Purchase Predction Model Selection

Due February 20, 2022

Problem Overview

The goal of this homework is hands-on practice with linear regression, logistic regression, classification, and model selection. You will:

1. Conduct basic exploratory analysis of a data set
2. Develop linear and logistic regression models
3. Interpret your models
4. Partition your dataset and evaluate your models in terms of classification performance

The Assignment

The data in the accompanying file “car_sales.csv” (posted on Canvas) contains data from 10,062 car auctions. Auto dealers purchase used cars at auctions with the plan to sell them to consumers, but sometimes these auctioned vehicles can have severe issues that prevent them from being resold. The data contains information about each auctioned vehicle (for instance: the make, color, and age, among other variables). A full data dictionary is given in carvana_data_dictionary.txt (we have included only a subset of the variables in their data set). See http://www.kaggle.com/c/DontGetKicked (http://www.kaggle.com/c/DontGetKicked) for documentation on the problem.

Your task is to develop models to predict the target variable “IsBadBuy”, which labels whether a car purchased at auction was a “bad buy” or not. The intended use case for this model is to help an auto dealership decide whether or not to purchase an individual vehicle.

```
car = read_csv("car_data.csv")  #read the car_data dataset in R
```

```
## Rows: 10062 Columns: 10
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (5): Auction, Make, Color, WheelType, Size
## dbl (5): VehicleAge, VehOdo, MMRAcquisitionAuctionAveragePrice, MMRAcquisiti...
```

```
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
names(car)                      #variables used in dataset
```

```
##  [1] "Auction"                          "VehicleAge"
##  [3] "Make"                             "Color"
##  [5] "WheelType"                        "VehOdo"
##  [7] "Size"                             "MMRAcquisitionAuctionAveragePrice"
##  [9] "MMRAcquisitionRetailAveragePrice" "IsBadBuy"
```

0: Example answer

What is the mean of VehicleAge variable?

ANSWER: The mean age of a vehicle in this dataset is 4.504969.

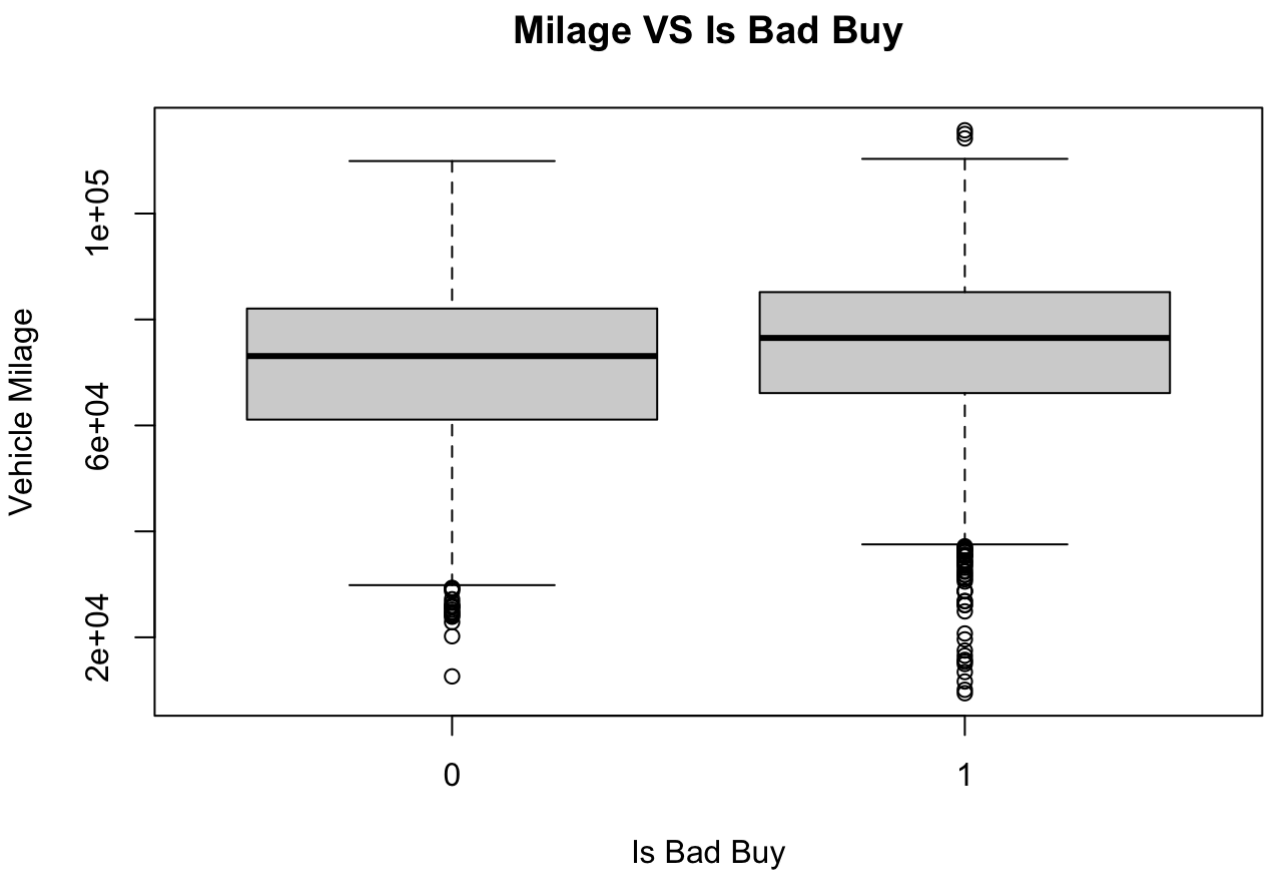
```
age_mean <- car %>%
  summarise(mean_age = mean(VehicleAge))
```

1: EDA and Data Cleaning

a. Construct and report boxplots of VehOdo and VehAge (broken up by values of IsBadBuy). Does it appear there is a relationship between either of these numerical variables and IsBadBuy?

ANSWER TO QUESTION 1a HERE: There is no specific relationship between these 2 numeric variables and and whether a car is a bad buy or not. The graph only points out that the median Vehodo and the median VehAge is higher when IsBadBuy = 1, which may indicate that older cars traveled more and they could be potentially bad buys. However, since we just compared medians. We cannot establish strong relationship between both sides.

```
#Boxplots of VehOdo VS IsBadGuy
VehOdo_boxplot=boxplot(car$VehOdo~car$IsBadBuy,main="Milage VS Is Bad Buy",
xlab="Is Bad Buy", ylab="Vehicle Milage")
```



b. Construct a two-way table of IsBadBuy by Make. Does it appear that any vehicle makes are particularly problematic?

ANSWER TO QUESTION 1b HERE: All the Lexus, Plymouth cars in this data set have only bad buys, which means 100% bad buys. Other very problematic makes include Infiniti (8 out of 10 bad buys), OLDSMOBILE (31 out of 43), Subaru (3 out of 4), etc.

```
table(car$Make, car$IsBadBuy)
```

```
##
##                 0    1
##   ACURA         4    5
##   BUICK        43   60
##   CADILLAC      1    2
##   CHEVROLET  1191  930
##   CHRYSLER    604  613
##   DODGE       911  742
##   FORD        774  990
##   GMC          42   43
##   HONDA        41   36
##   HYUNDAI     115  124
##   INFINITI      2    8
##   ISUZU        10    5
##   JEEP        108  134
##   KIA         203  169
##   LEXUS         0    8
##   LINCOLN       7   16
##   MAZDA        73   95
##   MERCURY      61   91
##   MINI          3    5
##   MITSUBISHI   81   65
##   NISSAN      138  191
##   OLDSMOBILE   12   31
##   PLYMOUTH      0    1
##   PONTIAC     317  280
##   SATURN      132  165
##   SCION        11    7
##   SUBARU        1    3
##   SUZUKI       84  110
##   TOYOTA       78   65
##   VOLKSWAGEN    8   10
##   VOLVO         3    0
```

c. Construct the following new variables :

- MPYind = 1 when the miles/year is above the median and 0 otherwise
- VehType which has the following values:
    - SUV when Size is LARGE SUV, MEDIUM SUV, or SMALL SUV
    - Truck when Size is Large Truck, Medium Truck, or Small Truck
    - Regular when Size is VAN, CROSSOVER, LARGE, or MEDIUM
    - Small when size is COMPACT, SPECIALTY, or SPORT Hint: there are lots of ways to do this one, but case_when might be a useful function that's part of the tidyverse
    - Price0 which is 1 when either the MMRAcquisitionRetailAveragePrice or MMRAcquisitionAuctionAveragePrice are equal to 0, and 0 otherwise

Also, modify these two existing variables:

- The value of Make should be replaced with "other_make" when there are fewer than 20 cars with that make
- The value of Color should be replaced with "other_color" when there are fewer than 20 cars with that color

ANSWER TO QUESTION 1c HERE:

```
car = car %>%
  mutate(MilesPerYear = VehOdo/VehicleAge)

car = car %>%
  mutate(MPYind = ifelse(MilesPerYear > median(MilesPerYear), 1, 0), # When the miles pe
r year is above the median, the variable MPYind should be marked as 1, otherwise 0.
         VehType = case_when(Size %in% c("LARGE SUV", "MEDIUM SUV","SMALL SUV") ~ "SUV",
                             Size %in% c("LARGE TRUCK", "MEDIUM TRUCK","SMALL TRUCK") ~
"Truck",
                             Size %in% c("VAN", "CROSSOVER", "LARGE", "MEDIUM") ~ "Regul
ar",
                             Size %in% c("COMPACT", "SPECIALTY", "SPORTS") ~ "Small"),
                            Price0 = ifelse(MMRAcquisitionRetailAveragePrice == 0 | MMRA
cquisitionAuctionAveragePrice == 0, 1, 0))
```

d. The rows where MMRAcquisitionRetailAveragePrice or MMRAcquisitionAuctionAveragePrice are equal to 0 are suspicious - it seems like those values might not be correct. Replace the two prices with the average grouped by vehicle make. Be sure to remove the 0's from the average calculation! Hint: this one is a little tricky. Consider using the special character NA to replace the 0's.

ANSWER TO QUESTION 1d HERE:

```
car_clean = car %>%
  mutate(MMRAcquisitionAuctionAveragePrice =ifelse(MMRAcquisitionAuctionAveragePrice ==
0, NA, MMRAcquisitionAuctionAveragePrice),
         MMRAcquisitionRetailAveragePrice = ifelse(MMRAcquisitionRetailAveragePrice == 0
, NA, MMRAcquisitionRetailAveragePrice)) %>%
  group_by(Make) %>%
         mutate(MMRAcquisitionAuctionAveragePrice = ifelse(is.na(MMRAcquisitionAuctionAv
eragePrice), mean(MMRAcquisitionAuctionAveragePrice, na.rm = TRUE), MMRAcquisitionAuctio
nAveragePrice)) %>%  #when the value for MMRAcquisitionAuctionAveragePrice is NA, replac
e it with the average price of its make.
         mutate(MMRAcquisitionRetailAveragePrice = ifelse(is.na(MMRAcquisitionRetailAver
agePrice), mean(MMRAcquisitionRetailAveragePrice, na.rm = TRUE), MMRAcquisitionRetailAve
ragePrice))
```

2: Linear Regression

a. Train a linear regression to predict IsBadBuy using the variables listed below. Report the R^2.

- Auction
- VehicleAge
- Make
- Color
- WheelType
- VehOdo
- MPYind
- VehType
- MMRAcquisitionAuctionAveragePrice
- MMRAcquisitionRetailAveragePrice

ANSWER TO QUESTION 2a HERE: Multiple R-squared: 0.1917, Adjusted R-squared: 0.187

```
car_clean = car_clean %>%
  mutate(Auction = as.factor(Auction),
         Make = as.factor(Make),
         Color = as.factor(Color),
         WheelType = as.factor(WheelType),
         MPYind = as.factor(MPYind),
         VehType = as.factor(VehType),
         IsBadBuy = as.numeric(IsBadBuy)) #make IsBadBuy an numeric variable so it could
be the dependent variable in linear regression

linreg1 = lm(data = car_clean, IsBadBuy ~ Auction + VehicleAge + Make + Color + WheelTyp
e + VehOdo + MPYind + VehType + MMRAcquisitionAuctionAveragePrice + MMRAcquisitionRetail
AveragePrice)

summary(linreg1)
```

```
##
## Call:
## lm(formula = IsBadBuy ~ Auction + VehicleAge + Make + Color +
##     WheelType + VehOdo + MPYind + VehType + MMRAcquisitionAuctionAveragePrice +
##     MMRAcquisitionRetailAveragePrice, data = car_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2697 -0.3934 -0.1606  0.4672  1.0413
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -5.564e-02  1.604e-01  -0.347 0.728664
## AuctionMANHEIM                 4.443e-02  1.200e-02   3.702 0.000215 ***
## AuctionOTHER                   8.808e-03  1.366e-02   0.645 0.519139
## VehicleAge                     4.687e-02  5.572e-03   8.412  < 2e-16 ***
## MakeBUICK                      1.534e-01  1.579e-01   0.971 0.331324
## MakeCADILLAC                   2.170e-01  3.009e-01   0.721 0.470688
## MakeCHEVROLET                  1.119e-01  1.520e-01   0.736 0.461732
## MakeCHRYSLER                   2.005e-01  1.524e-01   1.315 0.188422
## MakeDODGE                      1.553e-01  1.522e-01   1.021 0.307484
## MakeFORD                       1.773e-01  1.519e-01   1.167 0.243293
## MakeGMC                        1.147e-01  1.595e-01   0.719 0.472110
## MakeHONDA                      3.673e-02  1.595e-01   0.230 0.817878
## MakeHYUNDAI                    1.579e-01  1.546e-01   1.021 0.307196
## MakeINFINITI                   3.768e-01  2.080e-01   1.811 0.070105 .
## MakeISUZU                     -4.216e-02  1.919e-01  -0.220 0.826107
## MakeJEEP                       1.606e-01  1.549e-01   1.037 0.299853
## MakeKIA                        1.731e-01  1.540e-01   1.124 0.261122
## MakeLEXUS                      8.872e-01  2.221e-01   3.995 6.52e-05 ***
## MakeLINCOLN                    2.265e-01  1.779e-01   1.273 0.203022
## MakeMAZDA                      1.889e-01  1.554e-01   1.215 0.224241
## MakeMERCURY                    1.940e-01  1.560e-01   1.243 0.213808
## MakeMINI                       3.146e-01  2.211e-01   1.423 0.154836
## MakeMITSUBISHI                 3.991e-02  1.560e-01   0.256 0.798077
## MakeNISSAN                     1.895e-01  1.533e-01   1.237 0.216209
## MakeOLDSMOBILE                 2.305e-01  1.667e-01   1.382 0.166964
## MakePLYMOUTH                   4.109e-01  4.759e-01   0.863 0.387920
## MakePONTIAC                    1.414e-01  1.528e-01   0.925 0.354997
## MakeSATURN                     1.881e-01  1.541e-01   1.220 0.222327
## MakeSCION                      1.644e-01  1.849e-01   0.889 0.374031
## MakeSUBARU                     3.634e-01  2.721e-01   1.336 0.181713
## MakeSUZUKI                     2.864e-01  1.560e-01   1.836 0.066442 .
## MakeTOYOTA                     1.372e-01  1.558e-01   0.881 0.378516
## MakeVOLKSWAGEN                 1.965e-01  1.843e-01   1.066 0.286453
## MakeVOLVO                     -2.508e-01  3.009e-01  -0.833 0.404744
## ColorBLUE                     -8.856e-03  1.883e-02  -0.470 0.638137
## ColorGREY                     -7.574e-03  1.997e-02  -0.379 0.704464
## ColorSILVER                    1.712e-02  1.745e-02   0.982 0.326365
## ColorWHITE                     1.240e-02  1.824e-02   0.680 0.496513
## ColorBEIGE                    -2.348e-02  3.274e-02  -0.717 0.473352
## ColorGOLD                      3.073e-02  2.197e-02   1.399 0.161916
```

```
## ColorGREEN                          -2.409e-02  2.595e-02  -0.928 0.353241
## ColorMAROON                          6.640e-02  3.063e-02   2.168 0.030195 *
## ColorRED                             1.647e-02  2.099e-02   0.785 0.432729
## Color'NOT AVAIL'                    -1.600e-02  8.999e-02  -0.178 0.858877
## ColorBROWN                           1.222e-02  5.790e-02   0.211 0.832876
## ColorORANGE                         -3.302e-02  7.055e-02  -0.468 0.639774
## ColorOTHER                          -1.722e-01  7.594e-02  -2.267 0.023402 *
## ColorPURPLE                          4.421e-02  6.188e-02   0.715 0.474926
## ColorYELLOW                         -1.048e-01  7.723e-02  -1.358 0.174648
## ColorNULL                           -5.062e-01  3.201e-01  -1.581 0.113807
## WheelTypeNULL                        5.192e-01  1.507e-02  34.452  < 2e-16 ***
## WheelTypeCovers                     -2.690e-02  1.110e-02  -2.423 0.015392 *
## WheelTypeSpecial                    -1.316e-02  4.587e-02  -0.287 0.774245
## VehOdo                               2.487e-06  3.968e-07   6.268 3.80e-10 ***
## MPYind1                             -8.214e-03  1.513e-02  -0.543 0.587181
## VehTypeRegular                      -6.603e-02  1.381e-02  -4.782 1.76e-06 ***
## VehTypeSUV                          -4.386e-02  1.937e-02  -2.264 0.023597 *
## VehTypeTruck                        -8.574e-02  2.440e-02  -3.514 0.000444 ***
## MMRAcquisitionAuctionAveragePrice -6.989e-06  5.507e-06  -1.269 0.204413
## MMRAcquisitionRetailAveragePrice   1.129e-06  3.597e-06   0.314 0.753690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4509 on 10002 degrees of freedom
## Multiple R-squared:  0.1917, Adjusted R-squared:  0.187
## F-statistic: 40.21 on 59 and 10002 DF,  p-value: < 2.2e-16
```

b. What is the predicted value of IsBadBuy for a MANHEIM Auction, 4-year-old Compact Blue Volvo with 32000 miles, WheelType = Special, an MMR Auction Price of $8000, and an MMR Retail Price of $12000? What would be your predicted classification for the car, using a cutoff of 0.5?

ANSWER TO QUESTION 2b HERE: The predicted value for IsBadBuy for the test data is -0.0593. Having a cutoff being 0.5, the car would be classified as 0 which means the car is a good buy.

```
test1 = as.data.frame(car_clean)
test1 = test1[0,] #clean the observations
test1 = data.frame(Auction="MANHEIM", VehicleAge=4, Make="VOLVO", Color="BLUE", WheelTyp
e="Special", VehOdo=32000,Size="COMPACT", MMRAcquisitionAuctionAveragePrice=8000,VehType
="Small", MMRAcquisitionRetailAveragePrice=12000,MPYind="0") #fill in the values mention
ed above to the test1
IsBadBuylinreg = predict(linreg1,test1) #using the values filled to predict if the car i
s a good buy.
```

c. Do you have any reservations about this predicted IsBadBuy? That is, would you feel sufficiently comfortable with this prediction in order to take action based on it? Why or why not?

ANSWER TO QUESTION 2c HERE: It's not sufficient to predict categorical variables using a linear regression since the values of predicted model could be out of the range 0 to 1. -0.059 could not be used to predict 0

3: Logistic Regression

a. Train a Logistic Regression model using the same variables as in 2a. Report the AIC of your model.

ANSWER TO QUESTION 3a HERE: The AIC value for the logistic model is 11766.

```
logitic_model1= glm(data=car_clean, IsBadBuy ~ Auction + VehicleAge + Make + Color + Whe
elType + VehOdo + MPYind + VehType +MMRAcquisitionAuctionAveragePrice + MMRAcquisitionRe
tailAveragePrice,family="binomial")

summary(logitic_model1)
```

```
##
## Call:
## glm(formula = IsBadBuy ~ Auction + VehicleAge + Make + Color +
##     WheelType + VehOdo + MPYind + VehType + MMRAcquisitionAuctionAveragePrice +
##     MMRAcquisitionRetailAveragePrice, family = "binomial", data = car_clean)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1133  -0.9796  -0.5271   1.0917   2.3471
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -2.872e+00  7.946e-01  -3.614 0.000301 ***
## AuctionMANHEIM                   2.025e-01  5.993e-02   3.379 0.000726 ***
## AuctionOTHER                     3.379e-02  7.211e-02   0.469 0.639426
## VehicleAge                       2.398e-01  2.817e-02   8.511  < 2e-16 ***
## MakeBUICK                        8.238e-01  7.821e-01   1.053 0.292162
## MakeCADILLAC                     1.202e+00  1.600e+00   0.751 0.452393
## MakeCHEVROLET                    6.060e-01  7.519e-01   0.806 0.420312
## MakeCHRYSLER                     1.026e+00  7.538e-01   1.361 0.173374
## MakeDODGE                        8.032e-01  7.526e-01   1.067 0.285858
## MakeFORD                         9.216e-01  7.514e-01   1.226 0.220019
## MakeGMC                          6.041e-01  7.861e-01   0.768 0.442196
## MakeHONDA                        2.474e-01  7.946e-01   0.311 0.755491
## MakeHYUNDAI                      8.271e-01  7.646e-01   1.082 0.279337
## MakeINFINITI                     2.509e+00  1.202e+00   2.087 0.036897 *
## MakeISUZU                       -1.546e-01  9.870e-01  -0.157 0.875573
## MakeJEEP                         8.385e-01  7.658e-01   1.095 0.273497
## MakeKIA                          9.067e-01  7.620e-01   1.190 0.234071
## MakeLEXUS                        1.561e+01  1.729e+02   0.090 0.928057
## MakeLINCOLN                      1.125e+00  8.852e-01   1.271 0.203717
## MakeMAZDA                        9.647e-01  7.684e-01   1.255 0.209314
## MakeMERCURY                      9.973e-01  7.710e-01   1.294 0.195829
## MakeMINI                         1.499e+00  1.088e+00   1.378 0.168110
## MakeMITSUBISHI                   2.109e-01  7.732e-01   0.273 0.784988
## MakeNISSAN                       9.866e-01  7.580e-01   1.302 0.193024
## MakeOLDSMOBILE                   1.183e+00  8.318e-01   1.423 0.154821
## MakePLYMOUTH                     1.308e+01  5.354e+02   0.024 0.980515
## MakePONTIAC                      7.527e-01  7.559e-01   0.996 0.319368
## MakeSATURN                       9.818e-01  7.622e-01   1.288 0.197662
## MakeSCION                        8.699e-01  9.119e-01   0.954 0.340103
## MakeSUBARU                       1.734e+00  1.396e+00   1.242 0.214247
## MakeSUZUKI                       1.468e+00  7.719e-01   1.901 0.057276 .
## MakeTOYOTA                       6.853e-01  7.706e-01   0.889 0.373852
## MakeVOLKSWAGEN                   1.037e+00  9.071e-01   1.143 0.253135
## MakeVOLVO                       -1.234e+01  3.021e+02  -0.041 0.967419
## ColorBLUE                       -5.179e-02  9.357e-02  -0.554 0.579918
## ColorGREY                       -3.757e-02  9.905e-02  -0.379 0.704417
## ColorSILVER                      8.006e-02  8.667e-02   0.924 0.355644
## ColorWHITE                       5.570e-02  9.065e-02   0.614 0.538935
## ColorBEIGE                      -1.460e-01  1.682e-01  -0.868 0.385318
## ColorGOLD                        1.497e-01  1.088e-01   1.377 0.168625
```

```
## ColorGREEN                          -1.097e-01  1.275e-01  -0.860 0.389820
## ColorMAROON                          3.338e-01  1.509e-01   2.211 0.027004 *
## ColorRED                             7.724e-02  1.042e-01   0.741 0.458553
## Color'NOT AVAIL'                    -1.692e-01  5.692e-01  -0.297 0.766322
## ColorBROWN                           5.529e-02  2.753e-01   0.201 0.840834
## ColorORANGE                         -1.615e-01  3.684e-01  -0.438 0.661217
## ColorOTHER                          -1.373e+00  4.725e-01  -2.905 0.003668 **
## ColorPURPLE                          3.123e-01  3.172e-01   0.985 0.324814
## ColorYELLOW                         -5.349e-01  3.766e-01  -1.421 0.155454
## ColorNULL                           -3.446e+00  1.498e+00  -2.301 0.021383 *
## WheelTypeNULL                        3.473e+00  1.369e-01  25.370  < 2e-16 ***
## WheelTypeCovers                     -7.492e-02  5.289e-02  -1.417 0.156602
## WheelTypeSpecial                    -6.288e-02  2.119e-01  -0.297 0.766619
## VehOdo                               1.315e-05  1.986e-06   6.623 3.51e-11 ***
## MPYind1                             -2.977e-02  7.404e-02  -0.402 0.687599
## VehTypeRegular                      -3.325e-01  6.851e-02  -4.852 1.22e-06 ***
## VehTypeSUV                          -2.232e-01  9.605e-02  -2.324 0.020141 *
## VehTypeTruck                        -4.240e-01  1.190e-01  -3.562 0.000369 ***
## MMRAcquisitionAuctionAveragePrice -3.002e-05  2.768e-05  -1.085 0.278116
## MMRAcquisitionRetailAveragePrice   3.854e-06  1.781e-05   0.216 0.828691
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13949  on 10061  degrees of freedom
## Residual deviance: 11646  on 10002  degrees of freedom
## AIC: 11766
##
## Number of Fisher Scoring iterations: 12
```

b. What is the coefficient for VehicleAge? Provide a precise (numerical) interpretation of the coefficient.

ANSWER TO QUESTION 3b HERE: The coefficient for VehAge is 0.2398 which means that with everything else holding constant, generally if the VehAge increases by 1 unit, the odds that Vehicle becoming a bad buy increases by e^(0.2398) and the corresponding probability also increases.

c. What is the coefficient for VehType = Small? Provide a precise (numerical) interpretation of this coefficient.

ANSWER TO QUESTION 3c HERE: The coefficient for VehType = Small is 0 which means that with everything else holding constant, generally if the VehAge increases by 1 unit, the odds that Vehicle becoming a bad buy won't change and the corresponding probability also won't change

d. Compute the predicted probability that the same car as in #2b is a bad buy. Hint: you should use the predict function, but you need to specify type = "response" when predicting probabilities from logistic regression (otherwise, it will predict the value of logit). For example: predict(mymodel, newdata = mydata, type = "response").

ANSWER TO QUESTION 3d HERE: The predicted value for the test data is 8.84e-07

```
test2 = as.data.frame(car)
test2 = test2[0,]
test2 = data.frame(Auction="MANHEIM", VehicleAge=4, Make="VOLVO", Color="BLUE", WheelTyp
e="Special", VehOdo=32000,Size="COMPACT", MMRAcquisitionAuctionAveragePrice=8000,VehType
="Small", MMRAcquisitionRetailAveragePrice=12000,MPYind="0")
IsBadBuylogreg = predict(logitic_model1, test2, type = "response")
```

e. If you were to pick one model to use for the purposes of inference (explaining the relationship between the features and the target variable) which would it be, and why?

ANSWER TO QUESTION 3e HERE: Linear regression cannot be used for explaining the relationship between the features and the target variable. For logistic model, the values can be within the range of 0 and 1, which is more accurate to used in categorical prediction.

4: Classification and Evaluation

a. Split the data into 70% training and 30% validation sets, retrain the linear and logistic regression models using the training data only, and report the resulting R^2 and AIC, respectively.

ANSWER TO QUESTION 4a HERE: The R2 is 0.1957 and adjusted R2 is 0.1889 for linear model. The AIC value for logistic model is 8237.1

```
set.seed(1)

train_insts = sample(nrow(car), .7*nrow(car)) #Split the data into 70% training and 30%
 validation sets

data_train = car[train_insts,] #assign the 70% data into training data set
data_valid = car[-train_insts,] #assign the rest into validation data set

lm2 = lm(data = data_train, IsBadBuy ~ Auction + VehicleAge + Make + Color + WheelType +
VehOdo + MPYind + VehType + MMRAcquisitionAuctionAveragePrice + MMRAcquisitionRetailAver
agePrice)

logitic_model2 = glm(data = data_train, IsBadBuy ~ Auction + VehicleAge + Make + Color +
WheelType + VehOdo + MPYind + VehType + MMRAcquisitionAuctionAveragePrice + MMRAcquisiti
onRetailAveragePrice, family = "binomial")

summary(lm2)
```

```
##
## Call:
## lm(formula = IsBadBuy ~ Auction + VehicleAge + Make + Color +
##     WheelType + VehOdo + MPYind + VehType + MMRAcquisitionAuctionAveragePrice +
##     MMRAcquisitionRetailAveragePrice, data = data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2461 -0.3893 -0.1552  0.4653  0.9872
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -8.957e-02  2.043e-01  -0.438 0.661086
## AuctionMANHEIM                 5.521e-02  1.441e-02   3.832 0.000128 ***
## AuctionOTHER                   2.184e-02  1.636e-02   1.335 0.181850
## VehicleAge                     4.135e-02  6.474e-03   6.387 1.80e-10 ***
## MakeBUICK                      8.042e-02  1.685e-01   0.477 0.633238
## MakeCADILLAC                   4.193e-01  3.566e-01   1.176 0.239660
## MakeCHEVROLET                  3.710e-02  1.614e-01   0.230 0.818183
## MakeCHRYSLER                   1.213e-01  1.619e-01   0.749 0.453638
## MakeDODGE                      7.518e-02  1.616e-01   0.465 0.641692
## MakeFORD                       1.300e-01  1.612e-01   0.807 0.419825
## MakeGMC                        5.055e-02  1.714e-01   0.295 0.768038
## MakeHONDA                     -4.756e-02  1.702e-01  -0.279 0.779963
## MakeHYUNDAI                    9.011e-02  1.649e-01   0.546 0.584763
## MakeINFINITI                   7.631e-02  2.575e-01   0.296 0.766979
## MakeISUZU                     -7.541e-02  2.045e-01  -0.369 0.712343
## MakeJEEP                       8.865e-02  1.650e-01   0.537 0.591076
## MakeKIA                        1.170e-01  1.641e-01   0.713 0.475954
## MakeLEXUS                      7.900e-01  2.366e-01   3.338 0.000847 ***
## MakeLINCOLN                    1.250e-01  1.960e-01   0.638 0.523761
## MakeMAZDA                      1.098e-01  1.664e-01   0.660 0.509376
## MakeMERCURY                    1.375e-01  1.668e-01   0.824 0.409871
## MakeMINI                       2.721e-01  2.440e-01   1.115 0.264856
## MakeMITSUBISHI                -7.717e-02  1.670e-01  -0.462 0.644093
## MakeNISSAN                     1.277e-01  1.631e-01   0.783 0.433675
## MakeOLDSMOBILE                 1.533e-01  1.787e-01   0.858 0.391092
## MakePLYMOUTH                   3.533e-01  4.786e-01   0.738 0.460406
## MakePONTIAC                    8.211e-02  1.624e-01   0.505 0.613236
## MakeSATURN                     1.300e-01  1.644e-01   0.790 0.429391
## MakeSCION                      1.413e-01  2.008e-01   0.703 0.481794
## MakeSUBARU                     2.420e-01  3.065e-01   0.790 0.429802
## MakeSUZUKI                     2.039e-01  1.667e-01   1.223 0.221222
## MakeTOYOTA                     2.553e-02  1.669e-01   0.153 0.878454
## MakeVOLKSWAGEN                 2.243e-01  2.059e-01   1.089 0.276041
## MakeVOLVO                     -3.426e-01  3.054e-01  -1.122 0.261931
## ColorBEIGE                     3.273e-03  1.157e-01   0.028 0.977438
## ColorBLACK                     3.923e-02  1.112e-01   0.353 0.724332
## ColorBLUE                      2.504e-02  1.109e-01   0.226 0.821391
## ColorBROWN                     5.474e-02  1.265e-01   0.433 0.665315
## ColorGOLD                      7.282e-02  1.116e-01   0.652 0.514198
## ColorGREEN                     5.148e-02  1.130e-01   0.456 0.648679
```

```
## ColorGREY                           3.218e-02  1.112e-01   0.289 0.772282
## ColorMAROON                         9.862e-02  1.145e-01   0.861 0.389211
## ColorNULL                          -8.427e-01  4.640e-01  -1.816 0.069414 .
## ColorORANGE                         2.174e-02  1.376e-01   0.158 0.874474
## ColorOTHER                         -1.409e-01  1.410e-01  -0.999 0.317669
## ColorPURPLE                         9.196e-02  1.303e-01   0.706 0.480448
## ColorRED                            5.758e-02  1.114e-01   0.517 0.605363
## ColorSILVER                         6.682e-02  1.106e-01   0.604 0.545646
## ColorWHITE                          4.875e-02  1.107e-01   0.440 0.659792
## ColorYELLOW                        -7.529e-02  1.427e-01  -0.528 0.597764
## WheelTypeCovers                    -2.637e-02  1.326e-02  -1.989 0.046708 *
## WheelTypeNULL                       5.227e-01  1.798e-02  29.073  < 2e-16 ***
## WheelTypeSpecial                    8.217e-03  5.339e-02   0.154 0.877677
## VehOdo                              2.516e-06  4.730e-07   5.321 1.07e-07 ***
## MPYind                             -2.263e-02  1.808e-02  -1.252 0.210778
## VehTypeSmall                        7.904e-02  1.653e-02   4.781 1.78e-06 ***
## VehTypeSUV                          3.117e-02  1.911e-02   1.631 0.103037
## VehTypeTruck                       -1.805e-02  2.584e-02  -0.698 0.484938
## MMRAcquisitionAuctionAveragePrice -1.147e-05  6.494e-06  -1.766 0.077416 .
## MMRAcquisitionRetailAveragePrice   5.599e-06  4.290e-06   1.305 0.191873
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4503 on 6983 degrees of freedom
## Multiple R-squared:  0.1957, Adjusted R-squared:  0.1889
## F-statistic: 28.79 on 59 and 6983 DF,  p-value: < 2.2e-16
```

```
summary(logitic_model2)
```

```
##
## Call:
## glm(formula = IsBadBuy ~ Auction + VehicleAge + Make + Color +
##     WheelType + VehOdo + MPYind + VehType + MMRAcquisitionAuctionAveragePrice +
##     MMRAcquisitionRetailAveragePrice, family = "binomial", data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0789  -0.9726  -0.5011   1.0871   2.2301
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -3.089e+00  1.133e+00  -2.725 0.006423 **
## AuctionMANHEIM                  2.591e-01  7.239e-02   3.580 0.000344 ***
## AuctionOTHER                    1.117e-01  8.697e-02   1.285 0.198940
## VehicleAge                      2.155e-01  3.270e-02   6.589 4.44e-11 ***
## MakeBUICK                       4.512e-01  8.465e-01   0.533 0.594050
## MakeCADILLAC                    1.358e+01  2.884e+02   0.047 0.962427
## MakeCHEVROLET                   2.403e-01  8.110e-01   0.296 0.767011
## MakeCHRYSLER                    6.379e-01  8.134e-01   0.784 0.432899
## MakeDODGE                       4.104e-01  8.118e-01   0.506 0.613185
## MakeFORD                        6.805e-01  8.102e-01   0.840 0.400924
## MakeGMC                         2.932e-01  8.558e-01   0.343 0.731880
## MakeHONDA                      -1.677e-01  8.610e-01  -0.195 0.845605
## MakeHYUNDAI                     4.870e-01  8.283e-01   0.588 0.556587
## MakeINFINITI                    6.057e-01  1.579e+00   0.384 0.701281
## MakeISUZU                      -2.951e-01  1.048e+00  -0.281 0.778359
## MakeJEEP                        4.851e-01  8.286e-01   0.585 0.558238
## MakeKIA                         6.320e-01  8.241e-01   0.767 0.443126
## MakeLEXUS                       1.513e+01  1.831e+02   0.083 0.934153
## MakeLINCOLN                     6.374e-01  9.915e-01   0.643 0.520318
## MakeMAZDA                       5.710e-01  8.348e-01   0.684 0.493959
## MakeMERCURY                     7.118e-01  8.370e-01   0.850 0.395106
## MakeMINI                        1.285e+00  1.238e+00   1.038 0.299487
## MakeMITSUBISHI                 -3.710e-01  8.429e-01  -0.440 0.659783
## MakeNISSAN                      6.779e-01  8.198e-01   0.827 0.408264
## MakeOLDSMOBILE                  7.956e-01  9.030e-01   0.881 0.378289
## MakePLYMOUTH                    1.277e+01  5.354e+02   0.024 0.980966
## MakePONTIAC                     4.627e-01  8.161e-01   0.567 0.570728
## MakeSATURN                      7.051e-01  8.258e-01   0.854 0.393169
## MakeSCION                       7.566e-01  9.993e-01   0.757 0.448976
## MakeSUBARU                      1.131e+00  1.493e+00   0.757 0.448975
## MakeSUZUKI                      1.051e+00  8.373e-01   1.255 0.209506
## MakeTOYOTA                      1.385e-01  8.387e-01   0.165 0.868800
## MakeVOLKSWAGEN                  1.254e+00  1.046e+00   1.199 0.230475
## MakeVOLVO                      -1.280e+01  3.028e+02  -0.042 0.966291
## ColorBEIGE                     -1.863e-02  7.594e-01  -0.025 0.980426
## ColorBLACK                      1.975e-01  7.402e-01   0.267 0.789665
## ColorBLUE                       1.245e-01  7.390e-01   0.168 0.866234
## ColorBROWN                      2.611e-01  7.924e-01   0.330 0.741767
## ColorGOLD                       3.720e-01  7.417e-01   0.502 0.616009
## ColorGREEN                      2.793e-01  7.463e-01   0.374 0.708243
```

```
## ColorGREY                            1.719e-01  7.402e-01   0.232 0.816329
## ColorMAROON                          5.014e-01  7.520e-01   0.667 0.504946
## ColorNULL                           -1.617e+01  5.354e+02  -0.030 0.975912
## ColorORANGE                          1.095e-01  8.514e-01   0.129 0.897704
## ColorOTHER                          -1.167e+00  9.068e-01  -1.287 0.198092
## ColorPURPLE                          5.576e-01  8.163e-01   0.683 0.494599
## ColorRED                             2.942e-01  7.409e-01   0.397 0.691268
## ColorSILVER                          3.379e-01  7.377e-01   0.458 0.646908
## ColorWHITE                           2.394e-01  7.384e-01   0.324 0.745739
## ColorYELLOW                         -3.922e-01  8.548e-01  -0.459 0.646395
## WheelTypeCovers                     -7.232e-02  6.331e-02  -1.142 0.253343
## WheelTypeNULL                        3.489e+00  1.627e-01  21.448  < 2e-16 ***
## WheelTypeSpecial                     4.246e-02  2.461e-01   0.173 0.863031
## VehOdo                               1.340e-05  2.365e-06   5.666 1.46e-08 ***
## MPYind                              -1.045e-01  8.869e-02  -1.178 0.238635
## VehTypeSmall                         4.038e-01  8.210e-02   4.918 8.74e-07 ***
## VehTypeSUV                           1.444e-01  9.440e-02   1.530 0.126039
## VehTypeTruck                        -9.066e-02  1.265e-01  -0.716 0.473688
## MMRAcquisitionAuctionAveragePrice -5.051e-05  3.254e-05  -1.552 0.120615
## MMRAcquisitionRetailAveragePrice   2.592e-05  2.130e-05   1.217 0.223655
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9763.5  on 7042  degrees of freedom
## Residual deviance: 8117.1  on 6983  degrees of freedom
## AIC: 8237.1
##
## Number of Fisher Scoring iterations: 12
```

b. Compute the RMSE in the training and validation sets for the linear model (do not do the classifications, just use the predicted score). Which is better, and does this make sense? Why or why not?

ANSWER TO QUESTION 4b HERE: Linear RMSE = 1.3275 Logistic RMSE = 0.4518. The linear model has a higher RMSE than the logistic model. Using RMSE is no the correct way to compare the linear or logistic model when the target variable is categorical. The more appropriate way is to the TPR and TNR to judge the model.

```
predictions_linear = predict(logitic_model2, newdata = data_valid)
predictions_classify = predict(logitic_model2, newdata = data_valid, type = "response")
linear_RMSE = sqrt(mean((predictions_linear - as.numeric(as.character(data_valid$IsBadBu
y)))^2)) #compute the RMSE value for linear model
logistic_RMSE = sqrt(mean((predictions_classify - as.numeric(as.character(data_valid$IsB
adBuy)))^2)) #compute the RMSE value for logistic model
```

c. For each model, display the confusion matrix resulting from using a cutoff of 0.5 to do the classifications in the validation data set. Report the accuracy, TPR, and FPR. Which model is the most accurate?

ANSWER TO QUESTION 4c HERE: Accuracy for linear model = 0.309374. Accuracy for logistic model = 0.6690957. TPR for linear model = 0.3777 TPR for logistic model = 0.5669. FPR for linear model = 0.0685 FPR for logistic model = 0.2299. Logistic model is more accurate

```
classify = function(scores, cutoff){
  classifications = ifelse(scores > cutoff, 1 ,0)  # Define a function that uses scores
 to classify based on a cutoff c
  return(classifications)}

classification_linear = classify(predictions_linear, 0.5) #cutoff c=0.5
classification_logistic = classify(predictions_classify,0.5)

CM_linear = table(as.numeric(as.character(data_valid$IsBadBuy)), classification_linear)
CM_logistic = table(data_valid$IsBadBuy, classification_logistic)

TP_Linear = CM_linear[2,2]
FP_Linear = CM_linear[1,2]
TN_Linear = CM_linear[1,1]
FN_Linear = CM_linear[2,1]

TP_Logistic = CM_logistic[2,2]
FP_Logistic = CM_logistic[1,2]
TN_Logistic = CM_logistic[1,1]
FN_Logistic = CM_logistic[2,1]

TPR_Linear = TP_Linear/(TP_Linear + FN_Linear)
TNR_Linear = TN_Linear/(TN_Linear + FP_Linear)
TPR_Logistic = TP_Logistic/(TP_Logistic + FN_Logistic)
TNR_Logistic = TN_Logistic/(TN_Logistic + FP_Logistic)

FPR_linear = 1 - TNR_Linear
FPR_Logistic = 1 - TNR_Logistic

accuracy_linear = CM_linear[2,1]/(sum(CM_linear))
accuracy_logistic = (CM_logistic[2,2] + CM_logistic[1,1])/(sum(CM_logistic))
```

d. For the more accurate model, compute the accuracy, TPR, and FPR using cutoffs of .25 and .75 in the validation data. Which cutoff has the highest accuracy, highest TPR, and highest FPR?

ANSWER TO QUESTION 4d HERE: For cutoff 0.25, Accuracy is 0.5568, TPR for 0.25 is 0.95, FPR for 0.25 is 0.8320; For cutoff of .75, Accuracy for 0.75 is 0.6244, FPR is 0.0171. So there's higher accuracy for 0.75 cutoff model.

```
classifications_lower = classify(predictions_classify, 0.25)
classifications_higher = classify(predictions_classify, 0.75)

CM_lower = table(data_valid$IsBadBuy, classifications_lower)
CM_higher = table(data_valid$IsBadBuy, classifications_higher)

accuracy_lower = (CM_lower[2,2] + CM_lower[1,1])/(sum(CM_lower))
accuracy_higher = (CM_higher[2,2] + CM_higher[1,1])/(sum(CM_higher))

TP_Lower = CM_lower[2,2]
FP_Lower = CM_lower[1,2]
TN_Lower = CM_lower[1,1]
FN_Lower = CM_lower[2,1]

TP_Higher = CM_higher[2,2]
FP_Higher = CM_higher[1,2]
TN_Higher = CM_higher[1,1]
FN_Higher = CM_higher[2,1]

TPR_Lower = TP_Lower/(TP_Lower + FN_Lower)
TNR_Lower = TN_Lower/(TN_Lower + FP_Lower)
TPR_Higher = TP_Higher/(TP_Higher + FN_Higher)
TNR_Higher = TN_Higher/(TN_Higher + FP_Higher)

FPR_lower = 1 - TNR_Lower
FPR_Higher = 1 - TNR_Higher
```

e. In your opinion, which cutoff of the three yields the best results for this application? Explain your reasoning.

ANSWER TO QUESTION 4e HERE: The cutoff for 0.5 model has the highest accuracy value, which is 0.6561. We should choose this model