times an ad is viewed), Clicks (number of user interactions), and Spent (amount of money spent on the campaign).

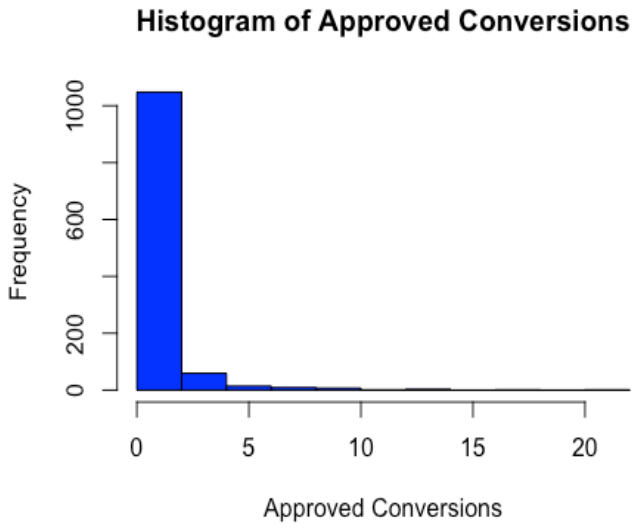
- **Conversion Metrics:** Total conversions and approved conversions that directly measure the effectiveness of the campaigns.



- **Non-essential ID:** Initial steps in the analysis involved removing non-essential ID columns to focus more on the attributes that potentially influence ad conversions.

```
     age              gender            interest       Impressions          Clicks            Spent        Total_Conversion Approved_Conversion
 Length:1143        Length:1143        Min.   :  2.00   Min.   :     87   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000   Min.   : 0.000
 Class :character   Class :character   1st Qu.: 16.00   1st Qu.:   6504   1st Qu.:  1.00   1st Qu.:  1.48   1st Qu.: 1.000   1st Qu.: 0.000
 Mode  :character   Mode  :character   Median : 25.00   Median :  51509   Median :  8.00   Median : 12.37   Median : 1.000   Median : 1.000
                                       Mean   : 32.77   Mean   : 186732   Mean   : 33.39   Mean   : 51.36   Mean   : 2.856   Mean   : 0.944
                                       3rd Qu.: 31.00   3rd Qu.: 221769   3rd Qu.: 37.50   3rd Qu.: 60.02   3rd Qu.: 3.000   3rd Qu.: 1.000
                                       Max.   :114.00   Max.   :3052003   Max.   :421.00   Max.   :639.95   Max.   :60.000   Max.   :21.000
```

- **Histogram of Approved Conversions:** This histogram provides a visual representation of the distribution of approved conversions across the dataset, which was instrumental in understanding the frequency and range of conversion counts.

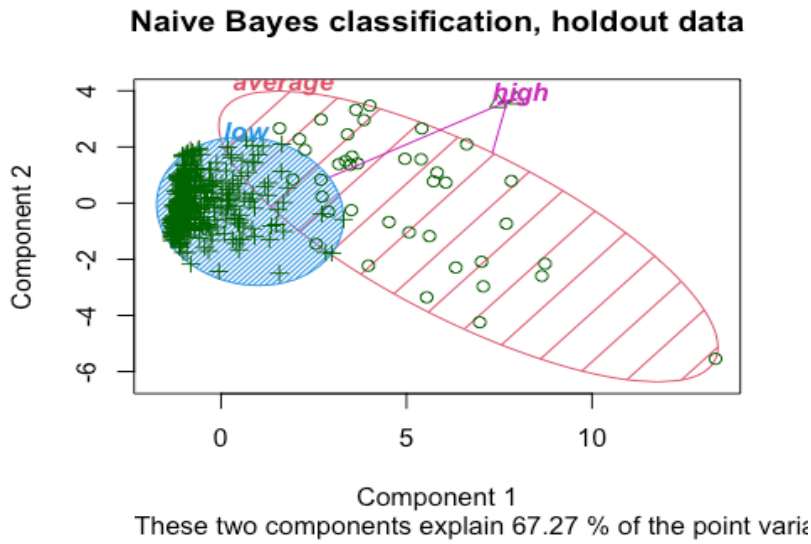


Analysis

We removed the approved and total conversions and considered conversions_category (low, average, high) in order to predict the best factors that drive conversion.

Naive Bayes Classifier

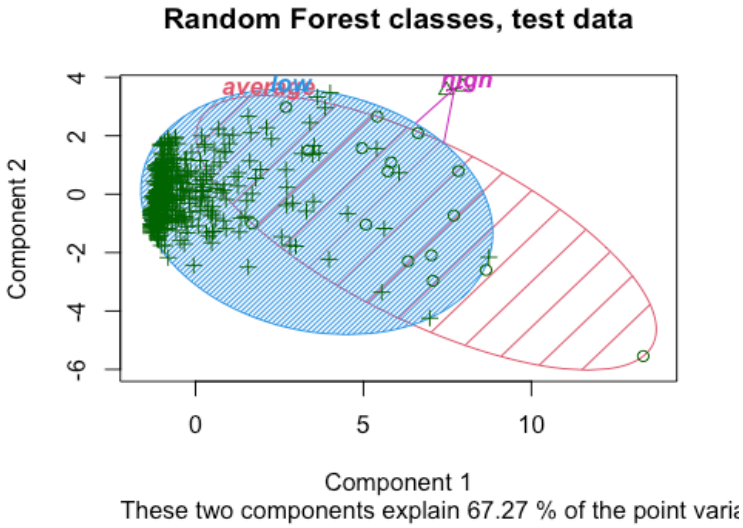
- **Model Execution:** The Naive Bayes classifier was trained using a subset of the data, focusing on predicting three categories of conversion: low, average, and high.
- **Model Evaluation:** The accuracy of the model on the test data was 89.77%, with an adjusted Rand Index of 0.4193, indicating a good match between the model's predictions and the actual data.
- **Visualization:** The clusplot for the Naive Bayes classification highlights the clustering of predicted categories in the multidimensional space of ad features.



Random Forest Classifier

- **Model Execution:** The Random Forest classifier was also trained on the dataset with the same objective as the Naive Bayes model. It utilized 11,430 trees to ensure robust predictions.
- **Model Evaluation:** The Random Forest model achieved an exceptional accuracy rate of 91.27% on the test data, with an adjusted Rand Index of 0.311.
- **Variable Importance:** Insights into feature importance revealed that **Clicks, Spent, and Impressions** are critical drivers of conversions, with Impressions being the most influential. This is visually represented in the Variable Importance Plot and further detailed in the heatmap analysis.

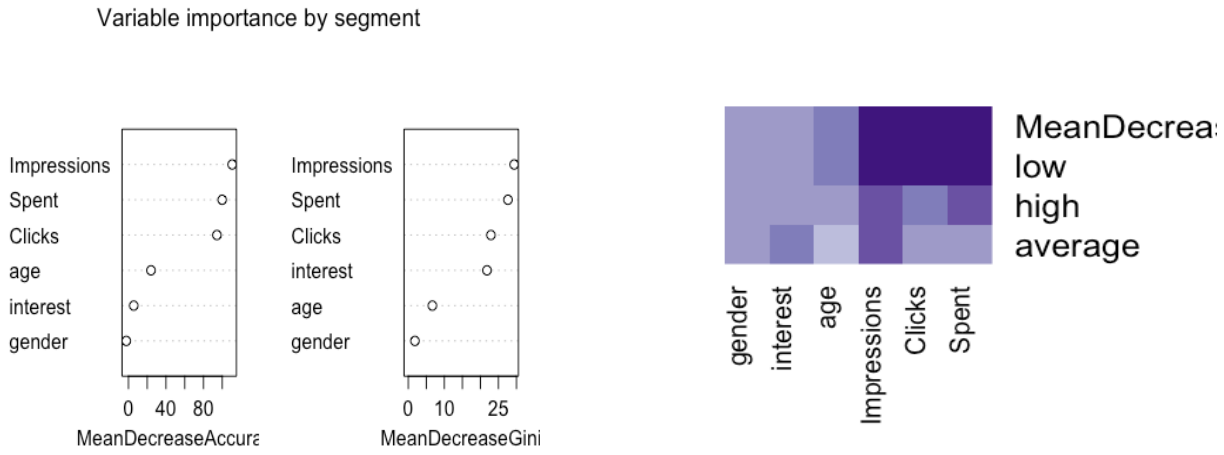
- **Visualization:** The heatmap of variable importance illustrates the relative importance of features in influencing ad conversions, highlighting the dominance of Clicks, Spent, and Impressions.



Conclusion

The comprehensive analysis using Naive Bayes and Random Forest classifiers indicates that **Impressions, Spent, and Clicks** are paramount in driving the success of Facebook ad campaigns. The Random Forest model, with its high accuracy and detailed insight into feature importance, particularly highlights the critical role of these variables in influencing ad conversions.

Given these insights, marketers are advised to focus on strategies that enhance user engagement through Impressions, optimize ad spending efficiently, and maximize the clicks of their ads to improve overall conversion rates.

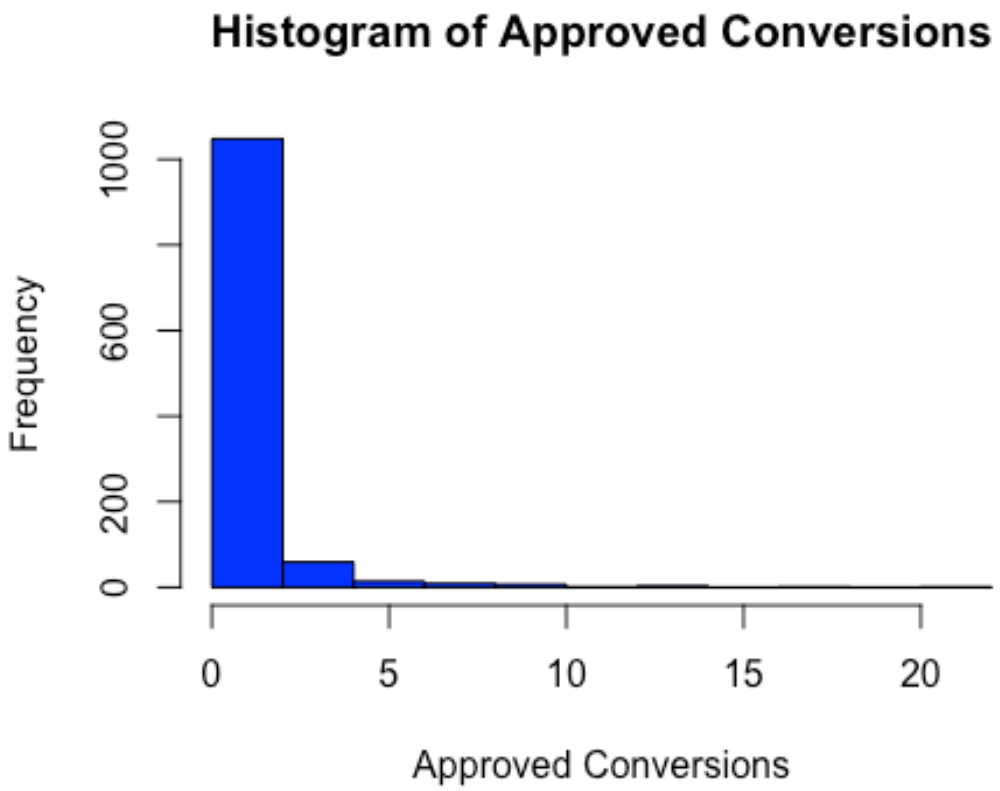


Appendix

```
#Understanding the dataset
fb.df <- read.csv("facebook.csv")
dim(fb.df)
## [1] 1143   11
summary(fb.df)
##      ad_id        xyz_campaign_id fb_campaign_id       age
##  Min.   : 708746   Min.   : 916    Min.   :103916   Length:1143
##  1st Qu.: 777632   1st Qu.: 936    1st Qu.:115716   Class :character
##  Median :1121185   Median :1178    Median :144549   Mode  :character
##  Mean   : 987261   Mean   :1067    Mean   :133784
##  3rd Qu.:1121804   3rd Qu.:1178    3rd Qu.:144658
##  Max.   :1314415   Max.   :1178    Max.   :179982
##     gender            interest       Impressions         Clicks
##  Length:1143        Min.   :  2.00   Min.   :    87   Min.   :  0.00
##  Class :character   1st Qu.: 16.00   1st Qu.:   6504   1st Qu.:  1.00
##  Mode  :character   Median : 25.00   Median :  51509   Median :  8.00
##                     Mean   : 32.77   Mean   : 186732   Mean   : 33.39
##                     3rd Qu.: 31.00   3rd Qu.: 221769   3rd Qu.: 37.50
##                     Max.   :114.00   Max.   :3052003   Max.   :421.00
##      Spent        Total_Conversion Approved_Conversion
##  Min.   :  0.00   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:  1.48   1st Qu.: 1.000   1st Qu.: 0.000
##  Median : 12.37   Median : 1.000   Median : 1.000
##  Mean   : 51.36   Mean   : 2.856   Mean   : 0.944
##  3rd Qu.: 60.02   3rd Qu.: 3.000   3rd Qu.: 1.000
##  Max.   :639.95   Max.   :60.000   Max.   :21.000
str(fb.df)
## 'data.frame':    1143 obs. of  11 variables:
##  $ ad_id             : int  708746 708749 708771 708815 708818 708820 708889 708895 70895
3 708958 ...
##  $ xyz_campaign_id    : int  916 916 916 916 916 916 916 916 916 916 ...
##  $ fb_campaign_id     : int  103916 103917 103920 103928 103928 103929 103940 103941 1
03951 103952 ...
##  $ age               : chr  "30-34" "30-34" "30-34" "30-34" ...
##  $ gender             : chr  "M" "M" "M" "M" ...
##  $ interest          : int  15 16 20 28 28 29 15 16 27 28 ...
##  $ Impressions        : int  7350 17861 693 4259 4133 1915 15615 10951 2355 9502 ...
##  $ Clicks            : int  1 2 0 1 1 0 3 1 1 3 ...
##  $ Spent             : num  1.43 1.82 0 1.25 1.29 ...
##  $ Total_Conversion   : int  2 2 1 1 1 1 1 1 1 1 ...
##  $ Approved_Conversion: int  1 0 0 0 1 1 0 1 0 0 ...
#Creating factors
fb.df$age <- factor(fb.df$age)
fb.df$gender <- factor(fb.df$gender)
```

```
fb.df$interest <- factor(fb.df$interest)
summary(fb.df)
##      ad_id         xyz_campaign_id fb_campaign_id      age      gender
##  Min.   : 708746   Min.   : 916    Min.   :103916   30-34:426   F:551
##  1st Qu.: 777632   1st Qu.: 936    1st Qu.:115716   35-39:248   M:592
##  Median :1121185   Median :1178    Median :144549   40-44:210
##  Mean   : 987261   Mean   :1067    Mean   :133784   45-49:259
##  3rd Qu.:1121804   3rd Qu.:1178    3rd Qu.:144658
##  Max.   :1314415   Max.   :1178    Max.   :179982
##
##    interest    Impressions         Clicks          Spent
##  16     :140   Min.   :    87   Min.   :  0.00   Min.   :  0.00
##  10     : 85   1st Qu.:   6504   1st Qu.:  1.00   1st Qu.:  1.48
##  29     : 77   Median :  51509   Median :  8.00   Median : 12.37
##  27     : 60   Mean   : 186732   Mean   : 33.39   Mean   : 51.36
##  15     : 51   3rd Qu.: 221769   3rd Qu.: 37.50   3rd Qu.: 60.02
##  28     : 51   Max.   :3052003   Max.   :421.00   Max.   :639.95
##  (Other):679
##  Total_Conversion Approved_Conversion
##  Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 1.000   1st Qu.: 0.000
##  Median : 1.000   Median : 1.000
##  Mean   : 2.856   Mean   : 0.944
##  3rd Qu.: 3.000   3rd Qu.: 1.000
##  Max.   :60.000   Max.   :21.000
##
str(fb.df)
## 'data.frame':    1143 obs. of  11 variables:
##  $ ad_id              : int  708746 708749 708771 708815 708818 708820 708889 708895 70895
3 708958 ...
##  $ xyz_campaign_id    : int  916 916 916 916 916 916 916 916 916 916 ...
##  $ fb_campaign_id     : int  103916 103917 103920 103928 103928 103929 103940 103941 1
03951 103952 ...
##  $ age                : Factor w/ 4 levels "30-34","35-39",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ gender             : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ interest           : Factor w/ 40 levels "2","7","10","15",..: 4 5 8 16 16 17 4 5 15 16 ...
##  $ Impressions        : int  7350 17861 693 4259 4133 1915 15615 10951 2355 9502 ...
##  $ Clicks             : int  1 2 0 1 1 0 3 1 1 3 ...
##  $ Spent              : num  1.43 1.82 0 1.25 1.29 ...
##  $ Total_Conversion   : int  2 2 1 1 1 1 1 1 1 1 ...
##  $ Approved_Conversion: int  1 0 0 0 1 1 0 1 0 0 ...
#Visualize the Approved conversion using histogram
hist(fb.df$Approved_Conversion,
    main = "Histogram of Approved Conversions",
    xlab = "Approved Conversions",
    ylab = "Frequency",
```

```
    col = "blue",
    breaks = 10)
```



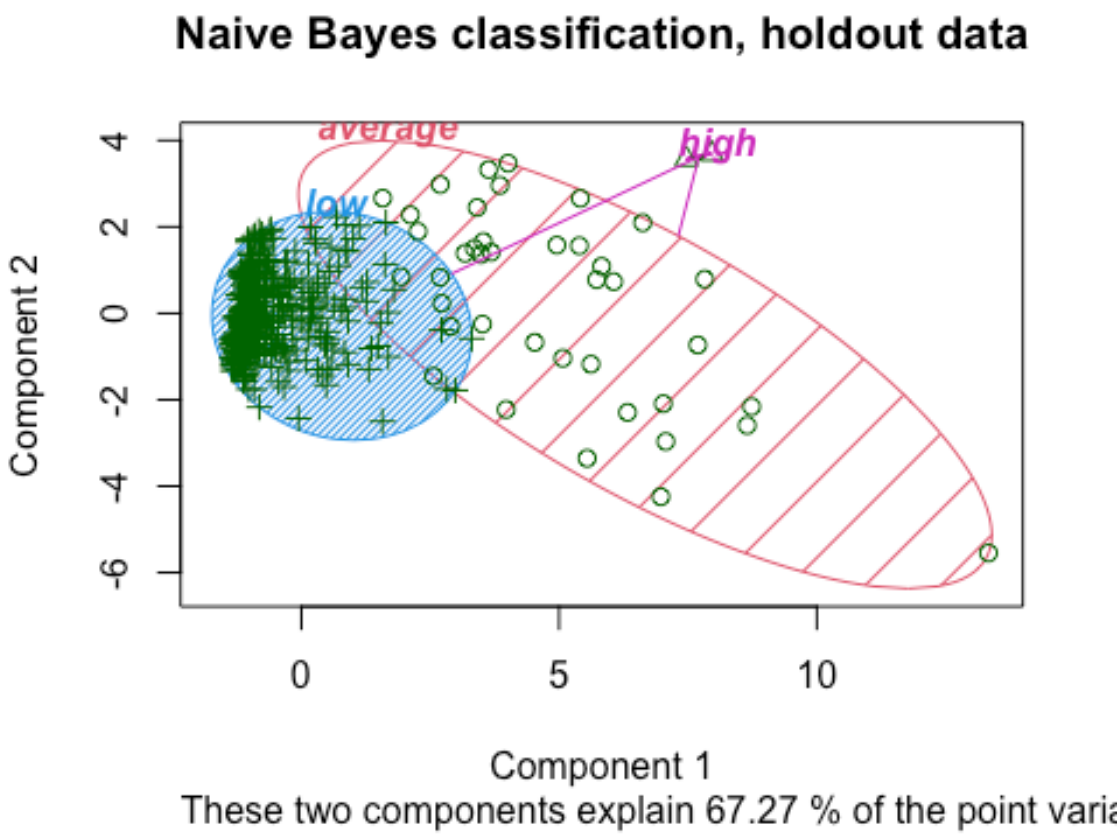

```
#Remove the ID columns
fb.cl <- fb.df[,-(1:3)]
summary(fb.cl)
##    age      gender     interest    Impressions         Clicks
##  30-34:426   F:551   16     :140   Min.   :    87   Min.   : 0.00
##  35-39:248   M:592   10     : 85   1st Qu.:  6504   1st Qu.: 1.00
##  40-44:210           29     : 77   Median :  51509   Median : 8.00
##  45-49:259           27     : 60   Mean   : 186732   Mean   : 33.39
##                      15     : 51   3rd Qu.: 221769   3rd Qu.: 37.50
##                      28     : 51   Max.   :3052003   Max.   :421.00
##                      (Other):679
##      Spent        Total_Conversion Approved_Conversion
##  Min.   :  0.00   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:  1.48   1st Qu.: 1.000   1st Qu.: 0.000
##  Median : 12.37   Median : 1.000   Median : 1.000
##  Mean   : 51.36   Mean   : 2.856   Mean   : 0.944
##  3rd Qu.: 60.02   3rd Qu.: 3.000   3rd Qu.: 1.000
##  Max.   :639.95   Max.   :60.000   Max.   :21.000
##
```

```
str(fb.cl)
## 'data.frame':    1143 obs. of  8 variables:
##  $ age                : Factor w/ 4 levels "30-34","35-39",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ gender             : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ interest           : Factor w/ 40 levels "2","7","10","15",..: 4 5 8 16 16 17 4 5 15 16 ...
##  $ Impressions        : int  7350 17861 693 4259 4133 1915 15615 10951 2355 9502 ...
##  $ Clicks             : int  1 2 0 1 1 0 3 1 1 3 ...
##  $ Spent              : num  1.43 1.82 0 1.25 1.29 ...
##  $ Total_Conversion   : int  2 2 1 1 1 1 1 1 1 1 ...
##  $ Approved_Conversion: int  1 0 0 0 1 1 0 1 0 0 ...
#Create a new column for approved conversion
fb.cl$conversion_category <- factor(ifelse(fb.cl$Approved_Conversion <= 2, "low",
                           ifelse(fb.cl$Approved_Conversion < 10, "average", "high")))
str(fb.cl)
## 'data.frame':    1143 obs. of  9 variables:
##  $ age                : Factor w/ 4 levels "30-34","35-39",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ gender             : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ interest           : Factor w/ 40 levels "2","7","10","15",..: 4 5 8 16 16 17 4 5 15 16 ...
##  $ Impressions        : int  7350 17861 693 4259 4133 1915 15615 10951 2355 9502 ...
##  $ Clicks             : int  1 2 0 1 1 0 3 1 1 3 ...
##  $ Spent              : num  1.43 1.82 0 1.25 1.29 ...
##  $ Total_Conversion   : int  2 2 1 1 1 1 1 1 1 1 ...
##  $ Approved_Conversion: int  1 0 0 0 1 1 0 1 0 0 ...
##  $ conversion_category: Factor w/ 3 levels "average","high",..: 3 3 3 3 3 3 3 3 3 3 ...
summary(fb.cl)
##    age      gender     interest    Impressions         Clicks
##  30-34:426   F:551   16     :140   Min.   :    87   Min.   : 0.00
##  35-39:248   M:592   10     : 85   1st Qu.:  6504   1st Qu.: 1.00
##  40-44:210           29     : 77   Median :  51509   Median : 8.00
##  45-49:259           27     : 60   Mean   : 186732   Mean   : 33.39
##                      15     : 51   3rd Qu.: 221769   3rd Qu.: 37.50
##                      28     : 51   Max.   :3052003   Max.   :421.00
##                      (Other):679
##      Spent        Total_Conversion Approved_Conversion conversion_category
##  Min.   :  0.00   Min.   : 0.000   Min.   : 0.000      average:  85
##  1st Qu.:  1.48   1st Qu.: 1.000   1st Qu.: 0.000      high   :  10
##  Median : 12.37   Median : 1.000   Median : 1.000      low    :1048
##  Mean   : 51.36   Mean   : 2.856   Mean   : 0.944
##  3rd Qu.: 60.02   3rd Qu.: 3.000   3rd Qu.: 1.000
##  Max.   :639.95   Max.   :60.000   Max.   :21.000
##
#Pre-modeling steps
set.seed(123)
train.prop  <- 0.65
train.cases  <- sample(nrow(fb.cl), nrow(fb.cl)*train.prop)
train.cases
```

```
##   [1]  415  463  179  526  195  938 1038  665  602  709 1011 1115  953  348 1017
##  [16]  840   26  519  211  932  593  555  373  844  544  490  905  937 1047  923
##  [31]  956  309  166  217  581   72  588  141  722  859  153  294  277   41  431
##  [46]   90  316  528  774  747  456  598 1063  752  374   34  516   13   69  755
##  [61]  409  928 1006  537  983  291  671  121  480   67 1014  165  236  610  330
##  [76]  726  127  212  686  814  931  878  243  847 1079  619  477  151  666  767
##  [91]  160  155  426 1029  326  789  985   39  822  986  137  455  589 1046  196
## [106]  680  500 1132  344  459  944  872 1139  876  534  177  554  827   84  523
## [121]  633  951  392  302  597  877  706 1058 1041  430  710  761  712  428  672
## [136]  250  804  429  398 1096 1059  381  545   40  936  522  473  200  978  125
## [151]  265  775  903  186  573  252  458  152  831   54  919  538  235  289  185
## [166]  765  413  627  998 1112  984  205  875  779 1080  564  794  391 1083  727
## [181]  346 1053  468  509  920   57  457  617  357  279  270 1062  646  347  129
## [196]  218  618  881  698  337  797 1127  539 1126  757 1055  553  724  390  498
## [211]  222 1077  861  657  421  958  660  163  238  673  578 1087 1069  225  389
## [226]  117  969  648   55 1110 1019  557  658  682 1143  134  711  873  688  939
## [241]  447  821  104  902 1036  907  961  210  349  401  737  258 1028  386 1106
## [256]   24  466  130 1072  703 1107  377  781  170  445 1013  874  234  422  508
## [271]   64   80  483  548  475 1078  995  343  323  479  560  450  111  791  317
## [286]  807  933  287  734  585  292  226  790  684  297  605  637  811 1040  237
## [301]  885 1016   33   83  396 1101  209   76   94  803   30  914  175 1089  865
## [316]  115  882  608  465  358  910  424   96  982  397  404  742  148 1021  714
## [331] 1123  991  338  744 1061  106   11  625  364  890  403  461  889   31  828
## [346]  851  835 1064   16 1134  420  178  417  464  412  891  524  437 1114  562
## [361]  204  899 1121 1022  798  384  122  399  834  315  259  494  760  980   48
## [376]  331  100  108  301   10  880  280 1130  402  837  897  649  974  395  766
## [391]    8  626  261  541  306 1049   74  282  854  267  262  736  783  723  219
## [406]  696  352  667  119  452  579  622  780   36  915 1118  240  632  304  764
## [421] 1136  971  612  105  388 1142  180  278  946  241  888  679  559  884   37
## [436] 1065  566  303   19 1124  378  549  615  244  769  188  970  393  139  299
## [451]  371  670  189  311  691  418  569  382   38 1025  319  771  642  120  768
## [466]  533  441  199  499  599 1108 1095   81  729  908 1045    6  128   49  476
## [481]  239  340  193 1034 1129  816  190  624  191  446  735  976  975  652   59
## [496]   61 1105  754  853  715  535  799  600  977  203  246  440  251  863 1001
## [511]  162  322  576  168  442   78  493  676  668  879   95  697  379  342  221
## [526]  161  448  242  536  407  229  224  474  677  171   23  948  484  867   79
## [541]  511  507  164 1120  733  647  650  759  824  595 1100  949  862  232  334
## [556]  839  813  264  201   52  959 1074  577  957  750   20  206  124   45  332
## [571]  281   91  138 1067  425  753  271  167  675  255   71  704  786  561 1131
## [586] 1048 1043   46  906  531  613  592  220  857 1135  777 1133  502  606  868
## [601]  169  812  341 1085  320  504  836    2  808 1091  830  924  449 1116 1024
## [616]  335   56  567  207  436  645  387  263   68 1137  955  256  895  471   88
## [631]  416  345  858 1094  718  809  945  856  921  173 1009  762 1119  616  266
## [646] 1122  328  172  298  962  707  966 1111   17  758  778  860  530  987  365
## [661]  485 1007  604  515  367  227 1068  748  110  503 1099 1018  993  826  989
## [676]  776  112 1093   93  838  653  699  580  248  720  609  184    4  327  312
```

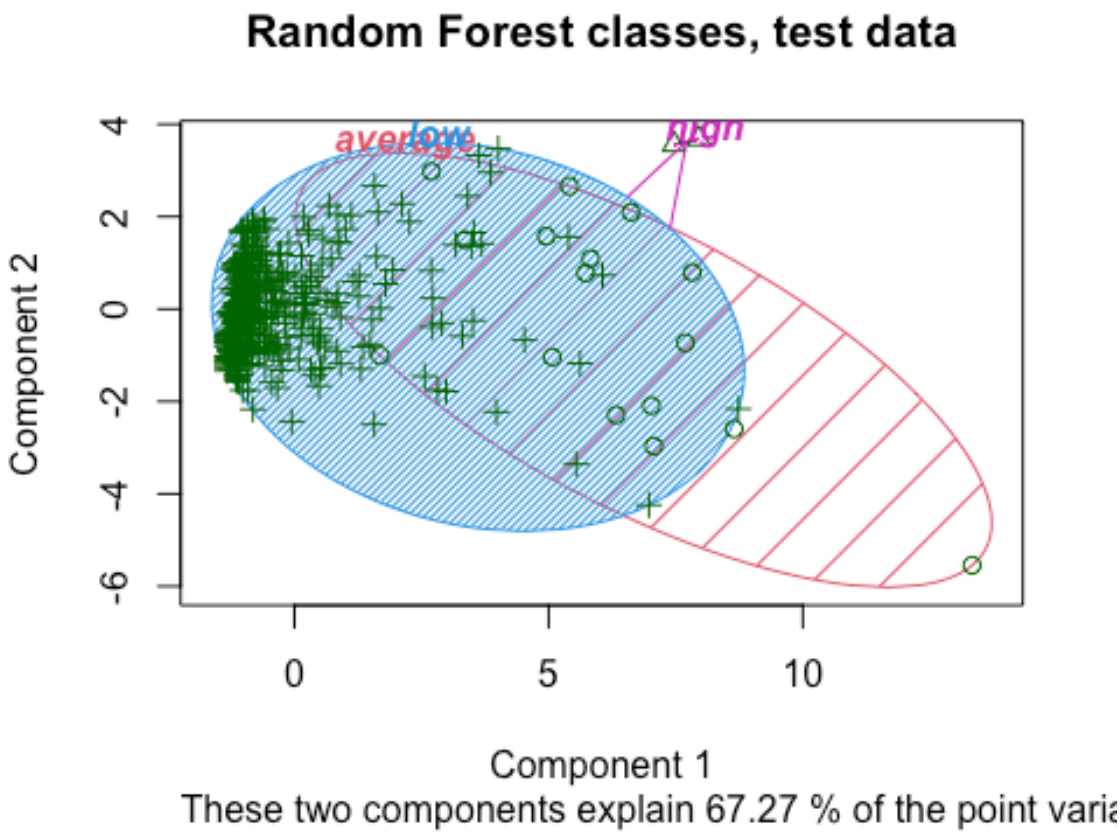
```
## [691]   50  380  695  638  547 1097 1008  230   85  505 1076  114 1031   99   98
## [706]   58  806 1086  568  892  123  743   87  728  513 1109 1090   77  497  594
## [721] 1030 1073 1042  692  855  705  796 1140 1005  607 1052  596  841  749  274
## [736]  911  950  495  510   53  683  329
fb.cl.train <- fb.cl[ train.cases, ]
fb.cl.test  <- fb.cl[-train.cases, ]
fb.cl.train<-fb.cl.train[,-c(7,8)]

#Let's check the details of sampling
nrow(fb.cl)
## [1] 1143
nrow(fb.cl)*train.prop
## [1] 742.95
nrow(fb.cl.train)
## [1] 742
nrow(fb.cl.test)
## [1] 401
#Naive Bayes
library(e1071)
fb.nb<-naiveBayes(conversion_category~.,data = fb.cl.train)
fb.nb.class<-predict(fb.nb,fb.cl.test)
fb.nb.class
##   [1] low     low     low     low     low     low     low     low     low
##  [10] low     low     low     low     low     low     low     low     low
##  [19] low     low     low     low     low     low     low     low     low
##  [28] low     low     low     low     low     low     low     low     low
##  [37] low     low     low     low     low     low     low     low     low
##  [46] low     low     low     low     low     low     low     low     low
##  [55] low     average low     low     low     low     low     low     low
##  [64] low     low     low     low     low     low     low     low     low
##  [73] low     low     average low     low     low     low     low     low
##  [82] low     low     low     low     low     low     low     low     low
##  [91] low     low     low     low     low     low     low     low     low
## [100] low     low     low     low     low     low     low     low     low
## [109] low     low     low     low     low     low     low     low     low
## [118] low     low     low     low     low     low     low     low     low
## [127] low     low     low     low     low     low     low     low     low
## [136] low     low     low     low     low     low     low     low     low
## [145] low     low     low     low     low     low     low     low     low
## [154] low     low     low     low     low     low     low     low     low
## [163] low     low     low     low     low     low     low     low     low
## [172] low     low     low     low     low     low     low     low     low
## [181] low     low     low     low     low     average low     average average
## [190] low     low     low     low     low     low     low     low     low
## [199] low     low     low     low     low     low     average low     low
## [208] low     low     low     low     low     low     low     low     average
```

```
## [217] low     low     low     average average low     low     low     low
## [226] low     low     low     low     low     low     low     low     low
## [235] low     low     low     low     average low     low     low     low
## [244] low     low     low     low     low     low     low     average low
## [253] low     low     low     low     low     low     low     low     low
## [262] low     low     low     low     low     low     low     low     low
## [271] low     low     average low     low     low     low     low     low
## [280] low     low     low     low     average low     low     low     low
## [289] low     low     low     low     low     low     low     low     low
## [298] low     average low     average low     low     low     low     low
## [307] low     average low     low     average low     low     low     low
## [316] low     low     low     low     low     average average average average
## [325] low     low     low     low     low     low     low     low     average
## [334] average low     low     low     low     low     low     average low
## [343] low     low     low     low     low     average low     low     low
## [352] low     low     low     low     low     high    average low     average
## [361] average average average average low     low     low     average average
## [370] high    low     average low     low     low     average low     low
## [379] low     low     low     low     low     low     low     low     low
## [388] low     low     low     average average average low     low     low
## [397] average low     average low     low
## Levels: average high low
prop.table(table(fb.nb.class))
## fb.nb.class
##     average        high         low
## 0.099750623 0.004987531 0.895261845
#Naive Bayes - Plot the predicted classes
library(cluster)
clusplot(fb.cl.test[, -9], fb.nb.class, color=TRUE, shade=TRUE,
        labels=4,
        main="Naive Bayes classification, holdout data")
```

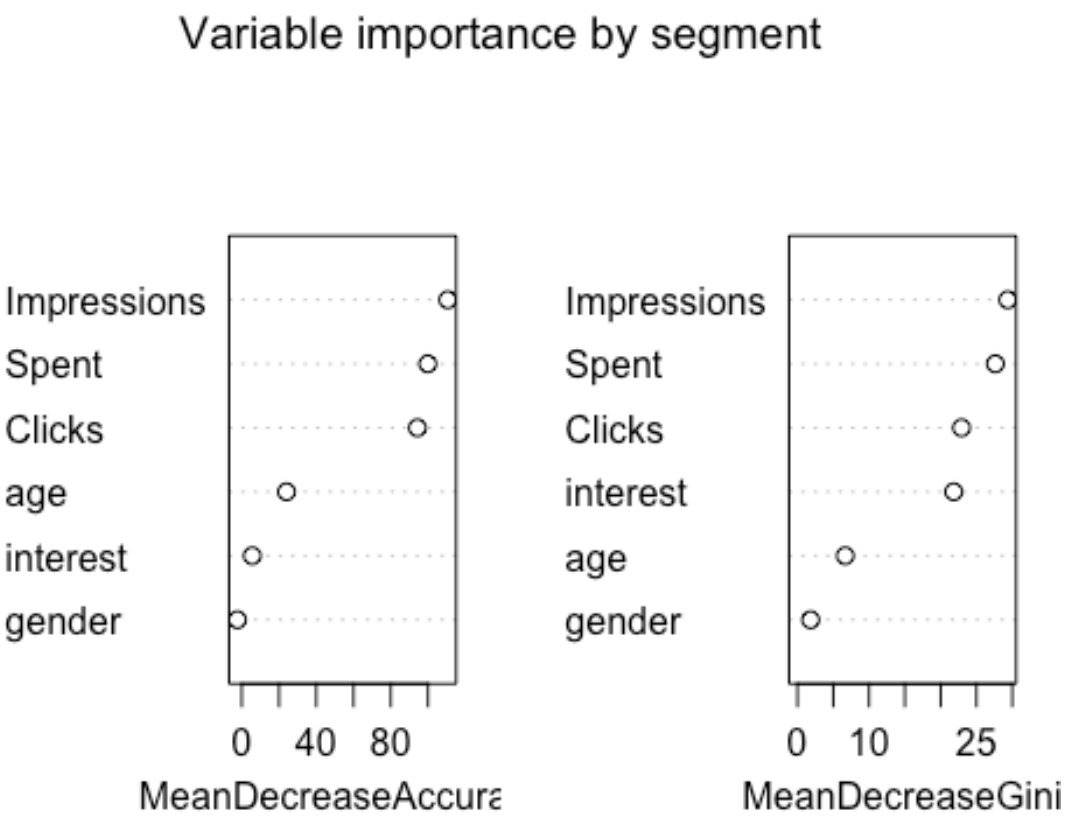



```
#How well did we predict in the test data?
mean(fb.cl.test$conversion_category==fb.nb.class)
## [1] 0.8977556
library(mclust)
## Package 'mclust' version 6.1
## Type 'citation("mclust")' for citing this R package in publications.
adjustedRandIndex(fb.nb.class,fb.cl.test$conversion_category)
## [1] 0.4193825
#Confusion Matrix
table(fb.nb.class,fb.cl.test$conversion_category)
##
## fb.nb.class average high low
##     average      16    4  20
##     high          0    0   2
##     low          15    0 344
#Random Forest
library(randomForest)
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
```

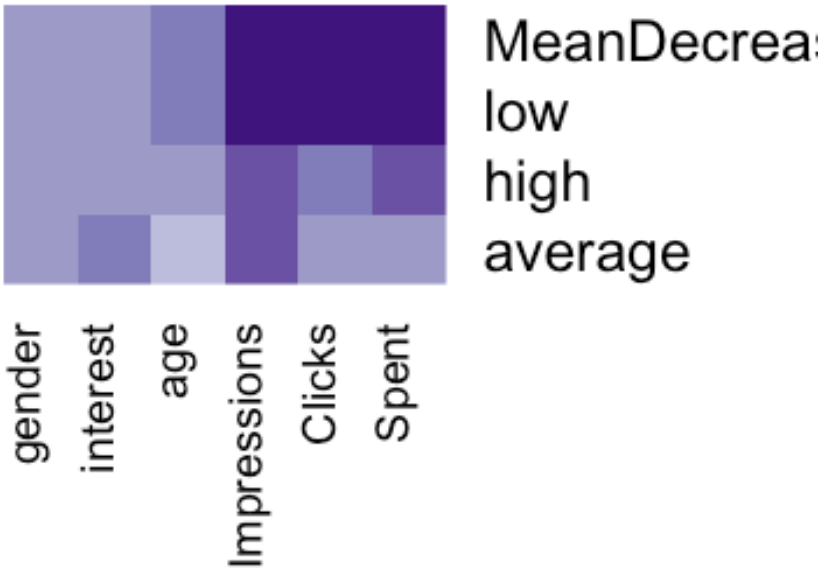
```r
set.seed(123)
fb.rf<-randomForest(conversion_category~.,data=fb.cl.train,ntree=11430)
fb.rf
##
## Call:
##  randomForest(formula = conversion_category ~ ., data = fb.cl.train,      ntree = 11430)
##                Type of random forest: classification
##                      Number of trees: 11430
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 7.82%
## Confusion matrix:
##         average high low class.error
## average      13    1  40  0.75925926
## high          2    3   1  0.50000000
## low          14    0 668  0.02052786
#Making Predictions
library(cluster)
fb.rf.class<-predict(fb.rf,fb.cl.test)
clusplot(fb.cl.test[, -9], fb.rf.class, color=TRUE, shade=TRUE,
        labels=4, main="Random Forest classes, test data")
```



```
#Individual prediction Probabilities
fb.rf.class.all <- predict(fb.rf, fb.cl.test, predict.all=TRUE)

#How well did we predict in the test data?
mean(fb.cl.test$conversion_category ==fb.rf.class)
## [1] 0.9127182
library(mclust)
adjustedRandIndex(fb.cl.test$conversion_category , fb.rf.class)
## [1] 0.3114741
#Confusion Matrix
table(fb.cl.test$conversion_category,fb.rf.class)
##          fb.rf.class
##           average high low
##   average       8    0  23
##   high          2    0   2
##   low           6    2 358
#Variable Importance
set.seed(123)
fb.rf.vi <- randomForest(conversion_category~., data = fb.cl.train, ntree=11430, importance=T
RUE)
importance(fb.rf.vi)
##                average      high       low MeanDecreaseAccuracy
## age         -12.706483  10.47963 32.9571523            24.003948
## gender       -2.550218 -11.41242  0.2367961            -2.337986
## interest     12.741816  11.24974 -2.5177716             5.634920
## Impressions  59.767155  59.46001 96.7181007           110.605294
## Clicks       -3.530428  32.70783 89.5264110            94.325478
## Spent         7.570335  55.91451 90.9448675            99.895645
##            MeanDecreaseGini
## age                6.677659
## gender             1.841500
## interest          21.830070
## Impressions       29.376355
## Clicks            22.925342
## Spent             27.649127
varImpPlot(fb.rf.vi,main = "Variable importance by segment")
```




```
#Heatmap
library(gplots)
##
## Attaching package: 'gplots'
## The following object is masked from 'package:stats':
##
##     lowess
library(RColorBrewer)
heatmap.2(t(importance(fb.rf.vi)[ , 1:4]), key=FALSE,
          col=brewer.pal(9, "Purples"),
          dend="none", trace="none", margins=c(10, 10),
          main="\n\n\n\n\n\n\n\n\nVar. importance by segment" )
```


MeanDecrea
low
high
average
gender
interest
age
Impressions
Clicks
Spent