C3T3

```
require(pacman)
```

```
## Loading required package: pacman
```

```
pacman:: p_load(pacman, dplyr, GGally, ggplot2, ggrepel, patchwork, gifski, ggforce, ggthemes, maps, sf
```

###Objective:

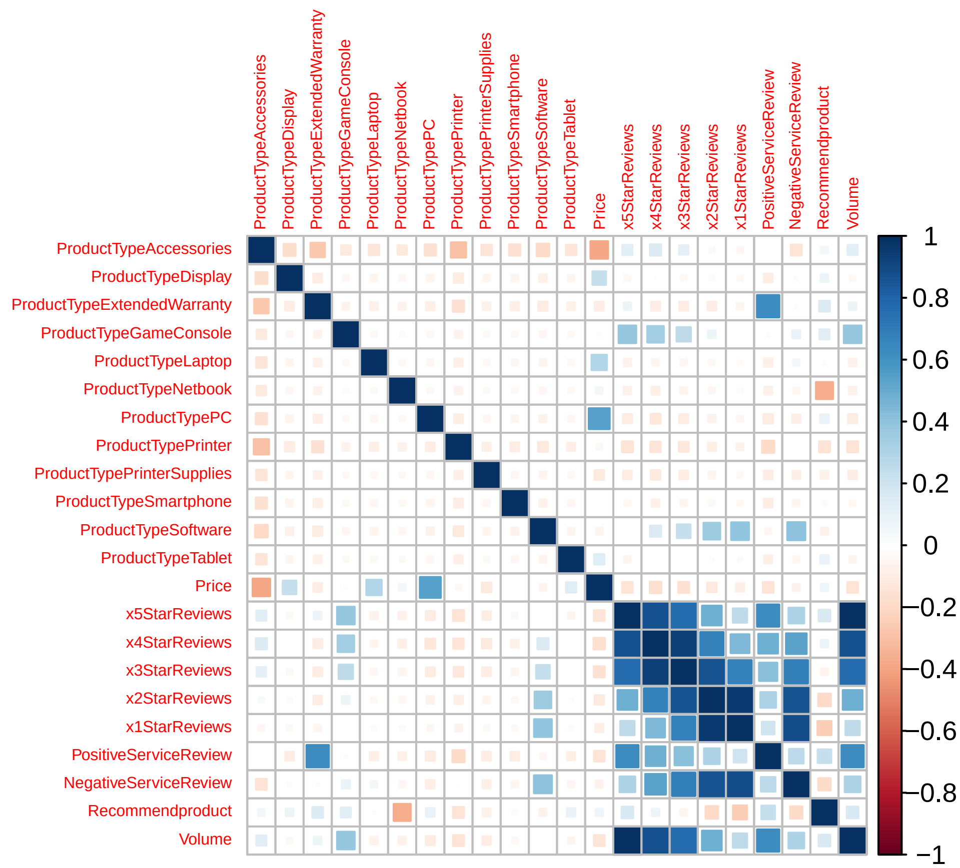
They have asked our team to analyze historical sales data and then make sales volume predictions for a list of new product types

- Predicting sales of four different product types: PC, Laptops, Netbooks and Smartphones.
- Assessing the impact services reviews and customer reviews have on sales of different product types. ###Importing Data

```
df1 <- import("existingproductattributes2017.csv")
df2 <- import("newproductattributes2017.csv")

#str(df1)
#names(df1)

df1 <- select(df1, -c(ProductNum, BestSellersRank, ProductWidth, ProductHeight, ProductDepth,ShippingWei

df2 <- select(df2, -c(ProductNum, BestSellersRank, ProductWidth, ProductHeight, ProductDepth,ShippingWei

#names(df1)
```

###Correlation Plot

```
Dummy <- dummyVars(" ~ .", data = df1)
df11 <- data.frame(predict(Dummy, newdata = df1))

Dummy2 <- dummyVars(" ~ .", data = df2)
df22 <- data.frame(predict(Dummy, newdata = df2))
#is.na(df11)
#explore(df11)
#df11
corrplot(cor(df11), method = "square", tl.cex=0.5)
```



###Building Models

```
set.seed(107)
inTrain <- createDataPartition(y = df11$Volume, p = .75, list = FALSE)
training <- df11[ inTrain,]
testing <- df11[-inTrain,]

ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3, classProbs = FALSE)

RF <- train(Volume~., data = training, method = "rf", tuneLength = 2, trControl=ctrl)
GBM = train(Volume ~., data=training, method="gbm", trControl=ctrl)
```

```
## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 6: ProductTypeNetbook has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 6: ProductTypeNetbook has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 6: ProductTypeNetbook has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 10: ProductTypeSmartphone has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 10: ProductTypeSmartphone has no variation.
```

```
## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 10: ProductTypeSmartphone has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 6: ProductTypeNetbook has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 6: ProductTypeNetbook has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 6: ProductTypeNetbook has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 10: ProductTypeSmartphone has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 10: ProductTypeSmartphone has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 10: ProductTypeSmartphone has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 10: ProductTypeSmartphone has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 10: ProductTypeSmartphone has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 10: ProductTypeSmartphone has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 6: ProductTypeNetbook has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 6: ProductTypeNetbook has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 6: ProductTypeNetbook has no variation.
```

```r
SVM <- train(Volume~., data = training, method = "svmLinear", trControl=ctrl, tuneLength = 5)
```

```
## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.
```

```r
importanceRF = varImp(RF, scale=TRUE)
importanceRF
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                              Overall
## x4StarReviews              1.000e+02
## x5StarReviews              9.505e+01
## PositiveServiceReview      1.928e+01
## x2StarReviews              1.683e+01
## x3StarReviews              6.179e+00
## ProductTypeGameConsole     3.166e+00
## x1StarReviews              2.278e+00
## NegativeServiceReview      3.866e-01
## Price                      2.370e-01
## ProductTypeAccessories     9.801e-02
## Recommendproduct           8.250e-02
## ProductTypeExtendedWarranty 5.941e-03
## ProductTypePrinter         3.034e-03
## ProductTypeLaptop          8.306e-04
## ProductTypeTablet          7.370e-04
## ProductTypeSoftware        3.509e-04
## ProductTypeSmartphone      2.027e-04
## ProductTypeDisplay         2.758e-05
## ProductTypePrinterSupplies 1.843e-05
## ProductTypeNetbook         1.561e-07
```

###RMSE of 1st Models

```r
RF
```

```
## Random Forest
##
## 61 samples
## 21 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 55, 57, 55, 54, 54, 55, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##    2    750.3515  0.8596402  398.7720
##   21    586.9938  0.9397717  262.6351
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 21.
```

```r
GBM
```

```
## Stochastic Gradient Boosting
##
## 61 samples
## 21 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 54, 53, 56, 55, 55, 56, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  RMSE       Rsquared   MAE
##   1                   50       983.0975  0.8197602  606.2506
##   1                  100      1069.7254  0.7980405  693.4259
##   1                  150      1102.3934  0.7688528  724.4534
##   2                   50       987.5413  0.8090297  601.5272
##   2                  100      1044.5612  0.7915764  675.9723
##   2                  150      1088.9529  0.7550649  714.3322
##   3                   50       982.3561  0.8134082  604.1387
##   3                  100      1084.7588  0.7805645  700.8077
##   3                  150      1094.7461  0.7537873  709.8546
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were n.trees = 50, interaction.depth =
##  3, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
SVM
```

```
## Support Vector Machines with Linear Kernel
##
## 61 samples
## 21 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 56, 56, 56, 55, 55, 54, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   181.8974  0.9701778  115.9242
##
## Tuning parameter 'C' was held constant at a value of 1
```

###2nd Try at Modelling

What if i get rid of the productype and price which dont have much impact/importance on the modelling? Second round of modelling:

```
RF2 <- train(Volume~ x5StarReviews + x4StarReviews  + x3StarReviews  + x2StarReviews + x1StarReviews + P

GBM2 = train(Volume ~x5StarReviews + x4StarReviews  + x3StarReviews  + x2StarReviews + x1StarReviews + P
```

```r
SVM2 <- train(Volume~x5StarReviews + x4StarReviews  + x3StarReviews  + x2StarReviews + x1StarReviews + P
```

```r
importanceRF2 = varImp(RF2, scale=TRUE)
importanceRF2
```

```
## rf variable importance
##
##                         Overall
## x4StarReviews          100.0000
## x5StarReviews           92.5605
## PositiveServiceReview   15.8241
## x2StarReviews           13.2118
## x3StarReviews            4.2520
## NegativeServiceReview    2.8367
## x1StarReviews            2.5669
## Recommendproduct         0.1323
## Price                    0.0000
```

###RMSE of 2nd Models

```r
RF2
```

```
## Random Forest
##
## 61 samples
##  9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 55, 54, 56, 56, 53, 56, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##   2     724.0560  0.9241494  333.8482
##   9     491.4904  0.9766840  212.0938
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 9.
```

```r
GBM2
```

```
## Stochastic Gradient Boosting
##
## 61 samples
##  9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 56, 54, 56, 56, 55, 53, ...
## Resampling results across tuning parameters:
##
```

```
##   interaction.depth  n.trees  RMSE       Rsquared   MAE
##   1                   50       958.3804  0.8335205  578.9721
##   1                  100      1017.9326  0.8060448  653.5496
##   1                  150      1050.3137  0.7960209  671.9234
##   2                   50       976.6771  0.8387113  597.6254
##   2                  100      1002.7364  0.8171829  633.6087
##   2                  150      1027.3280  0.7938211  653.2949
##   3                   50       948.4787  0.8408922  579.1643
##   3                  100      1018.0069  0.7985114  642.3926
##   3                  150      1064.0392  0.7759815  691.8848
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were n.trees = 50, interaction.depth =
##  3, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
SVM2
```

```
## Support Vector Machines with Linear Kernel
##
## 61 samples
##  9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 54, 55, 53, 56, 56, 57, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   292.9234  0.9441664  181.2669
##
## Tuning parameter 'C' was held constant at a value of 1
```

Since it does not make much of a difference, I will keep the original models and make predictions to assess the quality of each model.

###Predicions

```
RFpred <- predict(RF, newdata = testing)
GBMpred <- predict(GBM, newdata = testing)
SVMpred <- predict(SVM, newdata = testing)

RFpred
```

```
##           1           6           8          10          12          13
##   14.849333  377.389467  141.098667   41.124667  327.735067   43.005200
##          15          21          22          26          28          37
## 1444.322933  102.380400 1720.297867 1182.153733   76.665600 1236.556267
##          39          52          54          70          72          77
## 1236.556267  228.214933  363.170533    3.672667   14.146533   90.218133
##          79
## 1415.648400
```

```r
GBMpred
```

```
##  [1]   18.27739  857.93231  453.53939  329.46919  504.21420  330.61336
##  [7] 2495.17512 -218.19980 2632.29852 2474.80278   45.17215 1841.25618
## [13] 1868.93172  747.92614  880.58835  -61.46331  -78.79448 -170.39009
## [19] 1407.42803
```

```r
SVMpred
```

```
##           1           6           8          10          12          13
##   44.758252  126.253577    4.627424  -47.447123   85.847299  -64.863491
##          15          21          22          26          28          37
## 1413.620430   74.382382 1626.330441 1295.395199   55.132218 1260.111734
##          39          52          54          70          72          77
## 1255.761170  142.267706  399.444672 -171.801556  -94.373713   30.623296
##          79
## 1292.024281
```

###Applying Predictions

```r
RFpred2 <- predict(RF, newdata = df22)
```

```r
output <- df2
output$predictions <- RFpred2

head(output)
```

```
##   ProductType   Price x5StarReviews x4StarReviews x3StarReviews x2StarReviews
## 1          PC  699.00            96            26            14            14
## 2          PC  860.00            51            11            10            10
## 3      Laptop 1199.00            74            10             3             3
## 4      Laptop 1199.00             7             2             1             1
## 5      Laptop 1999.00             1             1             1             3
## 6     Netbook  399.99            19             8             4             1
##   x1StarReviews PositiveServiceReview NegativeServiceReview Recommendproduct
## 1            25                    12                     3              0.7
## 2            21                     7                     5              0.6
## 3            11                    11                     5              0.8
## 4             1                     2                     1              0.6
## 5             0                     0                     1              0.3
## 6            10                     2                     4              0.6
##   Volume predictions
## 1      0  445.156133
## 2      0  177.283333
## 3      0  276.714000
## 4      0   30.582800
## 5      0    5.603867
## 6      0   82.282133
```

```r
#write.csv(output, file="C3T3 Predictions.csv", row.names = TRUE)
```