Choose Your Own Project - BlackFriday

Michael Strenge

03 6 2019

1. Introduction

The purpose of this project is to develop a predictive model to forecast the costumer purchase on Black Friday for a particular retail store. The database used for this project is the Black Friday data set provided by Kaggle (https://www.kaggle.com/mehdidag/black-friday (https://www.kaggle.com/mehdidag/black-friday)), which represents a sample of more than 500,000 transactions made in a store. Since the store wants to know better the customer purchase behavior, the regression problem here is to find an algorithm that determines the dependent variable (the amount of purchase) with the help of customer related information such as age, gender, occupation (profession), marital status and city category (place of residence).A short description of the variables can be found on Kaggle as well.

This project constructs a model that identifies the drivers of costumer purchase and compares the performance of a class of machine learning techniques. In machine learning, a common task is the study and construction of algorithms that can learn from and make predictions on data. Such algorithms work by making data-driven predictions through building a model from input data. The model is first fit on a training dataset that is a set of examples used to fit the parameters of the model. In the next step, the fitted model is used to predict the responses for the observations in a second dataset called the validation dataset. This project trains an algorithm using multiple machine learning techniques and the customer related input mentioned above in one subset to predict the amount of purchase in the validation set. The residual mean squared error (RMSE) is used to evaluate how close predictions are to the true values in the validation set.

The report is structured according the performed key project steps and therefore proceeds as follows: In the next section, the methods and analysis procedures will be described by highlighting the data preparation, exploration and visualization techniques and outcomes. Based on these insights, the modelling approach is described and explained. The presentation and discussion of the modelling results (incl. RMSEs) follow. The report concludes with general learnings, outlines the limitations of the applied and tested approach, and provides directions for future modelling endeavors.

2. Methods & Analysis

2.1 Data Preparation

After the corresponding dataset has been loaded, the first step is to get a better understanding of the original database.

```
library(tidyverse)
library(caret)
library(caretEnsemble)
library(plyr)
library(MASS)
library(Hmisc)
library(gclus)
library(glmnet)

# Load data set BlackFriday from my GitHub account
df <- read.csv("https://raw.githubusercontent.com/mistrenge/BlackFriday/master/BlackFriday.csv")

# Explore original dataset
head(df[1:7])
```

	User_ID <int>	Product_ID <fctr>	Gen… <fctr>	Age <fctr>	Occupation <int>	City_Category <fctr>	Stay_In_Current_City_Years <fctr>
1	1000001	P00069042	F	0-17	10	A	2
2	1000001	P00248942	F	0-17	10	A	2
3	1000001	P00087842	F	0-17	10	A	2
4	1000001	P00085442	F	0-17	10	A	2
5	1000002	P00285442	M	55+	16	C	4+
6	1000003	P00193542	M	26-35	15	A	3

6 rows

```
head(df[8:12])
```

	Marital_Status <int>	Product_Category_1 <int>	Product_Category_2 <int>	Product_Category_3 <int>	Purchase <int>
1	0	3	*NA*	*NA*	8370
2	0	1	6	14	15200
3	0	12	*NA*	*NA*	1422
4	0	12	14	*NA*	1057
5	0	8	*NA*	*NA*	7969
6	0	1	2	*NA*	15227

6 rows

```
str(df)
```

```
## 'data.frame':    537577 obs. of  12 variables:
##  $ User_ID                   : int  1000001 1000001 1000001 1000001 1000002 1000003 1000004 100
0004 1000004 1000005 ...
##  $ Product_ID                : Factor w/ 3623 levels "P00000142","P00000242",..: 671 2375 851 8
27 2733 1830 1744 3319 3597 2630 ...
##  $ Gender                    : Factor w/ 2 levels "F","M": 1 1 1 1 2 2 2 2 2 2 ...
##  $ Age                       : Factor w/ 7 levels "0-17","18-25",..: 1 1 1 1 7 3 5 5 5 3 ...
##  $ Occupation                : int  10 10 10 10 16 15 7 7 7 20 ...
##  $ City_Category             : Factor w/ 3 levels "A","B","C": 1 1 1 1 3 1 2 2 2 1 ...
##  $ Stay_In_Current_City_Years: Factor w/ 5 levels "0","1","2","3",..: 3 3 3 3 5 4 3 3 3 2 ...
##  $ Marital_Status            : int  0 0 0 0 0 0 1 1 1 1 ...
##  $ Product_Category_1        : int  3 1 12 12 8 1 1 1 1 8 ...
##  $ Product_Category_2        : int  NA 6 NA 14 NA 2 8 15 16 NA ...
##  $ Product_Category_3        : int  NA 14 NA NA NA NA 17 NA NA NA ...
##  $ Purchase                  : int  8370 15200 1422 1057 7969 15227 19215 15854 15686 7871 ...
```

The original Blackfriday dataset consists of 537,577 observations and contains 12 variables with different formats. Since no detailed description of the author on Kaggle is available for the product-related data, the variables Product_Category 1, 2 and 3 as well as the Product_ID are not taken into account for further analysis. For example, no details are given in terms of the quantity or price per product or detailed information on the three product categories and their relationship to each other. In addition, the further analytic approach intends to form the sum of the purchase per customer because the project wants to examine how much each customer purchases in total and not how much she or he spends on each product. However, the aggregation of the multiple customer data forms a row per customer, making the later analysis of User_ID in the machine learning model irrelevant as well (see section Results).

```
# Select variables for further analysis
df <- df %>%
  dplyr::select(User_ID, Gender, Age, Occupation, City_Category, Stay_In_Current_City_Years, Marit
al_Status, Purchase)

# Transform format of variables to better manipulate data in next step
df <- within(df, {
  Occupation <- as.factor(Occupation)
  Marital_Status <- as.factor(Marital_Status)
})

# Summarize data set to calculate summary purchase for each customer
df <- df %>% group_by(User_ID) %>%
  summarise_each(funs(if(is.numeric(.)) sum(., na.rm = TRUE) else first(.)))
```

As the dataset shows, the variables are either integers or factors. Basically, machine learning techniques can be performed better on the basis of numerical data. Therefore, in the next step, the variables are transformed into a numeric format.

```
# Explore original dataset
# Transform data to numeric variables
revalue(df$Gender, c("F" = 1, "M" = 0)) -> df$Gender
revalue(df$Age, c("0-17" = 0, "18-25" = 1, "26-35" = 2, "36-45"= 3, "46-50" = 4, "51-55" = 5, "55
+" = 6)) -> df$Age
revalue(df$City_Category, c("C" = 0, "B" = 1, "A" = 2)) -> df$City_Category
revalue(df$Stay_In_Current_City_Years, c("0" = 0, "1" = 1, "2" = 2, "3"= 3, "4+" = 4)) -> df$Stay_
In_Current_City_Years

df <- within(df, {
  Gender <- as.numeric(as.character(Gender))
  Age <- as.numeric(as.character(Age))
  City_Category <- as.numeric(as.character(City_Category))
  Stay_In_Current_City_Years <- as.numeric(as.character(Stay_In_Current_City_Years))
  Marital_Status <- as.numeric(as.character(Marital_Status))
  Occupation <- as.numeric(as.character(Occupation))
  Purchase <- as.numeric(as.character(Purchase))
})
```

An excerpt of the dataset after preparation is shown in the following tables.

```
# Check dataset
head(df)
```

User_ID <int>	Gen… <dbl>	… <dbl>	Occupation <dbl>	City_Category <dbl>	Stay_In_Current_City_Years <dbl>	Marital_Status <dbl>
1000001	1	0	10	2	2	0
1000002	0	6	16	0	4	0
1000003	0	2	15	2	3	0
1000004	0	4	7	1	2	1
1000005	0	2	20	2	1	1
1000006	1	5	9	2	1	0

6 rows

```
str(df)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    5891 obs. of  8 variables:
##  $ User_ID                   : int  1000001 1000002 1000003 1000004 1000005 1000006 1000007 100
0008 1000009 1000010 ...
##  $ Gender                    : num  1 0 0 0 0 1 0 0 0 1 ...
##  $ Age                       : num  0 6 2 4 2 5 3 2 2 3 ...
##  $ Occupation                : num  10 16 15 7 20 9 1 12 17 1 ...
##  $ City_Category             : num  2 0 2 1 2 2 1 0 0 1 ...
##  $ Stay_In_Current_City_Years: num  2 4 3 2 1 1 1 4 0 4 ...
##  $ Marital_Status            : num  0 0 0 1 1 0 1 1 0 1 ...
##  $ Purchase                  : num  333481 810353 341635 205987 821001 ...
```

However, before proceeding with the description of the data exploration approach, the missing values are analyzed. Overall, no missing values could be identified in the dataset.

```
# Identify missing values
sum(is.na(df))
```

```
## [1] 0
```

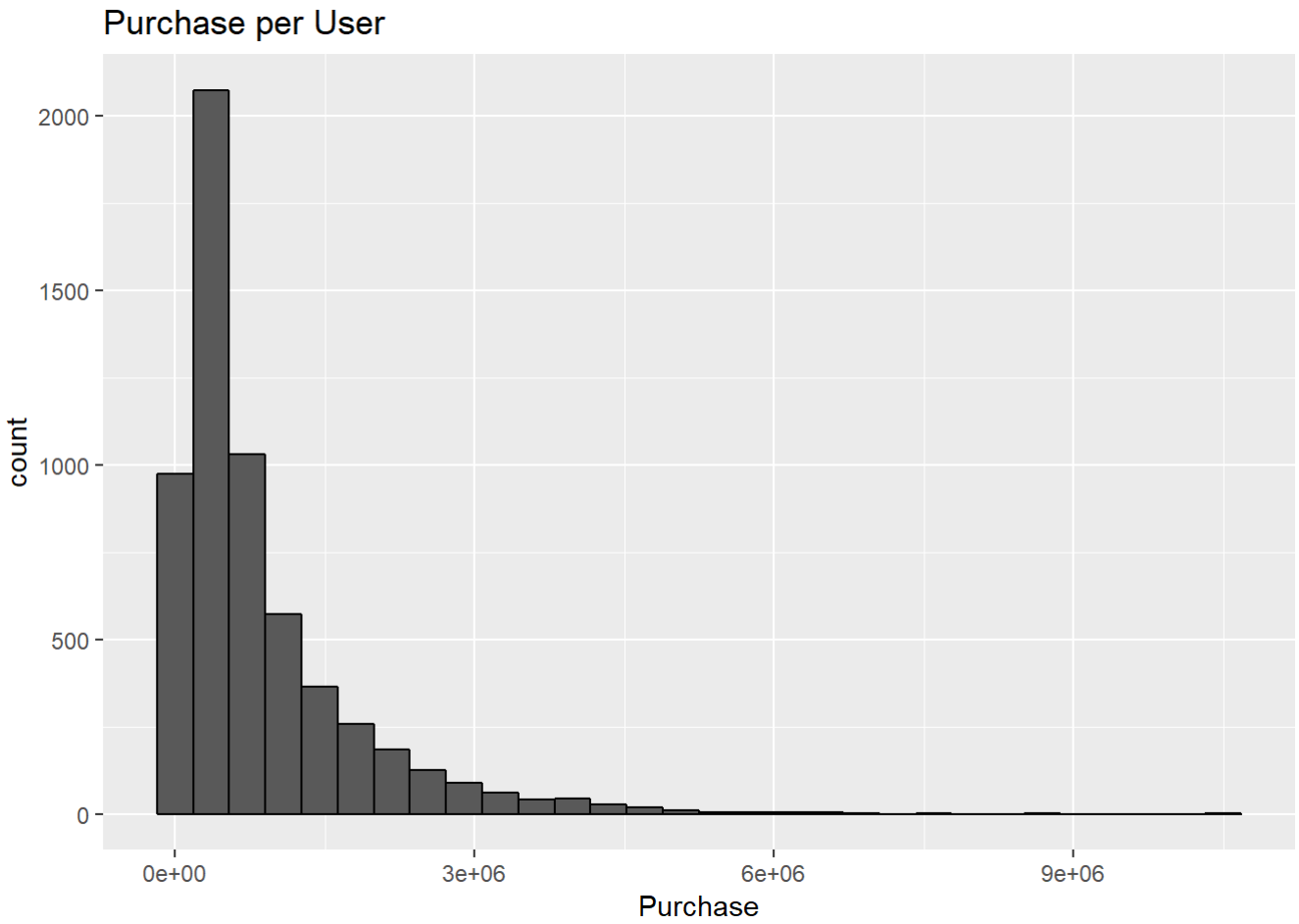
2.2 Data Exploration

The data exploration starts by analyzing the dataset in generel and the dependent variable purchase per costumer in detail, in order to get a better understanding of the distribution of the data.

```
# Explore data set and create histogram for purchase variable
summary(df)
```

```
##     User_ID            Gender            Age         Occupation
##  Min.   :1000001   Min.   :0.0000   Min.   :0.00   Min.   : 0.000
##  1st Qu.:1001518   1st Qu.:0.0000   1st Qu.:2.00   1st Qu.: 3.000
##  Median :1003026   Median :0.0000   Median :2.00   Median : 7.000
##  Mean   :1003025   Mean   :0.2828   Mean   :2.62   Mean   : 8.153
##  3rd Qu.:1004532   3rd Qu.:1.0000   3rd Qu.:3.00   3rd Qu.:14.000
##  Max.   :1006040   Max.   :1.0000   Max.   :6.00   Max.   :20.000
##  City_Category    Stay_In_Current_City_Years Marital_Status
##  Min.   :0.0000   Min.   :0.000              Min.   :0.00
##  1st Qu.:0.0000   1st Qu.:1.000              1st Qu.:0.00
##  Median :0.0000   Median :2.000              Median :0.00
##  Mean   :0.6445   Mean   :1.859              Mean   :0.42
##  3rd Qu.:1.0000   3rd Qu.:3.000              3rd Qu.:1.00
##  Max.   :2.0000   Max.   :4.000              Max.   :1.00
##     Purchase
##  Min.   :   44108
##  1st Qu.:  234914
##  Median :  512612
##  Mean   :  851752
##  3rd Qu.: 1099005
##  Max.   :10536783
```

```
df %>% group_by(User_ID) %>% ggplot(aes(Purchase)) +
  geom_histogram(bins = 30, color = "black") +
  ggtitle ("Purchase per User")
```



The histogram indicates that the distribution is skewed, suggesting to log transform this variable. If the data follows a log normal distribution or approximately so, then the log transformed data follows a normal or near normal distribution. In this case, the log transformation does remove or reduce skewness and simplify further analytic procedures.

```
# Log transform Purchase variable
df <- df %>% mutate(Purchase = log(Purchase))
```

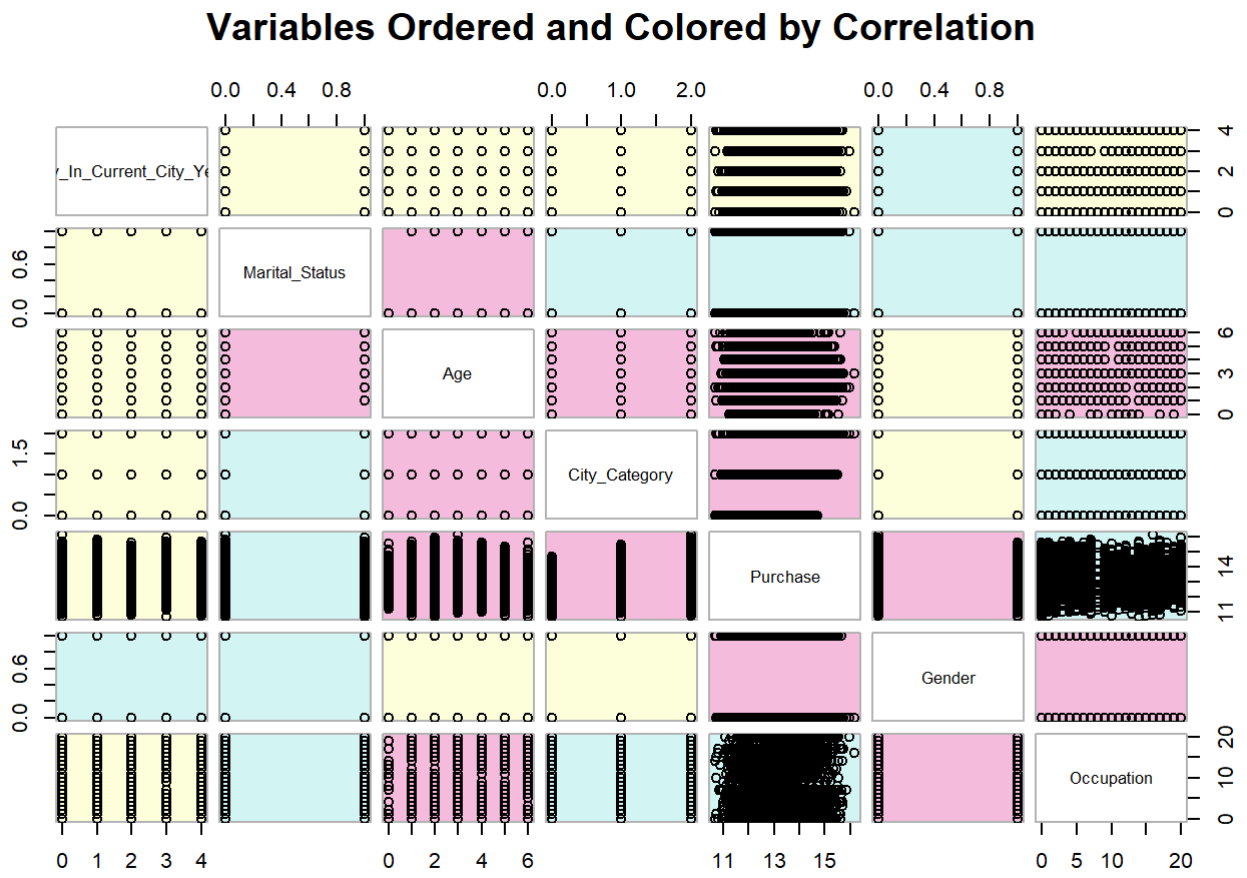
In order to gain initial insights into which independent variables determine the purchasing behavior of customers, the project calculates and illustrates the correlations between the variables using the Hmisc and gclus R packages. The results show that, in particular, there is a connection between the purchasing behavior of the customer and the variables Gender, Age, and City_Category.

```
# Show correlations between variables in training set
cor <-rcorr(as.matrix(df[2:8]))
cor
```

```
##                            Gender   Age Occupation City_Category
## Gender                       1.00  0.01      -0.15          0.01
## Age                          0.01  1.00       0.08         -0.10
## Occupation                  -0.15  0.08       1.00         -0.04
## City_Category                0.01 -0.10      -0.04          1.00
## Stay_In_Current_City_Years  -0.01 -0.01       0.01          0.00
## Marital_Status               0.01  0.33       0.03         -0.05
## Purchase                    -0.12 -0.08       0.02          0.30
##                            Stay_In_Current_City_Years Marital_Status
## Gender                                          -0.01           0.01
## Age                                             -0.01           0.33
## Occupation                                       0.01           0.03
## City_Category                                    0.00          -0.05
## Stay_In_Current_City_Years                       1.00          -0.01
## Marital_Status                                  -0.01           1.00
## Purchase                                         0.01          -0.02
##                            Purchase
## Gender                        -0.12
## Age                           -0.08
## Occupation                     0.02
## City_Category                  0.30
## Stay_In_Current_City_Years     0.01
## Marital_Status                -0.02
## Purchase                       1.00
##
## n= 5891
##
##
## P
##                            Gender Age    Occupation City_Category
## Gender                            0.4140 0.0000     0.4675
## Age                        0.4140        0.0000     0.0000
## Occupation                 0.0000 0.0000            0.0016
## City_Category              0.4675 0.0000 0.0016
## Stay_In_Current_City_Years 0.3382 0.5499 0.4926     0.9565
## Marital_Status             0.2570 0.0000 0.0140     0.0003
## Purchase                   0.0000 0.0000 0.2225     0.0000
##                            Stay_In_Current_City_Years Marital_Status
## Gender                     0.3382                      0.2570
## Age                        0.5499                      0.0000
## Occupation                 0.4926                      0.0140
## City_Category              0.9565                      0.0003
## Stay_In_Current_City_Years                             0.4080
## Marital_Status             0.4080
## Purchase                   0.5288                      0.0730
##                            Purchase
## Gender                     0.0000
## Age                        0.0000
## Occupation                 0.2225
## City_Category              0.0000
## Stay_In_Current_City_Years 0.5288
## Marital_Status             0.0730
## Purchase
```

```
# Create scatter plot matrix to show correlations between variables in training set
dta <- df[c(2:8)] # get data
dta.r <- abs(cor(dta)) # get correlations
dta.col <- dmat.color(dta.r) # get colors

# Reorder variables so those with highest correlation are closest to the diagonal
dta.o <- order.single(dta.r)
cpairs(dta, dta.o, panel.colors=dta.col, gap=.5,
  main="Variables Ordered and Colored by Correlation")
```



2.3 Modeling Approach

As already mentioned, the project uses the RMSE to evaluate how close predictions are to the true values in the validation set. The projects defines y as the observed purchase per costumer and denote the prediction with $\hat{y}$. Thus, the RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum(\hat{y} - y)^2}$$

with N being the sample size. The RMSE can be interpret similarly to a standard deviation.

The project uses a two-step procedure to construct the algorithm and compare the performance of the different techniques based on the RMSE results.

Step 1: Analysis of the entire set of independent variables

The first model trains the algorithm on the training set and validates the results on the test set, using all independent variables (Gender, Age, Occupation, City_Category, Stay_In_Current_City_Years, Marital_Status). Please note that the calculation is only based on seven different common used machine learning techniques for regression models

because of computational power issues. Furthermore, two ensemble models are calculated based on linear combination and stack method (glmnet). The modeling approach uses the R packages caret, caretEnsemble and glmnet. The glmnet package provides extremely efficient procedures for fitting the entire lasso or elastic-net regularization path for linear regression, logistic and multinomial regression models, Poisson regression and the Cox model. The projects controls also for cross-validation and correlation of the ensemble models and resamples the model results. Resampling is a series of methods used to reconstruct the sample data sets, including training and validation sets. It can provide more “useful” different sample sets for learning process in some way.

Step 2: Analysis of a selected set of independent variables

Before proceeding with the model approach, the project uses a step-wise-regression to reduce the independent variables. Stepwise regression is a modification of the forward selection so that after each step in which a variable was added, all candidate variables in the model are checked to see if their significance has been reduced below the specified tolerance level. If a nonsignificant variable is found, it is removed from the model. Nisbet et al. (2018) suggest that this procedure is a common practice for variable selection, before training a final model with a machine learning algorithm. Based on these results, the second model performs the same steps as the first one, but including only the independent variables that seem to have a significant influence on the dependent variable Purchase.

3. Results

3.1 Step 1: Analysis of the entire set of independent variables

As described in the previous section, the project proceeds stepwise to construct the models and compares the different approaches based on the RMSE results. First, the data base of the record is plotted in a training and test set, whereby the usual 30% to 70% ratio is used.

```
# Select variables for machine learning model
df <- df %>%
  dplyr::select(Gender, Age, Occupation, City_Category, Stay_In_Current_City_Years, Marital_Statu
s, Purchase)

# Split training and test set
set.seed(1)
test_index <- createDataPartition(y = df$Purchase, times = 1, p = 0.7, list = FALSE)
train_set <- df[-test_index,]
test_set <- df[test_index,]
```

The next step builds the machine learning algorithm for the training data with all independent variables.

```
# 1. Model: build machine learning model for training data with all independent variables
set.seed(222)
control <- trainControl(method = "cv", number = 5, savePredictions = "final",
  allowParallel = TRUE) # Check for cross-validation
fits <- caretList(Purchase ~ ., trControl = control, methodList =
  c("lm", "rpart", "rf", "glm", "gbm", "knn", "svmLinear"),
  data = train_set, tuneList = NULL, continue_on_fail = FALSE)
```

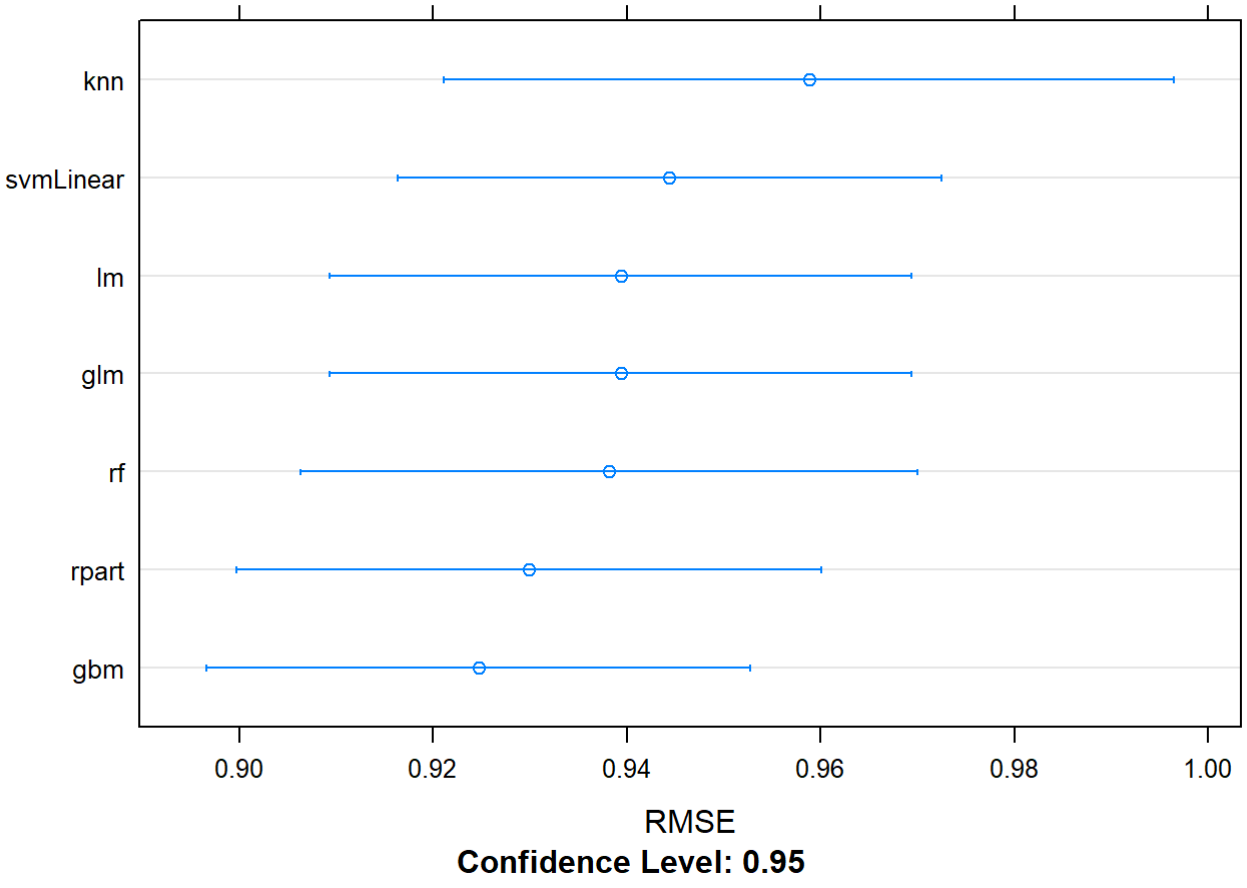
The table below reports the training results of the first modeling approach and the RMSE for each model. Basically, the models seem to perform very similarly, with the Gradient Boosting Machine model (“gbm”) having the best RMSE of 0.92.

```
# Print RMSE of each model for training data
model_results <- data.frame(
  lm = min(fits$lm$results$RMSE),
  rpart = min(fits$rpart$results$RMSE),
  rf = min(fits$rf$results$RMSE),
  glm = min(fits$glm$results$RMSE),
  gbm = min(fits$gbm$results$RMSE),
  knn = min(fits$knn$results$RMSE),
  svmLinear = min(fits$svmLinear$results$RMSE))
print(model_results)
```

```
##          lm    rpart        rf       glm       gbm       knn svmLinear
## 1 0.9393918 0.929926 0.9381937 0.9393918 0.9247223 0.9587952 0.9443941
```

The resampling procedure confirms the performance of the multiple models.

```
# Resample performance of models
resamples <- resamples(fits)
dotplot(resamples, metric = "RMSE")
```



Heareafter, the project calculates the first ensemble model based on the linear combination of multiple models.The ensemble model with a RMSE of 0.92 achieves a comparable value as the Gradient Boosting Machine model.

```
# Create ensemble of models by performing a linear combination
ensemble_a <- caretEnsemble(fits, metric = "RMSE", trControl = control)
summary(ensemble_a)
```

```
## The following models were ensembled: lm, rpart, rf, glm, gbm, knn, svmLinear
## They were weighted:
## -1.3475 0.2444 0.1716 0.1034 NA 0.7477 0.1667 -0.3305
## The resulting RMSE is: 0.9239
## The fit for each individual model on the RMSE is:
##     method      RMSE     RMSESD
##         lm 0.9393918 0.02417250
##      rpart 0.9299260 0.02428412
##         rf 0.9381937 0.02563000
##        glm 0.9393918 0.02417250
##        gbm 0.9247223 0.02260147
##        knn 0.9587952 0.03031451
##  svmLinear 0.9443941 0.02260527
```

In order to gain insights that are useful for the formation of the second ensemble model (stack approach), the correlation between the individual models is calculated. The correlation between the models generally seems to be very high, so the project propose to use the glmnet method. Also the stack model cannot achieve a significant improvement of the RMSE.

```
# Check correlation of models
modelCor(resamples)
```

```
##                  lm     rpart        rf       glm       gbm       knn
## lm        1.0000000 0.9568872 0.9471142 1.0000000 0.9418703 0.7232499
## rpart     0.9568872 1.0000000 0.9373078 0.9568872 0.8836960 0.6745870
## rf        0.9471142 0.9373078 1.0000000 0.9471142 0.9851296 0.8596030
## glm       1.0000000 0.9568872 0.9471142 1.0000000 0.9418703 0.7232499
## gbm       0.9418703 0.8836960 0.9851296 0.9418703 1.0000000 0.8925359
## knn       0.7232499 0.6745870 0.8596030 0.7232499 0.8925359 1.0000000
## svmLinear 0.9868946 0.9758533 0.9762295 0.9868946 0.9576618 0.7891803
##           svmLinear
## lm        0.9868946
## rpart     0.9758533
## rf        0.9762295
## glm       0.9868946
## gbm       0.9576618
## knn       0.7891803
## svmLinear 1.0000000
```

```
# Create ensemble of models by using glmnet ("meta model")
ensemble_b <- caretStack(fits,
                         method = "glmnet",
                         metric = "RMSE",
                         trControl = control)
print(ensemble_b)
```

```
## A glmnet ensemble of 2 base models: lm, rpart, rf, glm, gbm, knn, svmLinear
##
## Ensemble results:
## glmnet
##
## 1764 samples
##    7 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1411, 1411, 1411, 1412, 1411
## Resampling results across tuning parameters:
##
##   alpha  lambda        RMSE       Rsquared   MAE
##   0.10   0.0007421935  0.9260280  0.1397691  0.7769194
##   0.10   0.0074219350  0.9257827  0.1403493  0.7770289
##   0.10   0.0742193504  0.9253518  0.1417360  0.7774475
##   0.55   0.0007421935  0.9260191  0.1397925  0.7770253
##   0.55   0.0074219350  0.9257825  0.1403931  0.7773778
##   0.55   0.0742193504  0.9264549  0.1408960  0.7786304
##   1.00   0.0007421935  0.9260280  0.1397961  0.7771186
##   1.00   0.0074219350  0.9257454  0.1403811  0.7776016
##   1.00   0.0742193504  0.9288568  0.1390152  0.7810786
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0.1 and lambda
##  = 0.07421935.
```

The following table illustrates the results of the tested algorithm based on the validation set. The results show a very similar picture here, the RMSE values of the individual models are very close to each other. Both ensemble models and the Gradient Boosting Machine model provide the best performance.

```
# Validate model on test data
# Predict on test data
pred_lm <- predict.train(fits$lm, newdata = test_set)
pred_rpart <- predict.train(fits$rpart, newdata = test_set)
pred_rf <- predict.train(fits$rf, newdata = test_set)
pred_glm <- predict.train(fits$glm, newdata = test_set)
pred_gbm <- predict.train(fits$gbm, newdata = test_set)
pred_knn <- predict.train(fits$knn, newdata = test_set)
pred_svmLinear <- predict.train(fits$svmLinear, newdata = test_set)
predict_ensa <- predict(ensemble_a, newdata = test_set)
predict_ensb <- predict(ensemble_b, newdata = test_set)

# Get RMSE for test data
pred_RMSE <- data.frame(ensemble_a = RMSE(predict_ensa, test_set$Purchase),
                        ensemble_b = RMSE(predict_ensb, test_set$Purchase),
                        lm = RMSE(pred_lm, test_set$Purchase),
                        rf = RMSE(pred_rf, test_set$Purchase),
                        glm = RMSE(pred_glm, test_set$Purchase),
                        gbm = RMSE(pred_gbm, test_set$Purchase),
                        knn = RMSE(pred_knn, test_set$Purchase),
                        svmLinear = RMSE(pred_svmLinear, test_set$Purchase))
print(pred_RMSE)
```

```
##   ensemble_a ensemble_b        lm        rf       glm       gbm       knn
## 1  0.9206136  0.9217916 0.9409039 0.9365805 0.9409039 0.9207154 0.9715504
##   svmLinear
## 1 0.9449045
```

3.2 Step 2: Analysis of a selected set of independent variables

As described in the previous section, a stepwise regression is first calculated to select the independent variables that have a significant impact on customer purchasing behavior. The results are generally in line with the correlation analysis and suggest that the model should be limited to the independent variables Age, Gender and City_Category.

```
# Perform stepwise regression to reduce independent variables
lm_fit <- lm(Purchase ~ ., data = train_set)
step <- stepAIC(lm_fit, direction = "both")
```

```
# Print results of stepwise regression
step$anova
```

Step <fctr>	Df <dbl>	Deviance <dbl>	Resid. Df <dbl>	Resid. Dev <dbl>	AIC <dbl>
	NA	*NA*	1757	1545.900	-218.8091
- Marital_Status	1	0.05476194	1758	1545.954	-220.7466
- Stay_In_Current_City_Years	1	0.55374360	1759	1546.508	-222.1149
- Occupation	1	1.35590043	1760	1547.864	-222.5690

4 rows

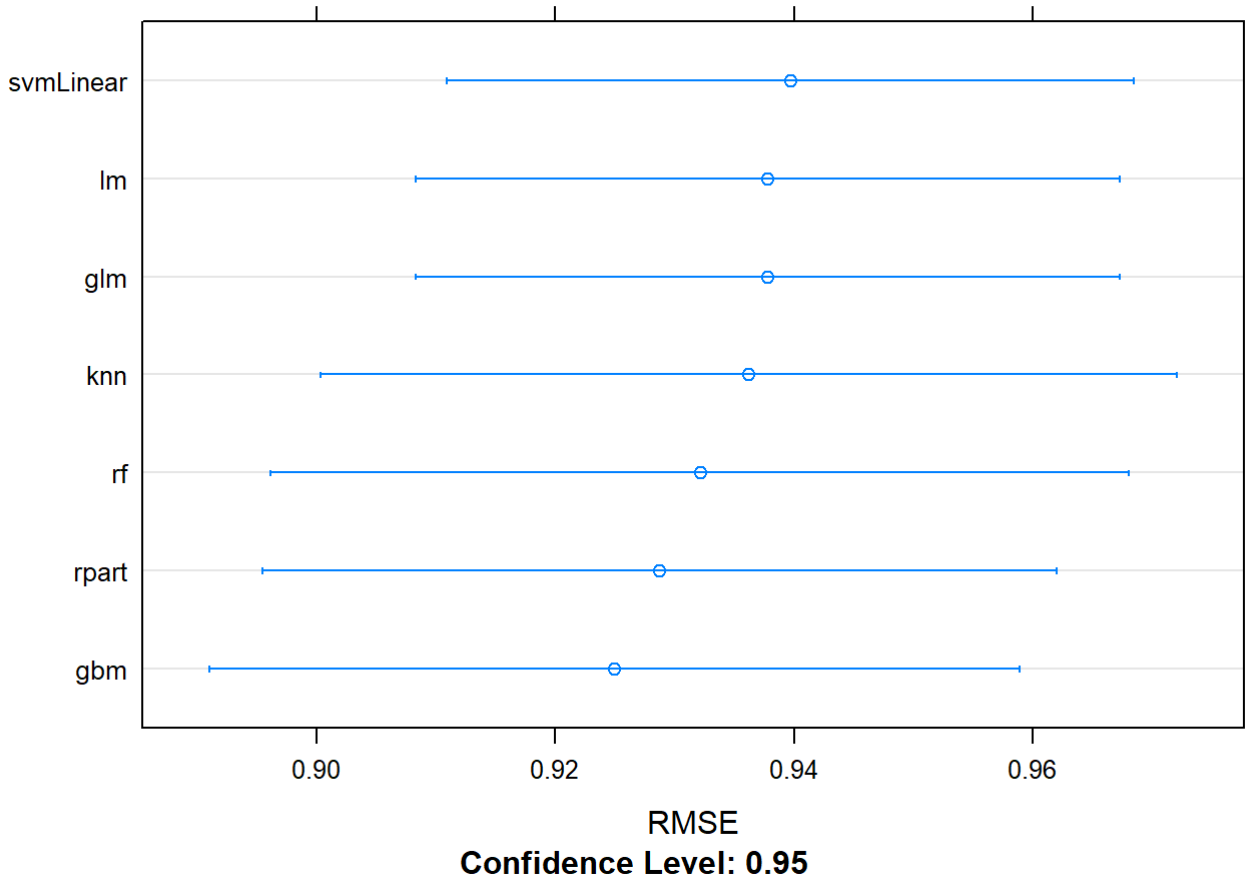
The second step performs the same procedure as the first one, but including only the three independent variables Age, Gender and City_Status. However, the performed analyzes show very similar results as in the model with all independent variables. The two ensemble models and the Gradient Boosting Machine model achieve the best performance with a RMSE of 0.92.

```
# 2. Step: Analysis of a selected set of independent variables
# Build machine learning model for training data with reduced independent variables
set.seed(222)
control <- trainControl(method = "cv", number = 5, savePredictions = "final",
  allowParallel = TRUE) # Check for cross-validation
fits_2 <- caretList(Purchase ~ Gender + Age + City_Category, trControl = control, methodList =
  c("lm", "rpart", "rf", "glm", "gbm", "knn", "svmLinear"),
  data = train_set, tuneList = NULL, continue_on_fail = FALSE)
```

```
# Print RMSE of each model for training data using reduced independent variables
model_results_2 <- data.frame(
  lm = min(fits_2$lm$results$RMSE),
  rpart = min(fits_2$rpart$results$RMSE),
  rf = min(fits_2$rf$results$RMSE),
  glm = min(fits_2$glm$results$RMSE),
  gbm = min(fits_2$gbm$results$RMSE),
  knn = min(fits_2$knn$results$RMSE),
  svmLinear = min(fits_2$svmLinear$results$RMSE))
print(model_results_2)
```

```
##          lm     rpart        rf       glm       gbm       knn svmLinear
## 1 0.9378017 0.9287382 0.9321085 0.9378017 0.9249671 0.9362016 0.9396968
```

```
# Resample performance of new models
resamples_2 <- resamples(fits_2)
dotplot(resamples_2, metric = "RMSE")
```




```
# Create a new ensemble of models by performing a linear combination
ensemble_2a <- caretEnsemble(fits_2, metric = "RMSE", trControl = control)
summary(ensemble_2a)
```

```
## The following models were ensembled: lm, rpart, rf, glm, gbm, knn, svmLinear
## They were weighted:
## -0.4981 -0.4085 0.2314 -0.2555 NA 0.9235 0.2004 0.3456
## The resulting RMSE is: 0.9285
## The fit for each individual model on the RMSE is:
##     method      RMSE     RMSESD
##         lm 0.9378017 0.02375301
##      rpart 0.9287382 0.02677508
##         rf 0.9321085 0.02895966
##        glm 0.9378017 0.02375301
##        gbm 0.9249671 0.02729871
##        knn 0.9362016 0.02886134
##  svmLinear 0.9396968 0.02314873
```

```
# Create ensemble of models by using glmnet ("meta model")
ensemble_2b <- caretStack(fits_2,
                          method = "glmnet",
                          metric = "RMSE",
                          trControl = control)
print(ensemble_2b)
```

```
## A glmnet ensemble of 2 base models: lm, rpart, rf, glm, gbm, knn, svmLinear
##
## Ensemble results:
## glmnet
##
## 1764 samples
##    7 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1409, 1412, 1411, 1412, 1412
## Resampling results across tuning parameters:
##
##   alpha  lambda        RMSE       Rsquared   MAE
##   0.10   0.0007399113  0.9284315  0.1367582  0.7782635
##   0.10   0.0073991135  0.9277389  0.1378490  0.7779077
##   0.10   0.0739911347  0.9279063  0.1374011  0.7786068
##   0.55   0.0007399113  0.9280738  0.1374699  0.7781063
##   0.55   0.0073991135  0.9273452  0.1384679  0.7777287
##   0.55   0.0739911347  0.9285434  0.1378876  0.7794691
##   1.00   0.0007399113  0.9278791  0.1377885  0.7779836
##   1.00   0.0073991135  0.9272241  0.1387202  0.7777264
##   1.00   0.0739911347  0.9295004  0.1397280  0.7807148
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 1 and lambda
##  = 0.007399113.
```

```
# Validate reduced model on test data
# Predict on test data
pred_lm2 <- predict.train(fits_2$lm, newdata = test_set)
pred_rpart2 <- predict.train(fits_2$rpart, newdata = test_set)
pred_rf2 <- predict.train(fits_2$rf, newdata = test_set)
pred_glm2 <- predict.train(fits_2$glm, newdata = test_set)
pred_gbm2 <- predict.train(fits_2$gbm, newdata = test_set)
pred_knn2 <- predict.train(fits_2$knn, newdata = test_set)
pred_svmLinear2 <- predict.train(fits_2$svmLinear, newdata = test_set)
predict_ens2a <- predict(ensemble_2a, newdata = test_set)
predict_ens2b <- predict(ensemble_2b, newdata = test_set)

# Get RMSE for test data
pred_RMSE_2 <- data.frame(ensemble_2a = RMSE(predict_ens2a, test_set$Purchase),
                          ensemble_2b = RMSE(predict_ens2b, test_set$Purchase),
                          lm_2 = RMSE(pred_lm2, test_set$Purchase),
                          rf_2 = RMSE(pred_rf2, test_set$Purchase),
                          glm_2 = RMSE(pred_glm2, test_set$Purchase),
                          gbm_2 = RMSE(pred_gbm2, test_set$Purchase),
                          knn_2 = RMSE(pred_knn2, test_set$Purchase),
                          svmLinear_2 = RMSE(pred_svmLinear2, test_set$Purchase))
print(pred_RMSE_2)
```

```
##   ensemble_2a ensemble_2b      lm_2      rf_2     glm_2     gbm_2
## 1   0.9185536   0.9187086 0.9400566 0.9230815 0.9400566 0.9191928
##       knn_2 svmLinear_2
## 1 0.9272265   0.9426946
```

4. Conclusion

The objective of this project is to develop a predictive model to forecast the costumer purchase on Black Friday for a particular retail store. A model is constructed that identifies the drivers of costumer purchase and compares the performance of a class of machine learning techniques. For this purpose, a two-step approach is chosen and first a model with all and then a model with selected variables is calculated. In particular, the results suggest that the variables Age, Gender and City_Category determine the purchasing behavior of customers in a store. Furthermore, the project shows that the machine learning techniques are relatively close to each other in their performance, but the best RMSE values are provided by the ensemble methods and the Gradient Boosting Machine method, both in the approach with all variables and in the one with the selected variables.

This project has some limitations that point to interesting avenues for future research and modelling endeavors. The project does not consider an analysis of product-related variables for the reasons already described. One possibility, for example, would be to deepen the analysis at the product level. If more information is available on the product categories, this may provide further insights into customer purchasing behavior. Although several machine learning techniques are used, no method can really stand out significantly. From a methodical perspective, future modeling efforts can leverage other machine learning techniques, compute other ensemble models, or even use more complex approaches from the deep learning repertoire such as neural networks.

The analyzes carried out are certainly only a starting point for really understanding customer purchasing behavior. However, the results of this project will hopefully help retailers to better tailor their products and services to their customer needs, taking into account certain characteristics such as age, gender or place of residence.

References

Robert Nisbet, Gary Miner and Ken Yale (2018): Handbook of Statistical Analysis and Data Mining Applications

https://github.com/mistrenge/BlackFriday.git (https://github.com/mistrenge/BlackFriday.git)