UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

ASSIGNMENT 3

Your objective is to develop models to predict the outcome variable "BadBuy", which labels whether a car purchased at an auction was a "bad buy" (lemon). Your task is to build a model to guide auto dealerships in their decisions on whether to bid for and purchase a vehicle. You can also apply your learning from this analysis to make more data-informed car-buying decisions!

You will use **carvana.csv** which contains data from 10,062 car auctions as provided by Carvana. Auto dealers purchase used cars at auctions with a plan to sell them to consumers, but sometimes these auctioned vehicles can have severe issues that prevent them from being resold at a profit (hence, lemons). The data contains information about each auctioned vehicle.

Data Dictionary

Variable	**Definition**
Auction	Auction provider where vehicle was purchased
Age	The years elapsed since the manufacturer's year (how old is the vehicle)
Make	Vehicle manufacturer
Color	Vehicle color
WheelType	Vehicle wheel type description (Alloy, Covers)
Odo	Vehicle odometer reading
Size	Size category of the vehicle (Compact, SUV, etc.)
MMRAauction	Auction price for this vehicle (in average condition) at the time of purchase
MMRAretail	Retail price for this vehicle (in average condition) at the time of purchase
BadBuy	Whether the vehicle is a bad purchase / lemon ("YES") or a good investment ("NO")

Before you start:

- Load the following libraries in the given order: *tidyverse, tidymodels, plotly, skimr, caret*
- Load the Carvana data and call it dfc
- Explore the dataset using `skim()` etc.

Assignment Instructions

There are two main objectives. The first is to predict the variable BadBuy as a function of the other variables. The second is to build alternative models, measure, and improve performance.

1) **(~5 points) Data preparation**
   a) Load the dataset into R and call it *dfc*. Inspect and describe the data.
   The dataset has five numerical and five categorical variables. BadBuy is the dependent variable and takes two values, 0 and 1.

   b) Set the seed to **52156**. Randomly split the dataset into a training dataset and a test dataset. Use **65%** of the data for training and hold out the remaining **35%** for testing.

2) **(~10 points) Exploratory analysis of the *training* data set**
   a) Construct and report boxplots of the (1) auction prices for the cars, (2) ages of the cars, and (3) odometer of the cars broken out by whether cars are lemons or not. Does it appear that there is a relationship between either of these numerical variables and being a lemon? Describe your observations from the box plots. Please also pay attention to the outliers detected by the box plots and make sense of them.

   The auction price boxplot indicates that there is not much difference between the ranges of the car prices and them being lemons or not. The average price of a good buy may be slightly higher than a lemon, but the difference is not significant. However, when considering the outliers, it is evident that the highest prices paid for cars have more often than not turned out to be lemons, meaning higher the car price, more the chances of it being a lemon.

   Age of the car is where a clear distinction can be observed. Higher the age of the car, more the chance of it being a lemon. There is a clear correlation between the age of the car and it being a lemon or not.

   Lastly, we can see that lower the odometer reading, more the chance of the car not being a lemon. The outliers on the higher end of the odometer reading also indicates that the car will likely be a lemon if that is the case.

   b) Construct and report a table for the count of good cars and lemons broken up by Size (i.e., How many vehicles of each size are lemons?).

Hint: Remember `tally()`? That's one way to do it. You may want to think more systematically and use a combination of summarize(), length(), mutate(), arrange()

i) Which size of vehicle contributes the most to the number of lemons? (That is, which vehicle size has the highest *percentage* of the total lemons?)
The vehicle size that has the highest percentage of the total lemons is Medium.

ii) Because the vehicles of the size you identified in (i) contribute so much to the number of lemons, would you suggest the auto dealership stop purchasing vehicles of that size? Why or why not?
No, would not suggest that the auto dealership stop buying vehicles of that size. This is because that same sized vehicle also brings in the highest percentage of good investment and more cars of that size turns out to be a good investment as well (2122 out of 4108), whereas in case of cars of other sizes, the trend is usually opposite, that is, more cars turn out to be lemons than good investments.

3) (~20 points) Run a linear probability model to predict a lemon using all other variables.

a) Compute and report the RMSE using your model for both the training and the test data sets. Use the predicted values from the regression equation. **Do not** do any classifications yet.
For training set, the RMSE is 0.4479165
For test set, the RMSE is 0.4528846

b) For which dataset is the error smaller? Does this surprise you? Why or why not?
The error is smaller for the training set. This is not surprising considering the training error will usually be less than the test error as the same training data is used to fit the model, which may lead to overfitting. In other words, a fitted model usually adapts to the training data and hence its training error will be smaller.

c) Use a cutoff of 0.5 and do the classification in the test data. Compute and report the confusion matrix (recall to convert BadBuy into a factor for the confusion matrix).

i) Which type of errors (false positives and false negatives) occur more here?
False negatives occur more here, meaning that even though the car is actually a lemon, the model predicts it to be a good investment.

ii) For this problem, do you think a false positive or a false negative is a more serious error? Based on your answer, which metric makes a better objective?
False negative is clearly a more serious error, as it would be a bigger loss for a person to buy a car thinking it is a good investment when in reality, the car is a lemon. Hence, sensitivity would be the best metric to judge the model on.

d) What is the testing accuracy of your model? Based on accuracy, does the model perform better than using a random classifier (i.e., the baseline accuracy)?
Hint 1: Calculate manually if you like, or use the `confusionMatrix()` function.
Hint 2: The baseline accuracy is the accuracy you would achieve if you classified every single class as a member of the most frequent class in the actual test dataset.

The testing accuracy of the model is 67.31% whereas the baseline accuracy is 50.61% meaning our model performs better than using a random classifier.

e) Compute and report the predicted "probability" that the following car is a lemon:

Auction="ADESA"	Age=1	Make="HONDA"	Color="SILVER"
WheelType="Covers"	Odo=10000	Size="LARGE"	
MMRAauction=8000	MMRAretail=10000		

Does the probability your model calculates make sense? Why or why not?

-0.1996 + 1*0.0515 + 0.1114 + 0.0481 - 0.0353 + 10000*0.000002888 - 0.1475 + 8000*0.000001595 + 10000*0.000001126 = -0.1185
Which does not make sense as the predicted probability cannot be negative.

4) **(~25 points) Run a logistic regression model to predict a lemon using all other variables.**
Hint 1: Don't forget to convert your dependent variable BadBuy to a factor in both datasets.
Hint 2: If you haven't yet, switch to using *caret* at this point.

a) Did you receive a rank-deficient fit error? Why do you think so? Figure out the variables causing the problem by running tally() for all your factor variables, and recode them in a way to prevent the error.
Hints: You will need to recode two factor variables:

1. *Color* has two redundant levels that need to be combined.
2. Create a new category for *Make*, call it OTHER, and recode any of the makes with less than 10 observations as OTHER.

Make sure to make the changes in the full dataset, convert BadBuy to a factor, repeat the process of setting the seed to 52156 and splitting the data.
Run your logistic regression again to confirm the rank-deficient fit error is gone.

b) What is the coefficient for Age? Provide an exact numerical interpretation of this coefficient.

Estimated coefficient of age = 0.2785. So, exp(0.2785)= 1.3211
1 year increase in age is associated with an increase in the odds of the
car being a lemon by a factor of 1.3211 (i.e., the odds are ~57%
higher), with the remaining parameters staying the same.

c) What is the coefficient for SizeVAN? Provide an exact numerical interpretation of this coefficient.

Estimated coefficient of SizeVAN = -0.5982. So, exp(-0.5982)= 0.5498
The odds of Van sized car being a lemon are 0.5498 times (i.e., the odds are ~35% lower) the odds of a Compact sized car, with the remaining parameters staying the same.

d) Use a cutoff of 0.5 and do the classification in the test data. Compute and report the confusion matrix for your test data predictions.

e) Compute and report the predicted probability using your logistic model for the same car from 3(e). What does the resulting value tell you about this particular car now? Does the result make more sense than the result in Question 3(e)? Why or why not?

Pro tip: Pipe a confusion matrix (from any model) into tidy() and see what happens!

-2.472 + 0.2785 - 0.6530 + 0.2850 - 0.1082 + 1000*0.00001484 - 0.7613 +
8000*0.00002895 - 0.08784 = -3.2724

1/1+e^-(-3.2724) = 0.0365

Now, this probability makes more sense than the previous result, as we can conclusively state that this should be classified as 0 since it is below 0.5, that is, the cutoff value.

(5) (~40 points) Explore alternative classification methods to improve your predictions.

- In the models below, use a 10-fold cross validation to make the results consistent across.
- Use the same training and test data you created and used after recoding the data in Q4.
- Make all comparisons to the logistic model you have run in Q4 after recoding the data.

a) Set the seed to **123** and run a linear discriminant analysis (LDA) using all variables.

i) Compute the confusion matrix and performance measures for the test data, and compare them **with the logistic regression results**. Discuss your findings.

We have the accuracy for the LDA model as 0.6723 as compared to the 0.67 with the logistic regression results. Furthermore, the sensitivity and specificity of the LDA model is 0.5693 and 0.7727 as compared to the corresponding 0.5854 and 0.7525 with the logistic regression model.

This tells us the logistic model is able to detect bad buys better than the LDA model, but the latter makes fewer false positive errors. However, since sensitivity is of higher importance here, the logistic model is marginally better.

b) Set the seed to **123** and run a kNN model using all variables.

i) Create a plot of the k vs. cross-validation accuracy.

ii) What is the optimal k? What else do you infer from the plot?

Hint: To inspect the details of any model, you will need to train the model and store it before piping it into predict(). See the GitHub repository for guidance.

We can see that as k increases, the accuracy increases as well, reaching its peak when k is 19, after which it starts to decrease again. This means that the optimal k value is 19.

iii) Compute the confusion matrix and performance measures for the test data, and compare them **with the logistic regression and LDA model** results. Discuss your findings.

	Accuracy	Sensitivity	Specificity
Logistic	0.67	0.5854	0.7525
LDA	0.6723	0.5693	0.7727
kNN	0.6285	0.5543	0.7009

The kNN model performs worst compared to the other two. Logistic model is still marginally better.

c) Set the seed to **123** and build a lasso model using all variables.

i) Set the seed to **123** and run a Lasso model using all variables. Report the table of variable importance in a tibble format and share your observations.

Hint: See the Github repo for help. Use a 100-point grid between 10^{-5} and 10^{2}

ii) Report the plot of variable importance for the 25 most important variables.

iii) What is the optimum lambda selected by the model? What does it mean that the algorithm chooses this particular lambda value?
The optimum lambda selected by the model here is 0.0003053856
This means that this particular lambda value gives the minimum error rate on validation data.

iv) Compute the confusion matrix and performance measures for the test data, and compare them **with the logistic regression, LDA, and kNN model** results. Discuss your findings.

	Accuracy	Sensitivity	Specificity
Logistic	0.67	0.5854	0.7525
LDA	0.6723	0.5693	0.7727
kNN	0.6285	0.5543	0.7009
Lasso	0.6694	0.5854	0.7514

The lasso model is an improvement over the previously two created models, however, the logistic model still performs best comparatively.

d) Set the seed to **123** and build a (I) ridge and (II) elastic net[1] model using all variables.

i) Compute the confusion matrix and performance measures for the test data, and compare them **only with the lasso model** results. Discuss your findings.
Hint: Use the same grid for lambda. Notice the different optimum value!

	Accuracy	Sensitivity	Specificity
Lasso	0.6694	0.5854	0.7514
Ridge	0.6711	0.5980	0.7424
Elastic net	0.6688	0.5842	0.7514

Here, the ridge model has the highest sensitivity and accuracy values while the specificity value is not very far behind either. It's clearly the best model of all three.

Also, we have the optimum value of lambda for ridge at 0.0559081 and for elastic, it is at 0.0005857021, which is considerably less.

e) Set the seed to **123** and run a quadratic discriminant analysis (QDA) with all variables

[1] Naive elastic net. Feel free to run a grid search but be careful not to hit the limits of your computational power!

i) Have you received an error? What do you think the error you received means? Do some research and explain what you think it is about.
Yes, there is a rank deficiency error observed in this model, which tells us that a few variables are collinear and one or more covariance matrices cannot be inverted to obtain the estimates

ii) Why is the rank deficiency a problem for QDA, but not for LDA?
LDA assumes equality of covariance among predictor variables and doesn't check for it while QDA doesn't assume equality of covariance among predictor variables and checks for the same.

iii) Compute the confusion matrix and performance measures for the test data, and compare them **only with the LDA model** results. Discuss your findings.

	Accuracy	Sensitivity	Specificity
LDA	0.6723	0.5693	0.7727
QDA	0.6387	0.4405	0.8322

While the QDA model performs significantly better in terms of specificity, it does equally poorly with sensitivity and accuracy, which are more important. The LDA model would be preferred over this.

f) Among all the models you have studied, which model do you think is better for the given business case/problem? Discuss why you think it is better than the others. Also report the ROC curves for the models you have developed on the same chart.
As we have observed, the logistic model performs much better than the rest in terms of sensitivity, which is of the primary concern in this particular business case, seeing that false negatives are more dangerous and would lead to greater loss.
Also, as seen from the ROC graph, LDA, Ridge, Logistic are the top performing models.

Bonus question: You may have noticed that lasso drops certain levels of Make and Color such as "Brown", keeping the other levels of the same variable ("Blue" etc.). This may not be helpful, so you may want to use a grouped lasso. Set the seed to 123 and try grouped lasso with the lambda values 50 and 100. Do the results make more sense now? Why or why not?
Hint: Run a plain lasso again with a lambda value of 0.01 and print the coefficients this time. Compare them with the coefficients from group lasso.

Since we have categorical predictors, which are encoded as multiple dummy variables, it needs to be considered together rather than separately. Lasso leaves certain levels out randomly and it does not add meaningful value to the model, leading to reduced accuracy. This is why for categorical variables, grouped lasso improves the accuracy of the model as it correctly identifies which categorical variable has an effect in predicting the dependent variable and does not randomly drop them.