R Project - Regression

4375 Machine Learning with Dr. Mazidi

Anthony Martinez | netid: amm180005

10/17/21

Project: Predict how much a person will spend on Black Friday using data set from Kaggle.

The data set was downloaded from Kaggle: https://www.kaggle.com/sdolezel/black-friday?select=train.csv (https://www.kaggle.com/sdolezel/black-friday?select=train.csv)

Load the data

```
df1<-read.csv("data/blackfriday.csv", header=TRUE)
```

Data Cleaning

- Link to data: https://www.kaggle.com/sdolezel/black-friday?select=train.csv (https://www.kaggle.com/sdolezel/black-friday?select=train.csv)
- describe what steps you had to do for data cleaning (more points for messier data that needed cleaning)

I Performed the following steps. 1) count NAs in all columns 2) Deleted unneeded (All NAs ended up being in columns that were not needed). 3) Made all qualitative data into factors

```
sapply(df1, function(x) sum(is.na(x)))
```

```
##                    User_ID                 Product_ID
##                          0                          0
##                     Gender                        Age
##                          0                          0
##                 Occupation              City_Category
##                          0                          0
## Stay_In_Current_City_Years             Marital_Status
##                          0                          0
##         Product_Category_1         Product_Category_2
##                          0                     173638
##         Product_Category_3                   Purchase
##                     383247                          0
```

```
# Here we see that product category2 and category 3 have a lot of NAs
# This is because category 2 and 3 are the "other" categories that a product could
# be included in. We will not be using these features as a predictor so I will
# remove these columns from the data frame

df <- df1[-c(7,10,11)]

# output data frame with removed columns
sapply(df, function(x) sum(is.na(x)))
```

```
##            User_ID         Product_ID             Gender                Age
##                  0                  0                  0                  0
##         Occupation      City_Category     Marital_Status Product_Category_1
##                  0                  0                  0                  0
##           Purchase
##                  0
```

```
# make save factor columns, as factors
df$Gender <- factor(df$Gender, levels=c("M", "F"))
contrasts(df$Gender)
```

```
##   F
## M 0
## F 1
```

```
# from the unique function it can be seen that there are 7 categories for age
unique(df$Age)
```

```
## [1] "0-17"  "55+"   "26-35" "46-50" "51-55" "36-45" "18-25"
```

```
# make age factor since the data does not report the person's age but rather their age range
df$Age <- factor(df$Age, levels=c("0-17", "55+","26-35","46-50","51-55","36-45","18-25"))
#contrasts(df$Age)

unique(df$Occupation)
```

```
##  [1] 10 16 15  7 20  9  1 12 17  0  3  4 11  8 19  2 18  5 14 13  6
```

```
df$Occupation <- factor(df$Occupation,levels=c(10,16,15,7,20,9,1,12,17,0,3,4,11,8,19,2, 18,5,14,
13,6))

#contrasts(df$Occupation)


unique(df$City_Category)
```

```
## [1] "A" "C" "B"
```

```
df$City_Category <- factor(df$City_Category, levels=c("A","B","C"))
#contrasts(df$City_Category)

# marital status
df$Marital_Status <- factor(df$Marital_Status, levels=c("1","0"))
#contrasts(df$Marital_Status)

# prod category
unique(df$Product_Category_1)
```

```
##  [1]  3  1 12  8  5  4  2  6 14 11 13 15  7 16 18 10 17  9 20 19
```

```
df$Product_Category_1 <- factor(df$Product_Category_1, levels=c(3,1, 12,8,5,4,2,6,14,11,13,15,7,
16,18,10,17,9,20,19))
#contrasts(df$Product_Category_1)
```

Step 2 Data Exploreation

- use at least 5 R functions for data exploration
- create at least 2 informative R graphs for data exploration

```
summary(df)
```

```
##     User_ID          Product_ID         Gender        Age
##  Min.   :1000001   Length:550068      M:414259   0-17 : 15102
##  1st Qu.:1001516   Class :character   F:135809   55+  : 21504
##  Median :1003077   Mode  :character              26-35:219587
##  Mean   :1003029                                 46-50: 45701
##  3rd Qu.:1004478                                 51-55: 38501
##  Max.   :1006040                                 36-45:110013
##                                                  18-25: 99660
##    Occupation     City_Category Marital_Status Product_Category_1
##  4      : 72308   A:147720      1:225337       5      :150933
##  0      : 69638   B:231173      0:324731       1      :140378
##  7      : 59133   C:171175                     8      :113925
##  1      : 47426                                11     : 24287
##  17     : 40043                                2      : 23864
##  20     : 33562                                6      : 20466
##  (Other):227958                                (Other): 76215
##     Purchase
##  Min.   :   12
##  1st Qu.: 5823
##  Median : 8047
##  Mean   : 9264
##  3rd Qu.:12054
##  Max.   :23961
##
```

```
head(df)
```

	User_ID <int>	Product_ID <chr>	Gen… <fct>	Age <fct>	Occupation <fct>	City_Category <fct>	Marital_Status <fct>	Product_Categ <fct>
1	1000001	P00069042	F	0-17	10	A	0	3
2	1000001	P00248942	F	0-17	10	A	0	1
3	1000001	P00087842	F	0-17	10	A	0	12
4	1000001	P00085442	F	0-17	10	A	0	12
5	1000002	P00285442	M	55+	16	C	0	8
6	1000003	P00193542	M	26-35	15	A	0	1

6 rows | 1-9 of 10 columns

```
colnames(df)
```

```
## [1] "User_ID"            "Product_ID"         "Gender"
## [4] "Age"                "Occupation"         "City_Category"
## [7] "Marital_Status"     "Product_Category_1" "Purchase"
```

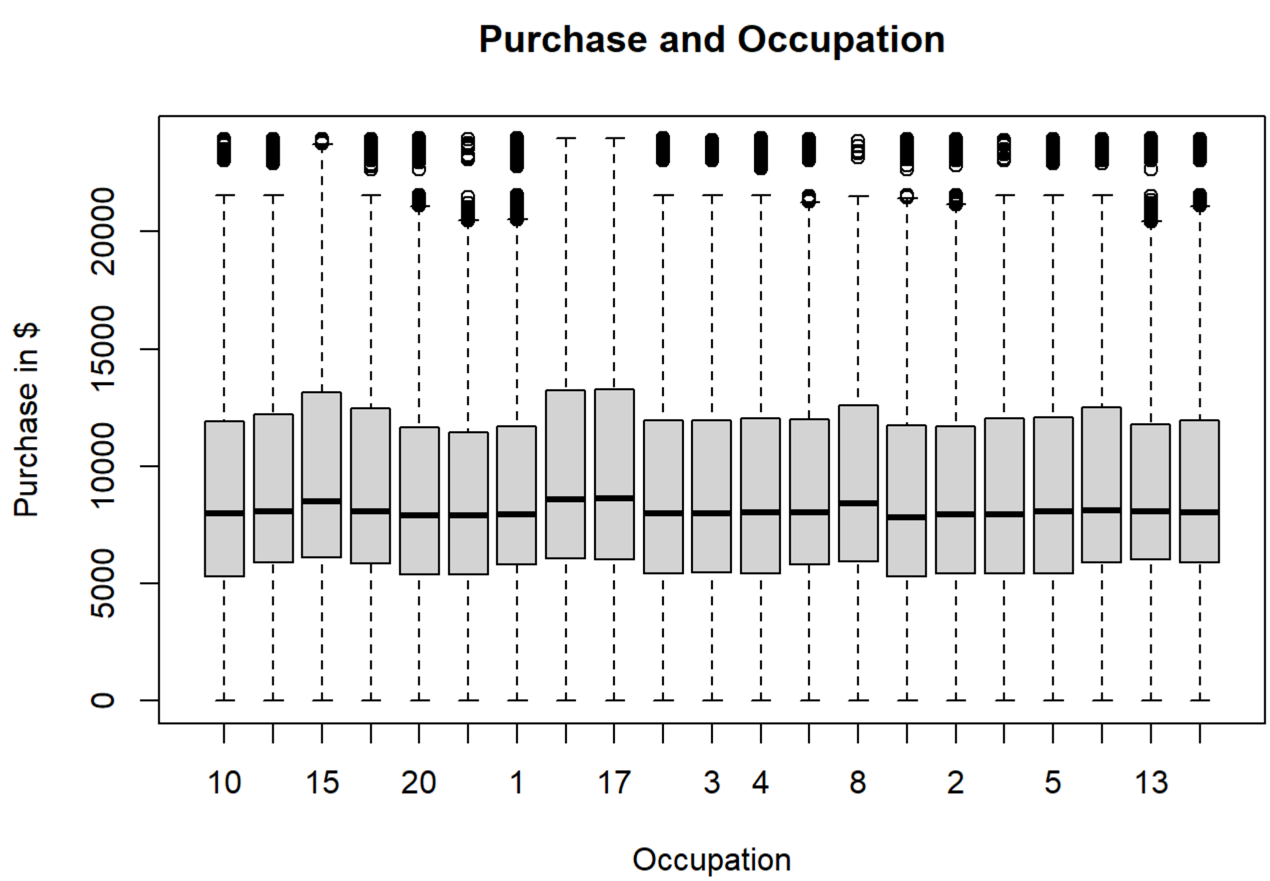
```
str(df)
```

```
## 'data.frame':    550068 obs. of  9 variables:
##  $ User_ID           : int  1000001 1000001 1000001 1000001 1000002 1000003 1000004 1000004 1
000004 1000005 ...
##  $ Product_ID        : chr  "P00069042" "P00248942" "P00087842" "P00085442" ...
##  $ Gender            : Factor w/ 2 levels "M","F": 2 2 2 2 1 1 1 1 1 1 ...
##  $ Age               : Factor w/ 7 levels "0-17","55+","26-35",..: 1 1 1 1 2 3 4 4 4 3 ...
##  $ Occupation        : Factor w/ 21 levels "10","16","15",..: 1 1 1 1 2 3 4 4 4 5 ...
##  $ City_Category     : Factor w/ 3 levels "A","B","C": 1 1 1 1 3 1 2 2 2 1 ...
##  $ Marital_Status    : Factor w/ 2 levels "1","0": 2 2 2 2 2 2 1 1 1 1 ...
##  $ Product_Category_1: Factor w/ 20 levels "3","1","12","8",..: 1 2 3 3 4 2 2 2 2 4 ...
##  $ Purchase          : int  8370 15200 1422 1057 7969 15227 19215 15854 15686 7871 ...
```

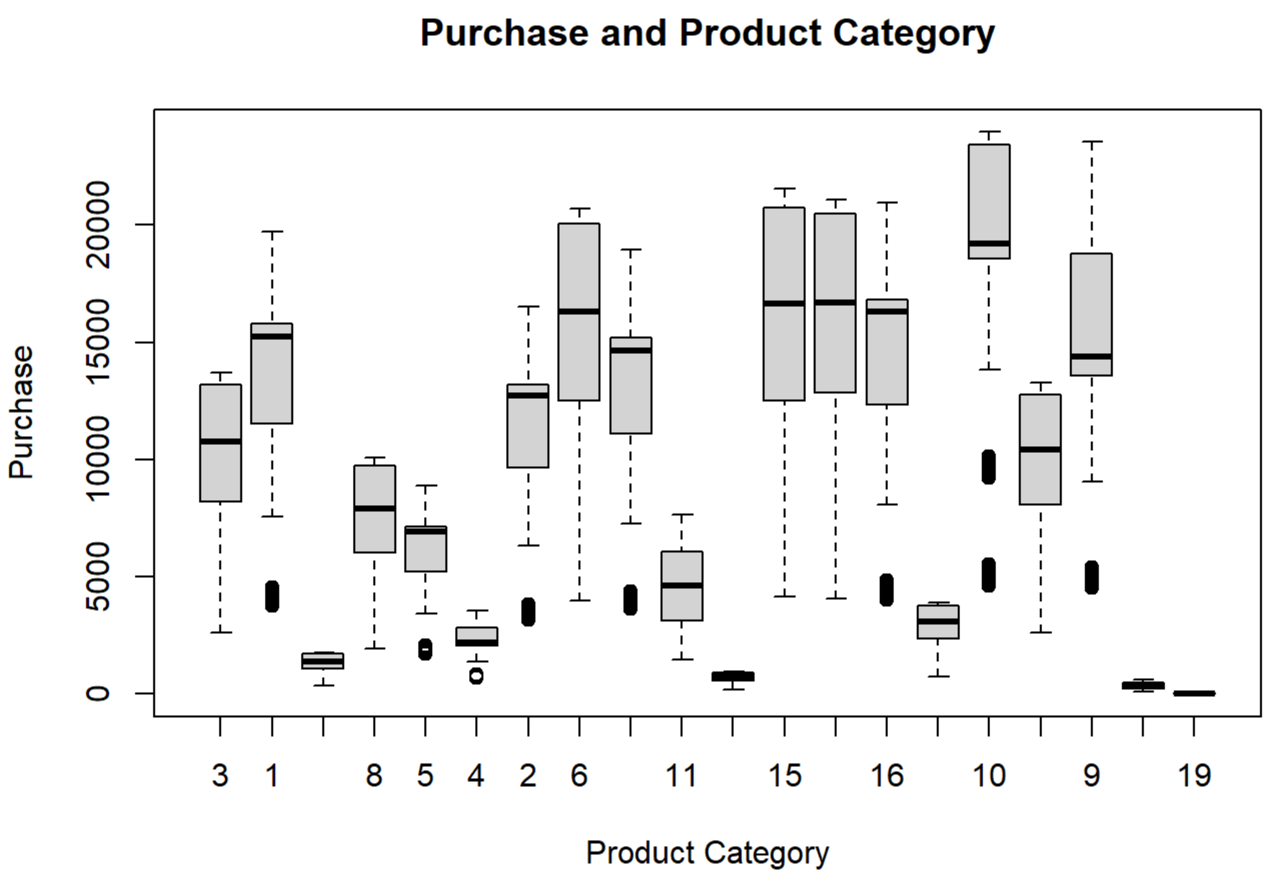
```
# Our target variable is Purchase, which is how much money the observation (a person) spent on b
lack Friday
mean(df$Purchase)
```

```
## [1] 9263.969
```

```
# Graphs
plot(df$Purchase~df$Occupation, xlab="Occupation", ylab="Purchase in $", main="Purchase and Occu
pation")
```



```
plot(df$Purchase~df$Product_Category_1, xlab="Product Category", ylab="Purchase", main="Purchase and Product Category")
```



Divide train/test

- Divide into 75/25 train/test, using seed 1234

```
set.seed(1234)
i <- sample(1:nrow(df), .75*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

Algorithm 1: Linear Regression (Multiple LR)

- code to run the algorithms
- commentary on feature selection you selected and why
- code to compute your metrics for evaluation as well as commentary discussing the results

Commentary on Features Chosen: I decided to use Age, Occupation, and City Category predictors because they seem to be a good indication of the amount of disposable income a person has available. The more disposable income they have, the more money they are able to spend on black Friday. I decided to use Product category as a predictor because the price of items is highly associated with its category. For example, tech/electronics will cost more than clothing.

Commentary on Results: The linear regression model did a pretty good job at predicting the target. The accuracy of 79% and mse of 9,129,462 which will be useful when comparing models. The R^2 value is .6 which is decent

```
lm1 <- lm(Purchase~Age+Occupation+City_Category+Product_Category_1, data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = Purchase ~ Age + Occupation + City_Category + Product_Category_1,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15556.3  -1598.7    398.9   1961.9   8334.0
##
## Coefficients:
##                        Estimate Std. Error  t value Pr(>|t|)
## (Intercept)           9.781e+03  3.993e+01  244.948  < 2e-16 ***
## Age55+                1.428e+02  5.284e+01    2.703 0.006865 **
## Age26-35             -1.278e+02  4.690e+01   -2.725 0.006424 **
## Age46-50             -1.438e+00  4.929e+01   -0.029 0.976725
## Age51-55              2.580e+02  4.984e+01    5.177 2.26e-07 ***
## Age36-45             -2.047e-01  4.757e+01   -0.004 0.996567
## Age18-25             -2.294e+02  4.719e+01   -4.861 1.17e-06 ***
## Occupation16          1.666e+02  5.468e+01    3.046 0.002319 **
## Occupation15          4.645e+02  5.911e+01    7.857 3.94e-15 ***
## Occupation7           1.808e+02  5.205e+01    3.473 0.000515 ***
## Occupation20         -3.114e+01  5.349e+01   -0.582 0.560419
## Occupation9           2.142e+02  6.670e+01    3.212 0.001319 **
## Occupation1           4.166e+01  5.228e+01    0.797 0.425469
## Occupation12          3.734e+02  5.346e+01    6.984 2.87e-12 ***
## Occupation17          2.386e+02  5.292e+01    4.509 6.51e-06 ***
## Occupation0           5.878e+01  5.061e+01    1.161 0.245525
## Occupation3           3.168e+02  5.645e+01    5.613 1.99e-08 ***
## Occupation4           2.352e+02  5.152e+01    4.566 4.98e-06 ***
## Occupation11          1.962e+02  5.952e+01    3.296 0.000982 ***
## Occupation8          -3.483e+02  1.012e+02   -3.442 0.000578 ***
## Occupation19         -2.447e+02  5.986e+01   -4.089 4.34e-05 ***
## Occupation2           1.249e+02  5.417e+01    2.307 0.021075 *
## Occupation18          9.102e+00  6.587e+01    0.138 0.890110
## Occupation5           1.228e+02  5.917e+01    2.076 0.037895 *
## Occupation14          2.779e+02  5.407e+01    5.141 2.74e-07 ***
## Occupation13          1.510e+02  6.567e+01    2.299 0.021514 *
## Occupation6           2.906e+02  5.583e+01    5.205 1.94e-07 ***
## City_CategoryB        1.300e+02  1.170e+01   11.117  < 2e-16 ***
## City_CategoryC        5.467e+02  1.269e+01   43.091  < 2e-16 ***
## Product_Category_11   3.503e+03  2.615e+01  133.941  < 2e-16 ***
## Product_Category_112 -8.782e+03  6.054e+01 -145.061  < 2e-16 ***
## Product_Category_18  -2.604e+03  2.657e+01  -98.018  < 2e-16 ***
## Product_Category_15  -3.844e+03  2.603e+01 -147.638  < 2e-16 ***
## Product_Category_14  -7.752e+03  4.040e+01 -191.882  < 2e-16 ***
## Product_Category_12   1.135e+03  3.323e+01   34.148  < 2e-16 ***
## Product_Category_16   5.752e+03  3.451e+01  166.654  < 2e-16 ***
## Product_Category_114  3.003e+03  9.272e+01   32.390  < 2e-16 ***
## Product_Category_111 -5.385e+03  3.313e+01 -162.541  < 2e-16 ***
## Product_Category_113 -9.388e+03  5.285e+01 -177.636  < 2e-16 ***
## Product_Category_115  4.755e+03  5.016e+01   94.814  < 2e-16 ***
## Product_Category_17   6.362e+03  6.209e+01  102.459  < 2e-16 ***
## Product_Category_116  4.667e+03  4.284e+01  108.927  < 2e-16 ***
```

```
## Product_Category_118 -7.179e+03  6.738e+01 -106.546  < 2e-16 ***
## Product_Category_110  9.563e+03  5.476e+01  174.625  < 2e-16 ***
## Product_Category_117 -5.514e+01  1.475e+02   -0.374 0.708531
## Product_Category_19   5.420e+03  1.714e+02   31.617  < 2e-16 ***
## Product_Category_120 -9.843e+03  7.321e+01 -134.460  < 2e-16 ***
## Product_Category_119 -1.020e+04  9.006e+01 -113.301  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3011 on 412503 degrees of freedom
## Multiple R-squared:  0.6405, Adjusted R-squared:  0.6404
## F-statistic: 1.564e+04 on 47 and 412503 DF,  p-value: < 2.2e-16
```

```
pred <- predict(lm1, newdata = test)
cor(pred, test$Purchase)
```

```
## [1] 0.7991911
```

```
mse <- mean((pred-test$Purchase)^2)
mse
```

```
## [1] 9128462
```

```
rmse <- sqrt(mse)
rmse
```

```
## [1] 3021.334
```

Algorithm 2: Decision Tree

- code to run the algorithms
- commentary on feature selection you selected and why
- code to compute your metrics for evaluation as well as commentary discussing the results

Commentary of feature selected: I decided to use Age, Occupation, and City Category predictors because they seem to be a good indication of the amount of disposable income a person has available. The more disposable income they have, the more money they are able to spend on black Friday.

Commentary on Results: The decision tree had an accuracy of 78.9 percent which is very close to the multiple linear regression model. The mse of the decision tree is slightly higher than at 9,524,735. This means the decision tree did slightly worse overall since it was not able to minimize the errors as well as the multiple linear regression model.

```
library(tree)
tree1 <- tree(Purchase~Age+Occupation+City_Category+Product_Category_1, data=train)
summary(tree1)
```

```
##
## Regression tree:
## tree(formula = Purchase ~ Age + Occupation + City_Category +
##     Product_Category_1, data = train)
## Variables actually used in tree construction:
## [1] "Product_Category_1"
## Number of terminal nodes:  6
## Residual mean deviance:  9466000 = 3.905e+12 / 412500
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -12590.0  -1633.0    381.4      0.0   2055.0   7838.0
```

```
pred_tree <- predict(tree1,newdata = test)
cor_tree <- cor(pred_tree, test$Purchase)
print(paste("Correlation:", cor_tree))
```

```
## [1] "Correlation: 0.789310840249841"
```

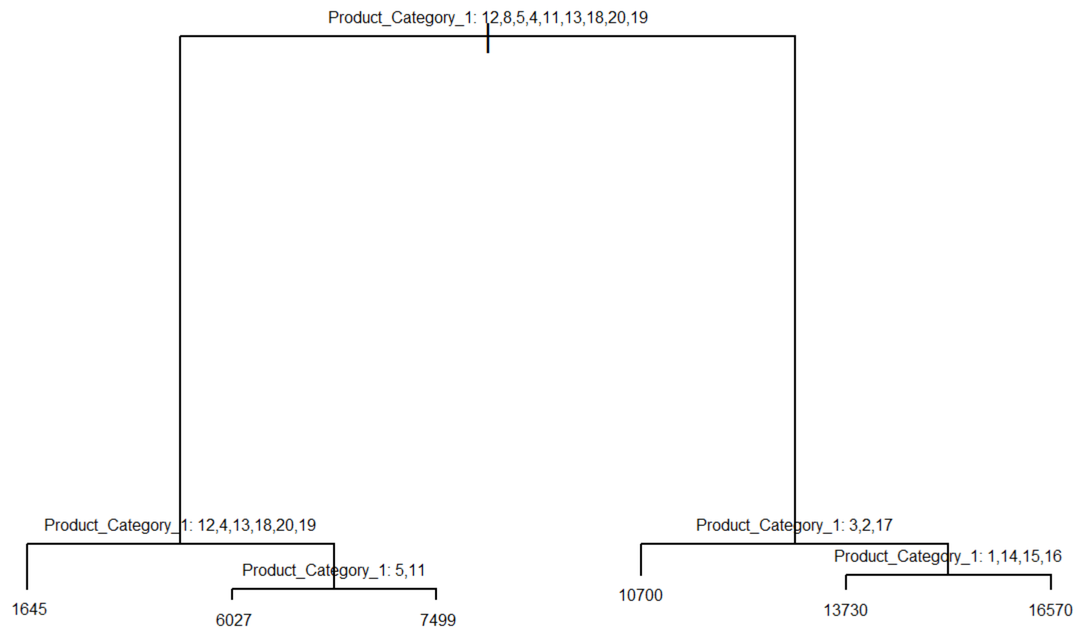
```
mse_tree <- mean((pred_tree-test$Purchase)^2)
mse_tree
```

```
## [1] 9524735
```

```
rmse_tree <- sqrt(mean((pred_tree-test$Purchase)^2))
print(paste("rmse:",rmse_tree))
```

```
## [1] "rmse: 3086.21692192795"
```

```
plot(tree1)
text(tree1, cex=.5, pretty = 0)
```



Algorithm 3: Simple Linear Regression * code to run the algorithms * commentary on feature selection you selected and why * code to compute your metrics for evaluation as well as commentary discussing the results

Commentary on Features: For this simple linear regression model I decided to predict on Occupation. My thought process was that the salary amount alone is enough to make accurate predictions on how much a person would spend on black friday.

Commentary on Results: Looking at the results we see that my thought process was incorrect. The model had a very low accuracy at .05% and a very large mse of 25,183,647. The model had terrible results which means that Occupation is a poor predictor for the target value which is Purchase amount.

```
lm2 <- lm(Purchase~Occupation, data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = Purchase ~ Occupation, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9811  -3556  -1138   2874  15258
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8992.70      50.88 176.746  < 2e-16 ***
## Occupation16   397.70      62.51   6.362 1.99e-10 ***
## Occupation15   789.67      73.00  10.817  < 2e-16 ***
## Occupation7    447.35      56.17   7.964 1.67e-15 ***
## Occupation20  -146.10      59.90  -2.439 0.014733 *
## Occupation9   -364.57      89.08  -4.093 4.27e-05 ***
## Occupation1    -46.70      57.39  -0.814 0.415838
## Occupation12   825.20      60.54  13.630  < 2e-16 ***
## Occupation17   830.38      58.54  14.185  < 2e-16 ***
## Occupation0    133.89      55.41   2.416 0.015680 *
## Occupation3    156.10      66.97   2.331 0.019764 *
## Occupation4    226.42      55.24   4.099 4.16e-05 ***
## Occupation11   221.61      73.90   2.999 0.002711 **
## Occupation8    476.32     155.31   3.067 0.002163 **
## Occupation19  -312.17      80.90  -3.859 0.000114 ***
## Occupation2    -54.77      62.03  -0.883 0.377257
## Occupation18   195.64      87.70   2.231 0.025700 *
## Occupation5    341.51      73.14   4.669 3.03e-06 ***
## Occupation14   525.74      61.73   8.516  < 2e-16 ***
## Occupation13   335.48      83.03   4.041 5.33e-05 ***
## Occupation6    273.74      65.12   4.203 2.63e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5013 on 412530 degrees of freedom
## Multiple R-squared:  0.003683,   Adjusted R-squared:  0.003635
## F-statistic: 76.25 on 20 and 412530 DF,  p-value: < 2.2e-16
```

```
pred_m <- predict(lm2, newdata = test)
cor(pred_m, test$Purchase)
```

```
## [1] 0.05667473
```

```
mse2 <- mean((pred_m-test$Purchase)^2)
mse2
```

```
## [1] 25183647
```

```
rmse2 <- sqrt(mse2)
rmse2
```

```
## [1] 5018.331
```

Step 8 Results analysis

- rank the algorithms from best to worst performing on your data
- add commentary on the performance of the algorithms
- your analysis concerning why the best performing algorithm worked best on that data
- commentary on what your script was able to learn from the data (big picture) and if this is likely to be useful

Rank: 1. Multiple Linear Regression 2. Decision Tree 3. Simple Linear Regression

Commentary on the performance: I have most of my commentary of the algorithms above. The reason why I ranked Multiple Linear regression the highest is because it had an accuracy slightly higher than the Decision tree and had a slightly lower mse. The lower mse score means it was able to minimize errors better which is desirable. The simple linear Regression had the worst scores by far. The accuracy was less than 1% and an enormous mse score.

Commentary on Best Preforming Algorithm: The best preforming algorithm was the Multiple Linear Regression model. This is because multiple LR is able to factor in several relationships that are somewhat correlated with the target variable to make predictions on future data. The model I created used enough variables to learn from the data to make accurate predictions.

Big Picture: The big picture takeaway from running these models on this data set is that product category is the best predictor on the amount of money a person will spend on Black Friday. This can be found by running a simple linear regression model on Purchase~Product_Category which will result in an accuracy rate of 78%. Knowing this information companies can focus on advertising their Black Friday sales based on product category rather than other factors. Since we are not given the names of the product category and are only able to see the category number, we can’t say exactly which product category induces the most spending from individuals.