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 data contains details about cars that have been bought in auctions and sold by dealers. Based on whether it made a profit or not it is classified as a badbuy or not in the data. This is our dependent variable which we are trying to predict so that dealers can make better decisions in future. We have other variables like color, make, age, wheel type, odometer readings and size which are attributes of the car. We also have the type of auction, auction price and retail price of the cars. The data has no missing values while there are rows which say not available where there is no meaningful data.*

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.

➔ *(1) **Auction Prices of cars:** We can see from the box plots that the auction prices of good cars are slightly higher than the auction price of lemons based on mean and median values. But there are very high auction prices for lemons which are outliers. This could be because of lack of verification from the buyer or misrepresentation of the value of the car by seller. This could also be due to some unknown problems with the cars which came to light only after the car was bought in the auction.*

*(2) **Ages of the cars:** Based on the mean and median values we can say that the as the age of the car increases more are the chances of it being a lemon. Thus, older cars have more chances of being lemon. But there is an outlier where the older car which is 9 years old is not a lemon. This could be because the owner took great care of the car, the car was a limited-edition model or the car brought in value because of its age.*

*(3) **Odometer of cars:** The total distance driven (odometer reading) is also related to the car being a lemon. Higher odometer readings are seen in lemons. But there are*

many outliers where though the car was driven for only a few kilometers it turned out to be a lemon. This may be due to neglect and lack of maintenance by the owner.

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?)

➔ *Medium cars have the highest percentage of the total lemons.*

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?

➔ *We cannot say that the dealership should stop purchasing the cars of medium size because they show higher percentage of lemons as they are the most frequently bought cars. Also, among the total medium cars bought the good cars are more than lemons. In terms of percentage by category compact cars have more lemons than medium cars. The percentage is high purely because the total number of medium cars bought is high. A better metric to make this decision would be proportion of lemons among the cars of each category and then compare.*

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.

b) For which dataset is the error smaller? Does this surprise you? Why or why not?

➔ *The error is smaller for the Train dataset. This is not a surprise as the model was trained on this model so the predicted probabilities for this model would have less error than the ones made on test data.*

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?

➔ *There are more false negatives (743) than false positives (408) in the predictions*

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?

➔ *For this problem, it is okay to not bid on a good car because our model falsely says it is a lemon (false positive) as we would make no profit or loss. But if we bid on a car thinking it is a good car based on our prediction, but it is actually a lemon (false* *negative) it will lead to losses. Thus, the false negatives are more critical than false positives. Thus, the metric sensitivity is more important for this model*

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)?

➔ *The baseline accuracy of the model, i.e. if we classify all the cars as not lemons is 50.61% as we have that percentage of good cars in the test dataset. Our model predicts with an accuracy of 0.6731 which is better than the baseline accuracy. Thus, the model performs better than using a random classifier.*

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.

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?

➔ *The probability calculated is -1.4098 which does not make sense as probability should always be between 0 and 1. For classification purpose this will be classified as a 0 as it is less than 0.5 (cut-off) but this is a limitation of the LPM that it can generate negative values for probability.*

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.

➔ *We may receive a rank-deficit error due to many scenarios. Yes, we receive a rank-deficit error in this case because there is insufficient information contained in the data to estimate the model. On using tally() we see that the make has 30 unique types but many of them have less than 10 rows of data. Thus, leading to insufficient data. Also, in terms of color there are two levels which indicate the same status. We can fix this my recoding the variables.*

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.

➔ *The coefficient of age is 2.785 x 10^{-1}. When this value is exponentiated it becomes 1.3211. This means that 1-year increase in the age of the car is associated with an increase in odds of it being a lemon by a factor of 1.3211 (about a 32.11% increase), holding everything else constant*

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

➔ *The coefficient of SizeVAN is -5.982 x 10^{-1}. When this value is exponentiated it becomes 0.5497. This means that the odds of a car of size VAN being a lemon is about 45.03% lower than the odds of a car of size Compact doing so, holding everything else constant*

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.

➔ *We achieve an accuracy of 67% which is better than the baseline accuracy. We achieve a sensitivity of 58.5% and a specificity of 75.2%.*

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?

➔ *The value of probability for this car is 0.0414 which means an approximately 4% chance that the car is a badbuy. Thus, this tells us that the car is probably going to be a good car. This is better than the LPM where we get negative value for probability and thus it makes more sense for us to make inferences.*

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

(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.

Performance Measure	***Logistic***	***LDA***
Accuracy	*0.67*	*0.6723*
Sensitivity	*0.5854*	*0.5693*
Specificity	*0.7525*	*0.7727*
Postivie pred value	*0.6977*	*0.7097*
Negative pred value	*0.6503*	*0.6477*
Balanced Accuracy	*0.6690*	*0.6710*

The accuracy of the LDA model increases slightly but the sensitivity goes down. This is compensated by an improvement in specificity. But in our business case we are looking at specificity as we want to reduce false negatives. Thus, though the accuracy is slightly higher a logistic regression model is better than the LDA. This may be because the variables are not normally distributed in the same way in each of the classes.

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?

➔ *The optimal value of K is 49 when we have a tuneLength of 30. From the graph we can see that for low values of k the model is overfitting the train set thus we see that the accuracy in the predictions is low due to the low variance and high bias. But as we increase k this bias is reduced, and the predictive performance of the model improves, and the accuracy increases. Beyond the optimal value of k the accuracy again starts decreasing as we move higher.*

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.

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.

➔

Performance Measure	***Logistic***	***LDA***	***Knn***
Accuracy	*0.67*	*0.6723*	*0.644*
Sensitivity	*0.5854*	*0.5693*	*0.5296*
Specificity	*0.7525*	*0.7727*	*0.7570*
Postivie pred value	*0.6977*	*0.7097*	*0.6802*
Negative pred value	*0.6503*	*0.6477*	*0.6225*
Balanced Accuracy	*0.6690*	*0.6710*	*0.6433*

We see that the accuracy and sensitivity both the metrics go down. Thus, the logistic or LDA models both perform better than Knn for this data. This may be due to lack of data as Knn needs a lot of data to build complex models and predictions. In our case there are many variables, but each variable has very few samples of its kind thus making it difficult for the Knn model to perform 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}

➔ *We see that the variable wheeltype with value null is of the highest importance, but at the same time other wheeltypes are of very low or negligible importance. This does not make much sense as we have to either use all categories of wheel type or none to make meaningful interpretations. This is a limitation of Lasso model when dealing with categorical variables. Similar pattern is observed in other categorical variables*

   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 for this model is 0.0003053856. The algorithm uses cross validation to compute the error rate on the validation data for each value of lambda. Then it chooses the lambda which gives the smallest error rate. So, in this case it means that this value of lambda has lowest error rate.*

   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.

Performance Measure	***Logistic***	***LDA***	***Knn***	***Lasso***
Accuracy	*0.67*	*0.6723*	*0.644*	*0.6694*
Sensitivity	*0.5854*	*0.5693*	*0.5296*	*0.5854*
Specificity	*0.7525*	*0.7727*	*0.7570*	*0.7514*
Postivie pred value	*0.6977*	*0.7097*	*0.6802*	*0.6968*
Negative pred value	*0.6503*	*0.6477*	*0.6225*	*0.6500*
Balanced Accuracy	*0.6690*	*0.6710*	*0.6433*	*0.6684*

The Lasso model performs almost as good as the logistic model in terms of accuracy and sensitivity, but due to the mismatch in dropping categorical variables the interpretation of the results may not be really meaningful with this model. A grouped Lasso may perform better.

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!

➔ *The lambda value for ridge is 0.0559081*
The lambda value for elastic net is 0.0005857021

Performance Measure	***Logistic***	***LDA***	***Knn***	***Lasso***	***Ridge***	***Elastic Net***
Accuracy	*0.67*	*0.6723*	*0.644*	*0.6694*	*0.6711*	*0.6688*
Sensitivity	*0.5854*	*0.5693*	*0.5296*	*0.5854*	*0.5980*	*0.5842*
Specificity	*0.7525*	*0.7727*	*0.7570*	*0.7514*	*0.7424*	*0.7514*
Postivie pred value	*0.6977*	*0.7097*	*0.6802*	*0.6968*	*0.6938*	*0.6964*
Negative pred value	*0.6503*	*0.6477*	*0.6225*	*0.6500*	*0.6543*	*0.6494*
Balanced Accuracy	*0.6690*	*0.6710*	*0.6433*	*0.6684*	*0.6702*	*0.6678*

The Ridge and the elastic net models perform at par with Lasso. They have similar accuracy and sensitivity. We can pick either Ridge or elastic net over Lasso in this case as these models do not drop any variables and thus will not lead to the mismatch in categorical variables.

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

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, we receive an error "model fit failed for Fold03: parameter=none Error in qda.default (x, grouping, ...): rank deficiency in group 0. There were missing values in resampled performance measures". This error means that when we were doing cross validations some or one of the predictors showed no variance. QDA assumes real values instead of factors in the independent variables. Thus, the error indicates that there is rank deficiency i.e. some variables are collinear i.e. there is a strong correlation between two variables, and one or more covariance matrices cannot be inverted to obtain estimates*

ii) Why is the rank deficiency a problem for QDA, but not for LDA?

➔ *LDA is based on the assumption that the classes have equal covariance. Thus, in case the classes do not have equal covariances, the classes with large variances are chosen. But in case of QDA it does not choose the class with large variances and thus we get the rank deficient error.*

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

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

➔ On comparison with LDA the QDA model performs worse in terms of accuracy and especially in terms of sensitivity. This could be because the predictors are linearly distributed across the 2 classes. Thus, the LDA performs better.

Performance Measure	***QDA***	***LDA***
Accuracy	*0.6387*	*0.6723*
Sensitivity	*0.4405*	*0.5693*
Specificity	*0.8322*	*0.7727*
Postivie pred value	*0.7192*	*0.7097*
Negative pred value	*0.6038*	*0.6477*
Balanced Accuracy	*0.6363*	*0.6710*

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.

➔ *Based on the area under the curve metric all the models except Knn and QDA perform equally good. But we can see that the Elastic Net model is the best one with highest Area under the curve value. Based on the performance measures in the confusion matrix, we can say that Ridge is the model with the highest sensitivity and given our requirement to avoid false negatives as they can be costly, we can choose this model over the others. Thus, we could choose the ridge model as it has better sensitivity and performs equally good on the AUC metric.*

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.

➔ *On running the grouped lasso model, we can see that the model either drops or keeps all the variables for each category. This makes more sense as interpretation of these individual variables are dependent on all the variables and dropping a few of them makes their interpretation irrelevant.*
Also, we see that the model removes more categorical variables when lambda is 100 than when it is 50 as an increase in lambda leads to higher penalties and quicker reduction of coefficients to zero.

The different between Lasso and Grouped Lasso is that group lasso handles categorical variables of same type together and this can be seen from the output coefficients of both the models.