Oscar Hernandez
MSDS 411
Fall 2018
Unit 2 – Insurance Claims

Contents

Bingo Bonus

Creative: To calculate the P_TARGET_AMT, I used a stepwise variable selection process to generate my multiple linear regression model. I used BLUEBOOK as the dependent variable, therefore my MLR model made a prediction for BLUEBOOK. I used this prediction along with the P_TARGET_FLAG to arrive at P_TARGET_AMT. This is outlined in the Code section.

Introduction

The purpose of this report is to cover all the analysis that was completed for the Auto Insurance data set. The Auto Insurance data set includes a variety of variables that an insurance company could utilize to predict the probability of a customer getting into a car crash and the amount that it would cost. Specifically, the data set includes 23 variables such as the age of the customer, the distance the customer travels to work, and the type of job they have. The data set includes a mix of categorical and numerical variables. Furthermore, the data set contains two response variables – TARGET_FLAG (a binary response variable, where 1 = customer did get into a crash and 0 = customer did not get into a crash) and TARGET_AMT which contains continuous, numerical values that represent the cost if the customer did get into a crash.
This report will cover the Exploratory Data Analysis that was conducted to better understand the data, how the data was prepared, the creation of three different logistic regression models (along with one multiple linear regression model) and how the "best" logistic regression model was determined. This analysis was done with the primary goal of developing a logistic regression model that can accurately predict the probability of a customer getting into a car crash. The secondary goal is to predict the cost associated with a car crash which will done by utilizing a multiple linear regression model.

Section 1 – Exploratory Data Analysis

Prior to building any model, it is prudent to better understand the data that will be used to train the model. Exploring data is important because we are looking to see the relationship amongst the variables included in our data, check to see if we have erroneous values and understand why we have missing values. As mentioned, our primary goal is to build a logistic regression model that can predict the probability of a customer getting into a car crash based on a variety of variables. Therefore, most of this report will be focused on how the logistic regression models were developed.

As mentioned, this data set contains both numerical and categorical variables. So, it's important to get an early understanding of how the variable are split up. Figure 1 is a table that displays all 23 predictor variables that are includes in our data set.

Variable	Type	Variable	Categorical
AGE	Continuous	KIDSDRIV	Continuous
BLUEBOOK	Categorical	MSTATUS	Categorical
CAR_AGE	Continuous	MVR_PTS	Continuous
CAR_TYPE	Categorical	OLDCLAIM	Categorical
CAR_USE	Categorical	PARENT1	Categorical
CLM_FREQ	Continuous	RED_CAR	Categorical
EDUCATION	Categorical	REVOKED	Categorical
HOMEKIDS	Continuous	SEX	Categorical
HOME_VAL	Categorical	TIF	Continuous
INCOME	Categorical	TRAVTIME	Continuous
JOB	Categorical	URBANICITY	Categorical
		YOJ	Continuous

Figure 1: Table of Data Set Variables

We can see from the table that 9 of the variables are continuous and 14 are categorical. However, this does not mean that the variables will stay as such to train the model. It may be prudent to transform some of these variables depending on their distribution and if the transformation provides better insight to the problem at hand. For example, the INCOME variable is initially provided as a categorical variable but it might make more sense to transform it to a continuous variable it could lead to a better model.

The next part of our analysis will include a review of all the continuous variables that are in our data set. Specifically, we want to check the distribution of these variables to see if there are some clearly erroneous values and see how much of this data subset has missing values. Figure 2 gives a robust overview of all the continuous variables.
The table suggests that we will have to come up with a way to impute for missing values since three of the continuous variables have missing values. CAR_AGE and YOJ have a significant amount of missing values. Also, the table shows us that many of these continuous variables have a high proportion of 0 values which may indicate reason for transformation into a binary or factor variable. Lastly, there doesn't appear to be any egregiously incorrect values except the -3 value for CAR_AGE. It is possible that a driver could be 81 years old, have 5 kids and be with an insurance provider for 25 years. Overall, Figure 2 provides us an idea of which variables may need to be inputted.

Variable	Missing	Min	Q1	Median	Mean	Q3	Max
AGE	6	16	39	45	44.79	51	81
CAR_AGE	510	-3	1	8	8.328	12	28
CLM_FREQ	0	0	0	0	0.7986	2	5
HOMEKIDS	0	0	0	0	0.7212	1	5
KIDSDRIV	0	0	0	0	0.1711	0	4
MVR_PTS	0	0	0	1	1.696	3	13
TIF	0	1	1	4	5.351	7	25
TRAVTIME	0	5	22	33	33.49	44	142
YOJ	454	0	9	11	10.5	13	23

Figure 2: Descriptive Statistics for Continuous Variables

Furthermore, it's important that we also review the categorical variables. After reviewing the data set, the following items were of particular interest:

- JOB has 526 missing values which represents 6.4% of the total values
- INCOME and HOME_VAL contain missing values
- The majority of the customers were not single parents – roughly 85%
- Approximately 85% of the customers had at least a high school education
- Approximately 88% of the customers never had their license revoked

As such, we can now conclude that perhaps we need to impute six of the variables in our data set to account for missing values. Another option is to create a missing value category for our categorical variables or create a binary variable where the 0 value is the non-existence of the characteristic.

Moreover, a visual inspection of our variables can give us further insight into how they are distributed. Figure 3 shows three histograms of the variables HOME_VAL, OLDCLAIM and CAR_AGE, respectively.

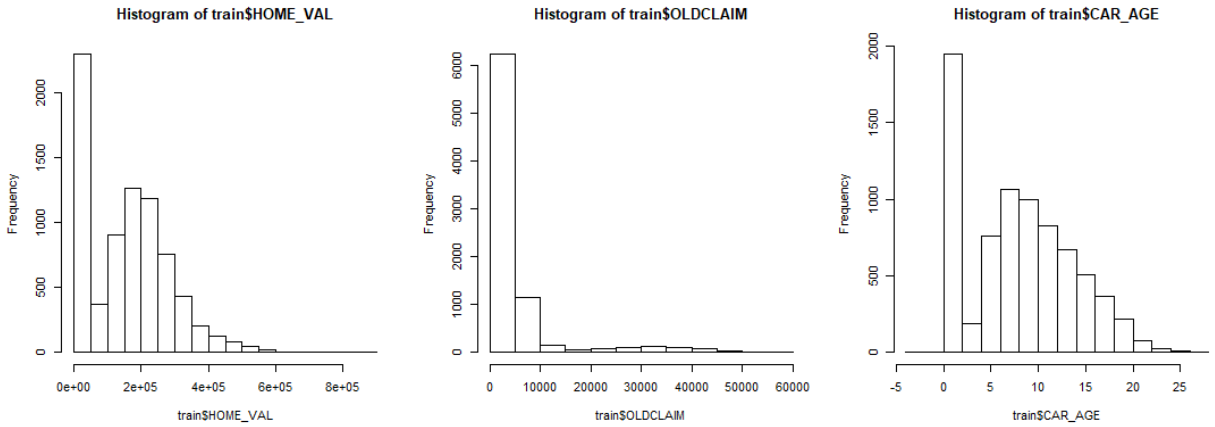

Figure 3: Histogram of HOME_VAL, OLDCLAIM, CAR_AGE

Figure 3 shows us the distribution of these variables and we can immediately see the high frequency of zero values. A log transformation of these variables will not be directly possible given so many zero values. Specifically, these plots demonstrate what's called a zero-inflated distribution. If the primary purpose of this analysis was to develop a multiple linear regression model, this would pose a problem. However, for a logistic regression model it is not a huge concern.

As mentioned initially, our secondary goal is to develop a multiple linear regression model that predicts TARGET_AMT. Therefore, the last step of our Exploratory Data Analysis is to check a correlation plot to see if there exists any correlation between some of our predictor variables and TARGET_AMT. Correlation variables are of interest given that they may add unnecessary complexity to our model and the inclusion of them may result in multicollinearity. The absence (or at least limited amount) of multicollinearity is an important assumption of any multiple linear regression model.

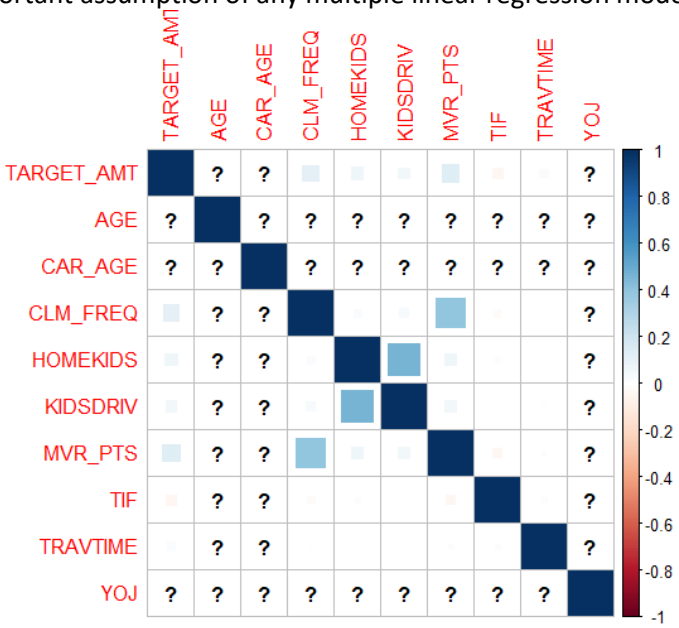

Figure 4: Correlation Plot

From Figure 4, we can see several predictor variables that are correlated amongst themselves. Notably, CLM_FREQ and MVR_PTS appear to be correlated. This makes sense given that we may expect a customer that has a lot of claims in the past, would probably have a lot of motor vehicle points. HOMEKIDS and KIDSDRIV appear to be correlated as well. This also shouldn't be surprising given that if a customer has kids at home there is a chance that those kids probably drive, depending on their age. Lastly, there are indications that TARGET_AMT and some of the predictor variables appear to be correlated although not strongly.

Model Preparation

The Model Preparation section will cover what was done on the training data set prior to fitting any models. As noted, the two main issues that we discovered during our EDA, is that the data set had missing values and the need to perhaps transform some variables.

The first part of the data preparation is to determine which variables would need stay as continuous/categorical or be converted. The first item that was necessary was converting TARGET_FLAG into a binary variable where 1 would indicate a car crash and 0 would mean there was no car crash. BLUEBOOK, HOME_VAL, INCOME, OLDCLAIM were converted to numerical variables since based on our EDA, we feel they are likely to contribute more meaningfully to our model as such.

The next step in our data preparation was to impute the missing values. A popular R package called "MICE" was used to impute the missing values as opposed to simply using mean or median. For the categorical variables, the use of mean/median wouldn't even make sense given their structure which further added to the decision to use the MICE package. The mice package uses an algorithm to replace the missing values with plausible data values. According to a popular mice tutorial, "these plausible values are drawn from a distribution specifically designed for each missing datapoint." Specifically, we used the RandomForest method within the MICE package to impute the missing values. This method was chosen because it can account for missing numerical and categorical values which was necessary for our data set.

After the missing values were imputed, the last data preparation step includes variable transformation. Based on our EDA, the decision was made to do a square root transformation on the following variables: INCOME, TRAVTIME, OLDCLAIM, BLUEBOOK. This transformation was done to change the distribution of these variables. Furthermore, we noticed that the TARGE_AMT variable had an odd distribution. Therefore, we completed a natural log + 1 transformation of the variable. The "+1" was done to account for the numerous 0 values.

The last transformations that were completed was changing certain variables into categorical variables. MVR_PTS was changed to a binary variable where the 0 value indicates that the driver had 0 points and 1 indicates that the driver has at least 1 point on their record. Based on the distribution of MVR_PTS, we felt it was best to make this transformation. The same logic was applied to transforming KIDSDRIV and HOMEKIDS, where a value of 0 would mean the customer has no kids driving and no kids at home, respectively.

Build Model

This section of the report will cover the three logistic regression models that were built and a discussion of some of their key aspects. Figure 5, 6 and 7 show the model summary data for the three models that were built – "Stepwise Model", "User-Defined Model" and "Categorical Model."

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.82	0.27	-6.686	0.00
URBANICITYUrban	2.40	0.11	21.44	< 2e-16
JOBDoctor	-0.76	0.29	-2.675	0.01
JOBHome Maker	-0.36	0.15	-2.418	0.02
JOBLawyer	-0.23	0.19	-1.221	0.22
JOBManager	-0.91	0.14	-6.295	0.00
JOBMissing_Job	-0.34	0.20	-1.727	0.08
JOBProfessional	-0.17	0.12	-1.364	0.17
JOBStudent	-0.36	0.14	-2.658	0.01
JOBz_Blue Collar	-0.03	0.11	-0.285	0.78
HOME_VAL	0.00	0.00	-3.48	0.00
CAR_TYPEPanel Truck	0.59	0.15	4.036	0.00
CAR_TYPEPickup	0.55	0.10	5.466	0.00
CAR_TYPESports Car	0.95	0.11	8.865	< 2e-16
CAR_TYPEVan	0.68	0.12	5.57	0.00
CAR_TYPEz_SUV	0.72	0.09	8.415	< 2e-16
CLM_FREQ	0.19	0.02	7.647	0.00
REVOKEDYes	0.74	0.08	9.158	< 2e-16
PARENT1Yes	0.25	0.12	2.078	0.04
CAR_USEPrivate	-0.78	0.09	-8.488	< 2e-16
TRAVTIME	0.17	0.02	8.189	0.00
TIF	-0.06	0.01	-7.657	0.00
KIDSDRIVz_Kids_NoDrive	-0.59	0.10	-6.121	0.00
BLUEBOOK	-0.01	0.00	-5.173	0.00
MVR_PTSZero_Points	-0.34	0.06	-5.577	0.00
EDUCATIONBachelors	-0.26	0.12	-2.261	0.02
EDUCATIONMasters	-0.10	0.18	-0.582	0.56
EDUCATIONPhD	0.01	0.21	0.041	0.97
EDUCATIONz_High School	0.06	0.09	0.673	0.50
MSTATUSz_No	0.57	0.09	6.504	0.00
INCOME	0.00	0.00	-4.784	0.00
HOMEKIDSz_No_Kids_Home	-0.23	0.09	-2.647	0.01
CAR_AGE	-0.01	0.01	-1.664	0.10

Figure 5: Stepwise Model

The Stepwise Model was calculated using the AIC statistic in Stepwise fashion. Figure 5 provides us a view of all the variables, their respective coefficients and statistical significance. We can see that the majority if the variables are statistically significant based on an alpha of 0.05. Notably, it appears that the majority of the education variables are not statistically significant. This provides some proof that education level doesn't have a meaningful impact on the probability of getting into an accident. Our model also confirms the idea that people living in urban areas have a higher chance of getting into a car accident. The URBANICITYUrban coefficient is the largest in this model. Another interesting takeaway is with the BLUEBOOK coefficient. The negative coefficient indicates that the higher the value of a customer's car the decreased probability of getting into car accident, which makes sense given that people who have more expensive cars probably drive safer.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.92	0.22	-8.877	< 2e-16
AGE	-0.02	0.00	-4.83	0.00
CAR_USEPrivate	-0.79	0.06	-13.69	< 2e-16
CLM_FREQ	0.20	0.02	8.554	< 2e-16
HOME_VAL	0.00	0.00	-3.674	0.00
INCOME	0.00	0.00	-12.239	< 2e-16
MSTATUSz_No	0.52	0.07	7.524	0.00
MVR_PTSZero_Points	-0.40	0.06	-6.683	0.00
REVOKEDYes	0.79	0.08	10.279	< 2e-16
TRAVTIME	0.16	0.02	8.035	0.00
URBANICITYUrban	2.14	0.11	19.554	< 2e-16

Figure 6: User-Defined Model

The User-Defined Model was determined manually as opposed to using a systematic fashion such as Stepwise. There was no specific rule used to manually develop this model. This model was created purely on our assumptions of what variables we believe would accurately predict the probability of a car accident.

Figure 6 provides us a view of all the variables, their respective coefficients and statistical significance. We can see that every single variable has a statistically significant coefficient. This is a bit surprising (but welcomed) given that the model was manually created. Specifically, the MVR_PTSZero_Points negative coefficient makes sense given that we believe someone without any points on their driving record would probably not get into a car crash. Our model confirms that idea. Once again, UrbanicityUrban has the largest coefficient. Furthermore, we see proof that older drivers may be more likely to get into a car accident as indicated by the negative AGE coefficient. Lastly, after exponentiating the REVOKEDYes coffecieint (0.79) we see that customers that have had their license revoked have 2.2 higher odds of getting into a car accident compared to customers that haven't had their license revoked. This also makes sense given that someone whose had a revoked license probably isn't a safe driver.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.07	0.20	-10.5	< 2e-16
CAR_TYPEPanel Truck	0.18	0.14	1.339	0.18
CAR_TYPEPickup	0.59	0.10	6.055	0.00
CAR_TYPESports Car	1.28	0.12	10.815	< 2e-16
CAR_TYPEVan	0.43	0.12	3.656	0.00
CAR_TYPEz_SUV	1.00	0.10	9.904	< 2e-16
CAR_USEPrivate	-0.78	0.09	-8.719	< 2e-16
EDUCATIONBachelors	-0.51	0.10	-4.866	0.00
EDUCATIONMasters	-0.45	0.16	-2.917	0.00
EDUCATIONPhD	-0.54	0.19	-2.886	0.00
EDUCATIONz_High School	-0.04	0.09	-0.459	0.65
HOMEKIDSz_No_Kids_Home	-0.27	0.09	-3.067	0.00
JOBDoctor	-0.97	0.28	-3.523	0.00
JOBHome Maker	0.14	0.13	1.04	0.30
JOBLawyer	-0.48	0.18	-2.649	0.01
JOBManager	-1.15	0.14	-8.312	< 2e-16
JOBMissing_Job	-0.61	0.19	-3.262	0.00
JOBProfessional	-0.38	0.12	-3.153	0.00
JOBStudent	0.15	0.12	1.246	0.21
JOBz_Blue Collar	-0.19	0.10	-1.878	0.06
KIDSDRIVz_Kids_NoDrive	-0.57	0.09	-6.073	0.00
MSTATUSz_No	0.72	0.07	9.774	< 2e-16
MVR_PTSZero_Points	-0.49	0.06	-8.336	< 2e-16
PARENT1Yes	0.20	0.12	1.711	0.09
RED_CAR1	0.03	0.08	0.303	0.76
REVOKEDYes	0.74	0.08	9.444	< 2e-16
SEXz_F	-0.31	0.10	-3.065	0.00
URBANICITYUrban	2.37	0.11	21.944	< 2e-16

Figure 7: Categorical Model

The Categorical Model was the last model that was created for this analysis. This model was created from using only categorical variables. After completing the data preparation step, we noticed that the majority of our variables were of the categorical type so the decision was made to create a model that only contained categorical variables.

Figure 7 provides us a view of all the variables, their respective coefficients and statistical significance. We can see that the majority of the variables have a statistically significant coefficient. This is also welcoming news. However, this model did produce variables that are not statistically significant (based on Alpha = 0.05). Notably, JobHome Maker, JobStudent and RED_CAR1 had coefficients that were not statistically significant. The RED_CAR1 variable is interesting because it provides some evidence that can refute the myth that red cars tend to get into more car accidents. Furthermore, we see for the last time that UrbanicityUrban has the largest coefficient. Another interesting aspect of this model is that this

contains multiple coefficients that are at least 1 or -1 which was not as abundant in our other models. Specifically, the CAR_TypeSports Car variable makes sense and has the second largest coefficient. We would expect that people who drive sports cars are more likely to get into an accident. Our model suggests that this idea is possibly correct.

Select Models

The final section of this report will review which model was determined to be our "best" model. In order to make such a validation, we calculated multiple metrics that are commonly used to compare models using training data along a ROC curve for each model. Figure 8 display the metrics that were calculated with respect to each model.

	AIC	Rank	BIC	Rank	-2*Log Likelihood	Rank	KS	Rank
Stepwise Model	7384.45	1st	7615.69	1st	7318.45	1st	0.4687	1st
User-Defined Model	7805.24	3rd	7882.32	3rd	7783.24	3rd	0.4025	3rd
Categorical Model	7658.84	2nd	7855.06	2nd	7602.84	2nd	0.4422	2nd

Figure 8: Model Comparison (Train Data)

All of the metrics used are considered more accurate the lower they are when compared to each other. Figure 8 ranks where each model lies with regards to each metric. The data suggests that the Stepwise Model is the most accurate model given than it ranked 1st in every single category. However, choosing it as the "best" model cannot solely be based on this table. When reviewing all our models, the coefficients made reasonable sense. We did not see any unexpected issues with the coefficients. The reason this is brought up is because if the Stepwise model has coefficients that were contrary to what we would expect would lead to better prediction of crashing, we could have gone with a different model as the "best" one. The User-Defined model was still a top contender because of its simplicity. However, it ranked last on every validation metric we used.

Prior to selecting it as the "best" model it's prudent to inspect each models ROC curve and its respective AUC. Figure 9 displays the Stepwise, User-Defined and Categorical Model from left to right. All three models showed evidence that they are useful in predicting the probability of getting into car crash. None of the models were perfect but more importantly, none of the models showed poor signs of being able to predict probability based on the line curvature.

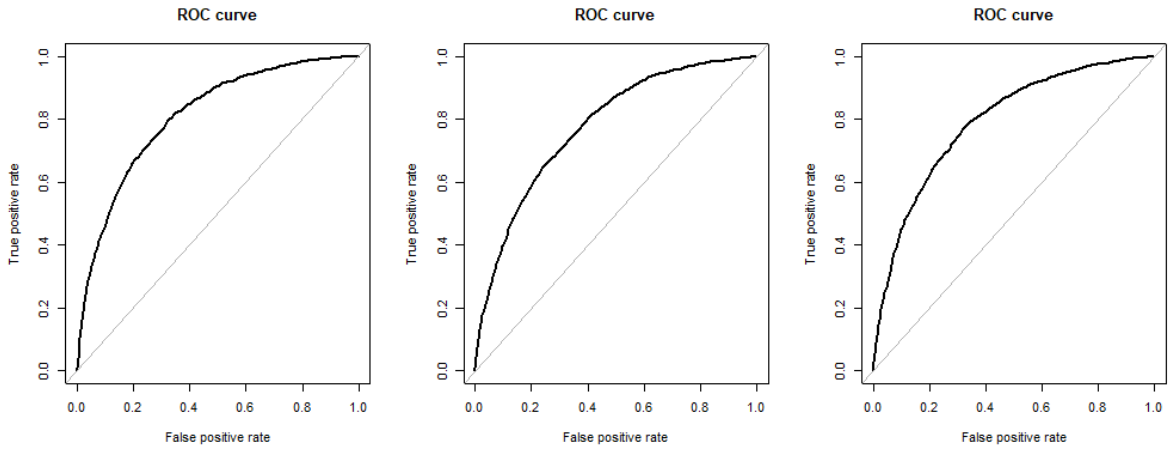

Figure 9: ROC Curves

The AUCs of each model were 0.812, 0.778 and 0.794, respectively. A higher AUC score means the model was better at predicting a car crash probability. Therefore, based on this final validation test, we will select the Stepwise model as the "best" model.

Conclusion

The purpose of this report was to explain the analysis and logic behind developing logistic regression models to predict the probability of a customer getting into a car crash based on a variety of variables. A secondary goal was to determine the amount that it would cost depending if the customer did get into a car crash. This secondary effort is outputted in our Scoring Program and not discussed at length in this report since it was not the primary focus.

The Exploratory Data Analysis section was the initial groundwork that allowed us to first notice some of the subsequent work that would have to be done with regards to imputing values and getting an idea of which variables may need to get transformed. The subsequent parts explained the logic in preparing the training data, how each model was built, which variables should be categorical/continuous and ultimately what criteria was used to select the model that we believe will do the best job at predicting the probability of getting into a car crash.

From this exercise, it appears that it's better to include a mix of categorical and continuous variables. Furthermore, the coefficients in our "best" model reaffirmed in what we initially believe would be good predictors of a car crash. The CAR_TYPE variable was significant in predicting a car crash as well as the JOB type of a customer. People that drive sports cars are more likely to get into a car crash and people in managerial positions are less likely. Lastly, it was interesting to see that education level wasn't a significant predictor of a car crash. Overall, our "best" model seems to be valuable in predicting the probability of a car crash.

Code

```
############ Insurance Claims Unit 2 - Oscar Hernandez
################################################
####Includes Stand Alone Scoring Program#####

#####Import the necessary packages
library(mice)
library(MASS)
library(gmodels)
library(forcats)
library(ROSE)
library(InformationValue)
library(corrplot)

#####

##### Set working directory and load test data
setwd("C:/Users/herna_000/Desktop")
train_original <- read.csv("logit_insurance.csv", header=TRUE)
train <- train_original

#####

############ PART 1: DATA EXPLORATION
#############################################################
str(train_original)
summary(train_original)
summary(train)

hist(train$TARGET_AMT)  #a lot of zero values; right, positive skew
boxplot(train$TARGET_AMT)

CrossTable(train$AGE)
hist(train$AGE) #looks normal
boxplot(train$AGE) #outliers present

hist(train$BLUEBOOK) #positive, right skew
boxplot(train$BLUEBOOK) #outliers present

hist(train$CAR_AGE) #normal but with zero-inflation shape
boxplot(train$CAR_AGE)

hist(train$CLM_FREQ)  # majority of zero values
```

```
boxplot(train$CLM_FREQ)
CrossTable(train$CLM_FREQ)

hist(train$HOMEKIDS) #majority of zero values
boxplot(train$HOMEKIDS)
CrossTable(train$HOMEKIDS)

par(mfrow=c(1,3))
hist(train$HOME_VAL) #normal with zero-inflation shape
hist(train$OLDCLAIM)
hist(train$CAR_AGE)
par(mfrow=c(1,1))

hist(train$INCOME) #heavy right, positive skew


boxplot(train$HOME_VAL)
boxplot(train$INCOME) #outliers present

hist(train$KIDSDRIV) #majority of zero values
boxplot(train$KIDSDRIV)

hist(train$MVR_PTS)  #majority of zero values
boxplot(train$MVR_PTS)
CrossTable(train$MVR_PTS)

#signficant amount of outliers; majority of zero values or less than $10000
boxplot(train$OLDCLAIM)

hist(train$TIF) #lot of zero values
boxplot(train$TIF)
CrossTable(train$TIF)

hist(train$TRAVTIME) #normal with some outliers
boxplot(train$TRAVTIME)

hist(train$YOJ) #normal with zero-inlation shape
boxplot(train$YOJ)

CrossTable(train$JOB)


numeric <- subset(train_original, select = c(TARGET_AMT, AGE, CAR_AGE, CLM_FREQ, HOMEKIDS,
KIDSDRIV, MVR_PTS, TIF, TRAVTIME, YOJ), na.rm = TRUE)
c <- cor(numeric)
corrplot(c, method = "square")
```

```
############ PART 2: DATA PREPARATION
############################################################

index_column <- data.frame(train$INDEX)

#Make sure that each variable is treated as continuous or categorical
train$TARGET_FLAG <- as.factor(train$TARGET_FLAG)
train$BLUEBOOK <- suppressWarnings(as.numeric(gsub("[^0-9.]", "", train$BLUEBOOK)))
train$CAR_TYPE <- as.factor(train$CAR_TYPE)
train$CAR_USE <- as.factor(train$CAR_USE)
train$EDUCATION <- as.factor(train$EDUCATION)
train$HOME_VAL <- suppressWarnings(as.numeric(gsub("[^0-9.]", "", train$HOME_VAL)))
train$INCOME <- suppressWarnings(as.numeric(gsub("[^0-9.]", "", train$INCOME)))
train$JOB <- as.factor(train$JOB)
train$MSTATUS <- as.factor(train$MSTATUS)
train$OLDCLAIM <- suppressWarnings(as.numeric(gsub("[^0-9.]", "", train$OLDCLAIM)))
train$PARENT1 <- as.factor(train$PARENT1)
train$RED_CAR <- as.factor(ifelse(train$RED_CAR=="yes", 1, 0))
train$REVOKED <- as.factor(train$REVOKED)
train$SEX <- as.factor(train$SEX)
train$URBANICITY <- ifelse(train$URBANICITY == "Highly Urban/ Urban", "Urban", "Rural")
train$URBANICITY <- as.factor(train$URBANICITY)

#Imput missing values using MICE package
train_temp1 <- mice(train[, !names(train) %in% "INDEX"], m =5, maxit=10, meth = "rf", seed = 500)
train_imp1 <-complete(train_temp1)  #restart here if needed
train_imp1 <- cbind(index_column, train_imp1)
colnames(train_imp1)[colnames(train_imp1)=="train.INDEX"] <- "INDEX"

#Make changes to the variables in the dataset

train_imp1$JOB <- as.character(train_imp1$JOB)
train_imp1$JOB[train_imp1$JOB==""] <- "Missing_Job"
train_imp1$JOB <- as.factor(train_imp1$JOB)

train_imp1$CAR_AGE[train_imp1$CAR_AGE <0 ] <- 0

train_imp1$INCOME <- sqrt(train_imp1$INCOME) #transformed income to its square root
train_imp1$TRAVTIME <- sqrt(train_imp1$TRAVTIME) #transformed travel time to its square root
train_imp1$OLDCLAIM <-sqrt(train_imp1$OLDCLAIM) #transformed oldclaim to its natural log
train_imp1$BLUEBOOK <-sqrt(train_imp1$BLUEBOOK) #transformed bluebook to its square root
train_imp1$TARGET_AMT <- log(train_imp1$TARGET_AMT+1) #transformed target amount to its
natural log

#Creating factor variable to show if record has or doesn't have points
train_imp1$MVR_PTS <-ifelse(train_imp1$MVR_PTS == 0, "Zero_Points", "Has_Points")
train_imp1$MVR_PTS <-as.factor(train_imp1$MVR_PTS)
```

```
#Creating factor variable to show if kids drive or don't drive
train_imp1$KIDSDRIV <-ifelse(train_imp1$KIDSDRIV == 0, "z_Kids_NoDrive", "Kids_Drive")
train_imp1$KIDSDRIV <-as.factor(train_imp1$KIDSDRIV)

#Creating factor variable to show if there are kids at home or not
train_imp1$HOMEKIDS <-ifelse(train_imp1$HOMEKIDS == 0, "z_No_Kids_Home", "Kids_Home")
train_imp1$HOMEKIDS <-as.factor(train_imp1$HOMEKIDS)

summary(train_imp1)


############ PART 3: BUILD MODELS
#####################################################################

#Building one model using Stepwise variable selection
#Need to specifiy the upper model and lower models
# Define the upper model as the FULL model
upper.glm <- glm(TARGET_FLAG ~ .,data=train_imp1[, !names(train_imp1) %in% c("INDEX",
"TARGET_AMT")], family = binomial())

# Define the lower model as the Intercept model
lower.glm <- glm(TARGET_FLAG ~ 1,data=train_imp1[, !names(train_imp1) %in% c("INDEX",
"TARGET_AMT")], family = binomial())

# Need a simple generalized linear regression model to initialize stepwise selection
initialize_stepwise.glm <- glm(TARGET_FLAG ~ AGE,data=train_imp1[, !names(train_imp1) %in%
c("INDEX", "TARGET_AMT")], family = binomial())



#Create stepwise.glm
stepwise.glm <- stepAIC(object=initialize_stepwise.glm,scope=list(upper=formula(upper.glm),lower=~1),
direction=c('both'))
summary(stepwise.glm)
train_imp1$StepwisePrediction <- predict(stepwise.glm, type = "response")

#Building second model that is user-defined
user_defined.glm <- glm(TARGET_FLAG ~ AGE +
CAR_USE+CLM_FREQ+HOME_VAL+INCOME+MSTATUS+MVR_PTS+REVOKED+TRAVTIME+URBANICITY,d
ata=train_imp1[, !names(train_imp1) %in% c("INDEX", "TARGET_AMT")], family = binomial())
summary(user_defined.glm)
train_imp1$UserPrediction <- predict(user_defined.glm, type = "response")

#Building third model with just numeric variables
numeric_model.glm <- glm(TARGET_FLAG ~
AGE+BLUEBOOK+CAR_AGE+CLM_FREQ+HOME_VAL+INCOME+OLDCLAIM+TIF+TRAVTIME+YOJ,data=trai
n_imp1[, !names(train_imp1) %in% c("INDEX", "TARGET_AMT")], family = binomial())
summary(numeric_model.glm)
```

```
train_imp1$NumericPrediction <- predict(numeric_model.glm, type = "response")

#Building fourth model with just categorical variables
categorical_model.glm <- glm(TARGET_FLAG ~
CAR_TYPE+CAR_USE+EDUCATION+HOMEKIDS+JOB+KIDSDRIV+MSTATUS+MVR_PTS+PARENT1+RED_CAR
+REVOKED+SEX+URBANICITY,data=train_imp1[, !names(train_imp1) %in% c("INDEX", "TARGET_AMT")],
family = binomial())
summary(categorical_model.glm)
train_imp1$CategoricalPrediction <- predict(categorical_model.glm, type = "response")
############ PART 4: SELECT MODELS
#####################################################################

AIC(stepwise.glm)
BIC(stepwise.glm)
AIC(user_defined.glm)
BIC(user_defined.glm)
AIC(numeric_model.glm)
BIC(numeric_model.glm)
AIC(categorical_model.glm)
BIC(categorical_model.glm)

print(-2*logLik(stepwise.glm, REML = TRUE))
print(-2*logLik(user_defined.glm, REML = TRUE))
print(-2*logLik(numeric_model.glm, REML = TRUE))
print(-2*logLik(categorical_model.glm, REML = TRUE))

library(InformationValue)
ks_stat(actuals=train_imp1$TARGET_FLAG, predictedScores=train_imp1$StepwisePrediction)
ks_stat(actuals=train_imp1$TARGET_FLAG, predictedScores=train_imp1$UserPrediction)
ks_stat(actuals=train_imp1$TARGET_FLAG, predictedScores=train_imp1$NumericPrediction)
ks_stat(actuals=train_imp1$TARGET_FLAG, predictedScores=train_imp1$CategoricalPrediction)

par(mfrow=c(1,3))
roc.curve(train_imp1$TARGET_FLAG, train_imp1$StepwisePrediction)
roc.curve(train_imp1$TARGET_FLAG, train_imp1$UserPrediction)
roc.curve(train_imp1$TARGET_FLAG, train_imp1$CategoricalPrediction)
par(mfrow=c(1,1))

roc.curve(train_imp1$TARGET_FLAG, train_imp1$NumericPrediction)
summary(stepwise.glm)
```