IS621 Homework 4

Sekhar Mekala, John DeBlase, Sonya Hong

Friday, November 11, 2016

Project requirements

The main goal of this project is to perform data analysis of an insurance company's data in order to predict if a person will be in a car crash, and the amount it will cost to the company if the person does crash his/her car. We are given 2 data sets: *training* and *test* data sets. The training data has input variables along with the observed response variable. We will use the training data set to train our model, and the predictions obtained on the test data will be submitted as a project deliverable.

Data Exploration

The training and test data sets have the following variables:

Figure-1: Training and test data sets variables

VARIABLE NAME	DEFINITION	THEORETICAL EFFECT
INDEX	Identification Variable (do not use)	None
TARGET_FLAG	Was Car in a crash? 1=YES 0=NO	None
TARGET_AMT	If car was in a crash, what was the cost	None
AGE	Age of Driver	Very young people tend to be risky. Maybe very old people also.
BLUEBOOK	Value of Vehicle	Unknown effect on probability of collision, but probably effect the payout if there is a crash
CAR_AGE	Vehicle Age	Unknown effect on probability of collision, but probably effect the payout if there is a crash
CAR_TYPE	Type of Car	Unknown effect on probability of collision, but probably effect the payout if there is a crash
CAR_USE	Vehicle Use	Commercial vehicles are driven more, so might increase probability of collision
CLM_FREQ	# Claims (Past 5 Years)	The more claims you filed in the past, the more you are likely to file in the future
EDUCATION	Max Education Level	Unknown effect, but in theory more educated people tend to drive more safely
HOMEKIDS	# Children at Home	Unknown effect
HOME_VAL	Home Value	In theory, home owners tend to drive more responsibly
INCOME	Income	In theory, rich people tend to get into fewer crashes
JOB	Job Category	In theory, white collar jobs tend to be safer
KIDSDRIV	# Driving Children	When teenagers drive your car, you are more likely to get into crashes
MSTATUS	Marital Status	In theory, married people drive more safely
MVR_PTS	Motor Vehicle Record Points	If you get lots of traffic tickets, you tend to get into more crashes
OLDCLAIM	Total Claims (Past 5 Years)	If your total payout over the past five years was high, this suggests future payouts will be high
PARENT1	Single Parent	Unknown effect
RED_CAR	A Red Car	Urban legend says that red cars (especially red sports cars) are more risky. Is that true?
REVOKED	License Revoked (Past 7 Years)	If your license was revoked in the past 7 years, you probably are a more risky driver.
SEX	Gender	Urban legend says that women have less crashes then men. Is that true?
TIF	Time in Force	People who have been customers for a long time are usually more safe.
TRAVTIME	Distance to Work	Long drives to work usually suggest greater risk
URBANICITY	Home/Work Area	Unknown
YOJ	Years on Job	People who stay at a job for a long time are usually more safe

The *test* data set has 2141 rows and the *train* data set has 8161 rows. Both the data sets have 26 columns (or variables) displayed in *Figure-1*.

Below is a summary of all the variables in training data set:

Figure-2: Summary of the training data set

```
##      INDEX        TARGET_FLAG       TARGET_AMT        KIDSDRIV
##  Min.   :    1   Min.   :0.0000   Min.   :     0   Min.   :0.0000
##  1st Qu.: 2559   1st Qu.:0.0000   1st Qu.:     0   1st Qu.:0.0000
##  Median : 5133   Median :0.0000   Median :     0   Median :0.0000
```

```
##  Mean   : 5152   Mean   :0.2638   Mean   :  1504   Mean   :0.1711
##  3rd Qu.: 7745   3rd Qu.:1.0000   3rd Qu.:  1036   3rd Qu.:0.0000
##  Max.   :10302   Max.   :1.0000   Max.   :107586   Max.   :4.0000
##
##       AGE           HOMEKIDS           YOJ              INCOME
##  Min.   :16.00   Min.   :0.0000   Min.   : 0.0   $0      : 615
##  1st Qu.:39.00   1st Qu.:0.0000   1st Qu.: 9.0           : 445
##  Median :45.00   Median :0.0000   Median :11.0   $26,840 :   4
##  Mean   :44.79   Mean   :0.7212   Mean   :10.5   $48,509 :   4
##  3rd Qu.:51.00   3rd Qu.:1.0000   3rd Qu.:13.0   $61,790 :   4
##  Max.   :81.00   Max.   :5.0000   Max.   :23.0   $107,375:   3
##  NA's   :6                        NA's   :454    (Other) :7086
##  PARENT1        HOME_VAL    MSTATUS      SEX              EDUCATION
##  No :7084   $0      :2294   Yes :4894   M  :3786   <High School :1203
##  Yes:1077           : 464   z_No:3267   z_F:4375   Bachelors    :2242
##             $111,129:   3                          Masters      :1658
##             $115,249:   3                          PhD          : 728
##             $123,109:   3                          z_High School:2330
##             $153,061:   3
##             (Other) :5391
##             JOB           TRAVTIME            CAR_USE        BLUEBOOK
##  z_Blue Collar:1825   Min.   :  5.00   Commercial:3029   $1,500 : 157
##  Clerical     :1271   1st Qu.: 22.00   Private   :5132   $6,000 :  34
##  Professional :1117   Median : 33.00                     $5,800 :  33
##  Manager      : 988   Mean   : 33.49                     $6,200 :  33
##  Lawyer       : 835   3rd Qu.: 44.00                     $6,400 :  31
##  Student      : 712   Max.   :142.00                     $5,900 :  30
##  (Other)      :1413                                      (Other):7843
##       TIF                CAR_TYPE    RED_CAR       OLDCLAIM
##  Min.   : 1.000   Minivan    :2145   no :5783   $0     :5009
##  1st Qu.: 1.000   Panel Truck: 676   yes:2378   $1,310 :   4
##  Median : 4.000   Pickup     :1389              $1,391 :   4
##  Mean   : 5.351   Sports Car : 907              $4,263 :   4
##  3rd Qu.: 7.000   Van        : 750              $1,105 :   3
##  Max.   :25.000   z_SUV      :2294              $1,332 :   3
##                                                 (Other):3134
##     CLM_FREQ      REVOKED       MVR_PTS         CAR_AGE
##  Min.   :0.0000   No :7161   Min.   : 0.000   Min.   :-3.000
##  1st Qu.:0.0000   Yes:1000   1st Qu.: 0.000   1st Qu.: 1.000
##  Median :0.0000              Median : 1.000   Median : 8.000
##  Mean   :0.7986              Mean   : 1.696   Mean   : 8.328
##  3rd Qu.:2.0000              3rd Qu.: 3.000   3rd Qu.:12.000
##  Max.   :5.0000              Max.   :13.000   Max.   :28.000
##                                               NA's   :510
##                   URBANICITY
##  Highly Urban/ Urban  :6492
##  z_Highly Rural/ Rural:1669
##
##
##
##
##
```

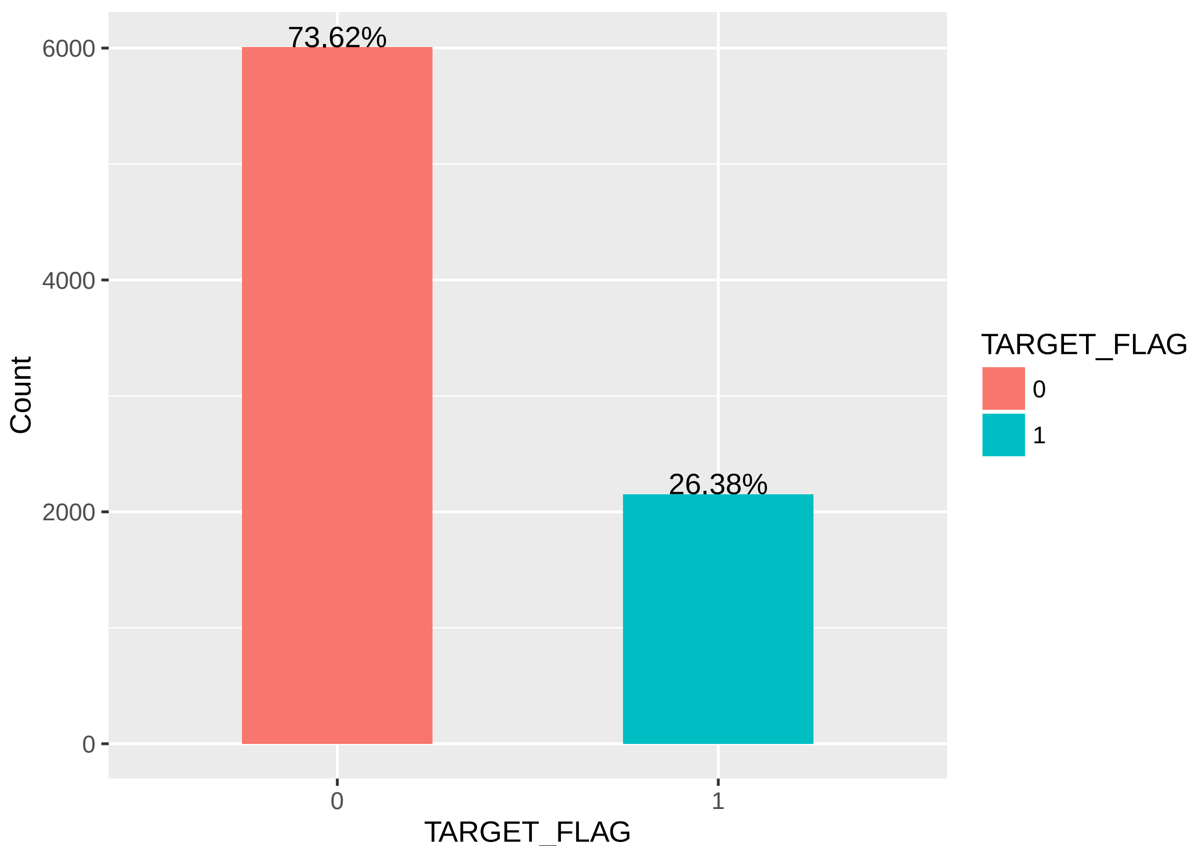
The summary details show that the variables (such as INCOME, HOME_VAL etc) representing money contain symbols such as “$” and “,”. There are also variables with NA (unavailable data). We will therefore perform the following data changes:

1. Create a dummy variable NA_AGE to represent NA values in AGE variable. If AGE has NA values, then the corresponding values in NA_AGE will have 1 else it will have 0. The NA values in AGE will be imputed with median values of AGE variable
2. Create a dummy variable NA_YOJ to represent NA values in the YOJ variable. If YOJ has NA values, then the corresponding values in NA_YOJ will have 1 else it will have 0. The NA values in YOJ will be imputed with median values of YOJ variable
3. Create a dummy variable NA_INCOME to represent NA values in INCOME variable. If INCOME has NA values, then the corresponding values in NA_INCOME will have 1 else it will have 0. The INCOME variable has “$” and “,”, so these characters will be deleted, and the variable will be converted to numeric. The NA values in INCOME will be imputed with median values of INCOME variable.
4. Create a dummy variable NA_HOME_VAL to represent NA values in the HOME_VAL variable. If HOME_VAL has NA values, then the corresponding values in NA_HOME_VAL will have 1 else it will have 0. The HOME_VAL variable has “$” and “,”, so these characters will be deleted, and the variable is converted to numeric. The NA values in NA_HOME_VAL will be imputed with 0 values, since we are assuming that NA values in the HOME_VAL variable represents that the driver is not a home owner.
5. The BLUEBOOK variable has “$” and “,”, so these characters will be deleted, and the variable converted to numeric.
6. The OLDCLAIM variable has “$” and “,”, so these characters will be deleted, and the variableconverted to numeric.
7. For MSTATUS, create a dummy variable DUMMY_MSTATUS, so that “Yes” will be represented by 1 and “No” by 0
8. For SEX, create a dummy variable DUMMY_SEX, so that “M” will be represented by 1 and “F” by 0
9. For PARENT1, create a dummy variable DUMMY_PARENT1, so that “Yes” will be represented by 1 and “No” by 0
10. For CAR_AGE replace all NA with median values of CAR_AGE, and convert the negative value to positive value, since negative values might have been accidentally entered while gathering the data.
11. For EDUCATION, create 4 dummy variables DUMMY_NO_HS, DUMMY_HS, DUMMY_BACHELOR, DUMMY_MASTERS to represent “< high school”“,”high school“,”Bachelors“,”Masters" respectively. A value of 1 in the corresponding dummy variable represents the respective level of education, and a value of 0 in all the dummy variables represent PhD as the level of education.
12. The JOB variable has 8 job levels and also NA values. We will create 8 dummy variables “DUMMY_Clerical”, “DUMMY_Doctor”,“DUMMY_Home_Maker”, “DUMMY_Lawyer”, “DUMMY_Manager”, “DUMMY_Professional” and “DUMMY_Student” and “DUMMY_Blue Collar” representing the levels “Clerical”, “Doctor”, “Home Maker”, “Lawyer”, “Manager”, “Professional”, “Student”, “z_Blue Collar” respectively. These dummy variables will contain 1, if the observation has the corresponding level as the JOB. But if the JOB variable value is unknown, then all these dummy variables will have 0 values. This means, we are treating the unknown variables as separate values.
13. For URBANICITY, create a dummy variable DUMMY_URBANICITY variable, such that “Highly Urban/ Urban” is represented as 1, 0 for “Highly Rural/Rural” value
14. For CAR_USE, create a dummy variable DUMMY_CAR_USE, such that “Commercial” car use is represented as 1, and “Private” as 0.

15. The CAR_TYPE variable contains the car type, and it has 6 different values. We will create 5 dummy variables "DUMMY_Minivan" "DUMMY_Panel_Truck", "DUMMY_Pickup", "DUMMY_Sports_Car" and "DUMMY_Van" to represent "Minivan", "Panel Truck", "Pickup", "Sports Car", "Van" rerpectively. If these dummy variables contain 1 for an observation, then that observation has the respective CAR_TYPE level. If all these dummy variables contain 0, then that represents the "SUV" CARTYPE.
16. For RED_CAR, create a dummy variable DUMMY_RED_CAR to represent a "YES" with 1, and "NO" with 0.
17. For REVOKED, create a dummy variable DUMMY_REVOKED to represent a "YES" with 1, and "NO" with 0.
18. We will create a new variable called DUMMY_HOME_OWNER that represents if the driver is a home owner. If the variable HOME_VAL has a value greater than 0, then this variable will have 1, else it will have 0. For HOME_VAL with NA values, the DUMMY_HOME_OWNER variable will have 0.

Let us check how the TARGET_FLAG's data is distributed.

Figure-3: TARGET_FLAG classes proportion



From *Figure-3* we can conclude that we have an approximate ratio of 74%:26% between 0 and 1 classes of TARGET_FLAG. There is no severe data imbalance among the classes. However, without any modeling effort, we can still achieve an accuracy of 73.62%, if we classify all the samples as class-0. We can view this as the *null model* accuracy. Our model should have a better accuracy than the null model, and our model should accurately predict the class-1 cases. To achieve this, we need to estimate an optimal cut-off point for the class-1 probability, instead of using the default 0.5 cut-off (i.e., classifying the sample as class-1, when the probability of class-1 is greater than 0.5).

Data Preparation

Based on the modifications listed in data exploration, we modified the training and test data sets. We also created another data set named "train_df_mod", which has the columns that would be used in building the models. The list of variables present in "train_df_mod" are displayed below:

Figure-4: Modified training data set's variables

```
##  [1] "TARGET_FLAG"        "TARGET_AMT"         "KIDSDRIV"
##  [4] "AGE"                "HOMEKIDS"           "YOJ"
##  [7] "INCOME"             "HOME_VAL"           "TRAVTIME"
## [10] "BLUEBOOK"           "TIF"                "OLDCLAIM"
## [13] "CLM_FREQ"           "MVR_PTS"            "CAR_AGE"
## [16] "NA_AGE"             "NA_YOJ"             "NA_INCOME"
## [19] "NA_HOME_VAL"        "DUMMY_HOME_OWNER"   "DUMMY_MSTATUS"
## [22] "DUMMY_SEX"          "DUMMY_PARENT1"      "NA_CAR_AGE"
## [25] "DUMMY_NO_HS"        "DUMMY_HS"           "DUMMY_BACHELOR"
## [28] "DUMMY_MASTERS"      "DUMMY_Clerical"     "DUMMY_Doctor"
## [31] "DUMMY_Home_Maker"   "DUMMY_Lawyer"       "DUMMY_Manager"
## [34] "DUMMY_Professional" "DUMMY_Student"      "DUMMY_Blue_Collar"
## [37] "DUMMY_URBANICITY"   "DUMMY_CAR_USE"      "DUMMY_MINI_VAN"
## [40] "DUMMY_Panel_Truck"  "DUMMY_Pickup"       "DUMMY_Sports_Car"
## [43] "DUMMY_Van"          "DUMMY_RED_CAR"      "DUMMY_REVOKED"
```

The modified data set has 45 columns, while the initial training data set has 26 columns. The summary information of all the variables in the modified data set is given in Appendix-A , *Figure-A.1*.

Model building

We will build 3 models based for identifying the TARGET_FLAG value. Two models are based on the logistic regression, and the third model will be based on KNN (K Nearest Negihbors). The first model will include all the columns of the training data, including the associations between the NA dummy variables with the corresponding variables. The *stepAIC()* function of MASS library is used to select the significant variables (using mixed variable selection method). *Model-1* is again rebuilt to include only the significant variables indentified by *stepAIC()* function.

We will build the second model (*Model-2*), using second order polynomials of the variables used in *Model-1*. For the third model (*Model-3*), we will use a non-parametric method known as K Nearest Negihbors. The three models will be compared using AUC (Area Under the Cure) of ROC (Receiver Operating Characteristics). The model that has the least AUC will be dropped from further consideration. If there is a tie, then we will use 5 fold Cross Validation to estimate the classification error. Once we determine the best model among the 3 models, we will determine the cut-off probability (i.e., probability that TARGET_FLAG=1), so that the *sensitivity* of the model is improved. The identified cut-off point should have better sensitivity than the NULL model, and the accuracy should be at least that of the NULL model. When we decrease the threshold value, the accuracy usually decreases, but the sensitivity increases. The challenge is to identify the optimal threshold, so that both sensitivity and accruacy are maximized.

Once the probabilities of TARGET_FLAG=1 are identified (let us address the probabilities variable as PROB), we will use the TARGET_FLAG variable, PROB variable and other variables in the data set to predict the potential claim amount using linear regression techniques.

Building *Model-1*

Using the *glm()* function we fit a basic logistic regression model. The summary of the *model-1* coefficients is given below:

Figure-5: Variable coefficients, std. errors and p-values of logistic model

Variable	Coefficient	Std_Error	P_value
(Intercept)	-2.9821728	0.3402136	0.0000000
KIDSDRIV	0.3865450	0.0613036	0.0000000
AGE	-0.0017381	0.0040361	0.6667398
NA_AGE	2.1932008	1.2561064	0.0808056
HOMEKIDS	0.0489225	0.0371917	0.1883708
YOJ	-0.0101847	0.0086176	0.2372663
NA_YOJ	0.0726288	0.1268888	0.5670638
INCOME	-0.0000043	0.0000012	0.0003525
NA_INCOME	-0.0367467	0.1299585	0.7773629
HOME_VAL	-0.0000004	0.0000006	0.4803932
NA_HOME_VAL	-0.3011712	0.1363558	0.0271944
TRAVTIME	0.0146096	0.0018853	0.0000000
BLUEBOOK	-0.0000207	0.0000053	0.0000844
TIF	-0.0555275	0.0073550	0.0000000
OLDCLAIM	-0.0000140	0.0000039	0.0003654
CLM_FREQ	0.1967184	0.0285737	0.0000000
MVR_PTS	0.1125855	0.0136568	0.0000000
CAR_AGE	-0.0002650	0.0075556	0.9720261
NA_CAR_AGE	0.1484140	0.1182280	0.2093626
DUMMY_HOME_OWNER	-0.2670011	0.1533007	0.0815640
DUMMY_MSTATUS	-0.4541069	0.0867491	0.0000002
DUMMY_SEX	0.0933646	0.1121694	0.4052097
DUMMY_PARENT1	0.3671877	0.1100200	0.0008455
DUMMY_NO_HS	0.1847098	0.2143255	0.3887870
DUMMY_HS	0.1961520	0.1964020	0.3179270
DUMMY_BACHELOR	-0.2109788	0.1796790	0.2403156
DUMMY_MASTERS	-0.1173735	0.1525445	0.4416335
DUMMY_Clerical	0.4287632	0.1967645	0.0293267
DUMMY_Doctor	-0.4380204	0.2668167	0.1006620
DUMMY_Home_Maker	0.2557246	0.2105987	0.2246429
DUMMY_Lawyer	0.1121885	0.1693719	0.5077281
DUMMY_Manager	-0.5501017	0.1713519	0.0013257
DUMMY_Professional	0.1657778	0.1783367	0.3525897
DUMMY_Student	0.1300383	0.2197979	0.5541005
DUMMY_Blue_Collar	0.3140648	0.1855525	0.0905329
DUMMY_URBANICITY	2.3944103	0.1130193	0.0000000
DUMMY_CAR_USE	0.7700247	0.0920621	0.0000000
DUMMY_MINI_VAN	-0.7706134	0.1113970	0.0000000
DUMMY_Panel_Truck	-0.2249201	0.2004360	0.2617968
DUMMY_Pickup	-0.2162545	0.1168223	0.0641493
DUMMY_Sports_Car	0.2583883	0.0983096	0.0085810
DUMMY_Van	-0.1658427	0.1591287	0.2973225
DUMMY_RED_CAR	-0.0191415	0.0866398	0.8251454
DUMMY_REVOKED	0.8825388	0.0914506	0.0000000

Figure-5 shows that the p-value of some of the variables are greater than 0.05. Only the following variables have a p-value of less than 0.05:

Figure-6: Model-1 (Logistic regression) variables, which have p-values greater than 0.05

Variable	Coefficient	Std_Error	P_value
(Intercept)	-2.9821728	0.3402136	0.0000000
KIDSDRIV	0.3865450	0.0613036	0.0000000
INCOME	-0.0000043	0.0000012	0.0003525
NA_HOME_VAL	-0.3011712	0.1363558	0.0271944
TRAVTIME	0.0146096	0.0018853	0.0000000
BLUEBOOK	-0.0000207	0.0000053	0.0000844
TIF	-0.0555275	0.0073550	0.0000000
OLDCLAIM	-0.0000140	0.0000039	0.0003654
CLM_FREQ	0.1967184	0.0285737	0.0000000
MVR_PTS	0.1125855	0.0136568	0.0000000
DUMMY_MSTATUS	-0.4541069	0.0867491	0.0000002
DUMMY_PARENT1	0.3671877	0.1100200	0.0008455
DUMMY_Clerical	0.4287632	0.1967645	0.0293267
DUMMY_Manager	-0.5501017	0.1713519	0.0013257
DUMMY_URBANICITY	2.3944103	0.1130193	0.0000000
DUMMY_CAR_USE	0.7700247	0.0920621	0.0000000
DUMMY_MINI_VAN	-0.7706134	0.1113970	0.0000000
DUMMY_Sports_Car	0.2583883	0.0983096	0.0085810
DUMMY_REVOKED	0.8825388	0.0914506	0.0000000

Based on the p-values, these are the significant variables. But we will use mixed variable selection to identify significant variables. We cannot use p-value to determine if a variable is significant, since there is always 5% chance (assuming 95% significance level) that we might have obtained a p-value of less than 0.05 (or 5%) just by chance. We therefore need to use a variable selection method in order to identify the important variables.

The *stepAIC()* function of MASS library is now used to identify the significant variables. These variables are used to build *Model-1*. The variable coefficients and p-values of the variables are displayed in *Figure-7*.

Figure-7: Model-1, built using significant variables only

Variable	Coefficient	Std_Error	P_value
(Intercept)	-2.9877986	0.1789623	0.0000000
KIDSDRIV	0.3841405	0.0601383	0.0000000
NA_AGE	2.0844474	1.2426900	0.0934707
HOMEKIDS	0.0536153	0.0340112	0.1149337
YOJ	-0.0120648	0.0080239	0.1326818
INCOME	-0.0000049	0.0000009	0.0000001
NA_HOME_VAL	-0.2871134	0.1345452	0.0328465
TRAVTIME	0.0145761	0.0018823	0.0000000
BLUEBOOK	-0.0000243	0.0000042	0.0000000
TIF	-0.0555659	0.0073430	0.0000000
OLDCLAIM	-0.0000141	0.0000039	0.0003291
CLM_FREQ	0.1971224	0.0285121	0.0000000
MVR_PTS	0.1124211	0.0136161	0.0000000

Variable	Coefficient	Std_Error	P_value
DUMMY_HOME_OWNER	-0.3377710	0.0799887	0.0000241
DUMMY_MSTATUS	-0.4649555	0.0846638	0.0000000
DUMMY_PARENT1	0.3670721	0.1090749	0.0007645
DUMMY_NO_HS	0.3593640	0.1024052	0.0004494
DUMMY_HS	0.3937770	0.0793030	0.0000007
DUMMY_Clerical	0.2586084	0.0958074	0.0069495
DUMMY_Doctor	-0.4219564	0.2206405	0.0558229
DUMMY_Manager	-0.6861196	0.1093040	0.0000000
DUMMY_Blue_Collar	0.1754095	0.0871672	0.0441847
DUMMY_URBANICITY	2.3944253	0.1126806	0.0000000
DUMMY_CAR_USE	0.7004020	0.0757435	0.0000000
DUMMY_MINI_VAN	-0.6901413	0.0791184	0.0000000
DUMMY_Pickup	-0.1222357	0.0830584	0.1411063
DUMMY_Sports_Car	0.2808015	0.0952582	0.0032005
DUMMY_REVOKED	0.8845958	0.0913060	0.0000000

Figure-7 shows that we still have some variables included in the model that have a high p-value (greater than 0.05). Even though we have variables with higher p-values, we still want to include them in the model since there is always a chance of getting Type-2 error, and the variable selection method avoids type-1 and type-2 errors.

Using logistic regression, we obtained the following model (*Model-1*):

$$P(TARGET\_FLAG = 1) = \frac{e^{f(x)}}{1 + e^{f(x)}}$$

where

$$f(x) = -2.9873679 + 0.3840843 KIDSDRIV + 2.0853961 NA\_AGE +$$
$$0.0536735 HOMEKIDS + -0.0119663 YOJ - 0.0000049 INCOME$$
$$-0.2864266 NA\_HOME\_VAL + 0.014592 TRAVTIME - 0.0000243 BLUEBOOK$$
$$-0.0555523 TIF - 0.0000139 OLDCLAIM + 0.1976401 CLM\_FREQ$$
$$0.1121727 MVR\_PTS - 0.3359639 DUMMY\_HOME\_OWNER - 0.467941 DUMMY\_MSTATUS +$$
$$0.3651455 DUMMY\_PARENT1 + 0.3579248 DUMMY\_NO\_HS + 0.3929838 DUMMY\_HS +$$
$$0.2569926 DUMMY\_Clerical + -0.4239305 DUMMY\_Doctor - 0.6878651 DUMMY\_Manager +$$
$$0.1749694 DUMMY\_Blue\_Collar + 2.394212 DUMMY\_URBANICITY + 0.6983876 DUMMY\_CAR\_USE$$
$$-0.6897985 DUMMY\_MINI\_VAN + -0.1189946 DUMMY\_Pickup + 0.2805493 DUMMY\_Sports\_Car +$$
$$0.8862538 DUMMY\_REVOKED$$

If a variable has a positive coefficient, then the probability of filing the claim increases (given all other variables are constant), else the probability decreases. Based on this criteria, we can make the following inferences:

- The variables NA_AGE and DUMMY_URBANICITY have bigger coefficients (when compared to the other variables). This implies that these two variables increase the probability of TARGET_FLAG=1, given the other variables are constant.
- Additionally, the probability of filing a claim increases

- if kids drive the car (KIDSDRIV=1)
  - if the family has kids (HOMEKIDS=1)
  - if travel time is more (TRAVTIME)
  - if claim frequency is more (CLM_FREQ)
  - if MVR_PTS is more
  - if the driver is a single parent (DUMMY_PARENT1=1)
  - If education level is less than or equal to the high school (DUMMY_NO_HS=1, DUMMY_HS=1)
  - if the job is clerical (DUMMY_Clerical), or blue collar(DUMMY_Blue_Collar=1)
  - if the car is used commercially (DUMMY_CAR_USE=1)
  - if the car is a sports car (DUMMY_Sports_Car=1)
  - if the license is revoked (DUMMY_REVOKED=1)

- The claim probability decreases with the following an increase in the following variables or if the variable is enabled (in case of dummy or NA place holder variables):
  - Years in Job (YOJ) INCOME, NA_HOME_VAL, BLUEBOOK, TIF, OLDCLAIM, DUMMY_HOME_OWNER, DUMMY_MSTATUS, DUMMY_DOCTOR, DUMMY_Manager, DUMMY_MINI_Van, DUMMY_PICKUP

- The extent of the probability change depends on the coefficient of the corresponding variable
- In all other cases the probability remains unchanged

Building *Model-2*

We will build *Model-2* by raising all the *Model_1* variables to the power of 2, and subsequently applying the *stepAIC()* function for variable selection. This process has gives us the following model:

Figure-8: Model-2, built using Model-1 variables raised to the power of 2

Variable	Coefficient	Std_Error	P_value
(Intercept)	-3.2133646	0.1364233	0.0000000
poly(KIDSDRIV, 2)1	19.3857985	2.5624631	0.0000000
poly(KIDSDRIV, 2)2	-4.6235790	2.4277729	0.0568512
NA_AGE	2.0963478	1.2539963	0.0945769
poly(YOJ, 2)1	-2.2364491	2.9741934	0.4520802
poly(YOJ, 2)2	6.2480250	2.8840556	0.0302805
poly(INCOME, 2)1	-20.8015709	3.7674842	0.0000000
poly(INCOME, 2)2	6.1559226	3.2356315	0.0571004
NA_HOME_VAL	-0.2874481	0.1348615	0.0330536
poly(TRAVTIME, 2)1	20.8173688	2.7834197	0.0000000
poly(TRAVTIME, 2)2	-8.0387343	3.1385173	0.0104276
poly(BLUEBOOK, 2)1	-17.7985317	3.1121326	0.0000000
poly(BLUEBOOK, 2)2	7.6886437	2.6931103	0.0043046
poly(TIF, 2)1	-19.9592126	2.7203314	0.0000000
poly(TIF, 2)2	5.9106936	2.7472087	0.0314346
poly(OLDCLAIM, 2)1	-12.9103117	3.5942995	0.0003283
poly(OLDCLAIM, 2)2	-3.8470269	3.1642650	0.2240715
poly(CLM_FREQ, 2)1	21.2635951	3.8189576	0.0000000

Variable	Coefficient	Std_Error	P_value
poly(CLM_FREQ, 2)2	-7.3595082	2.8080940	0.0087719
poly(MVR_PTS, 2)1	19.5363641	2.7443636	0.0000000
poly(MVR_PTS, 2)2	5.7422528	2.5466623	0.0241449
DUMMY_HOME_OWNER	-0.3078101	0.0805008	0.0001315
DUMMY_MSTATUS	-0.4970594	0.0833290	0.0000000
DUMMY_PARENT1	0.4171775	0.0958412	0.0000134
DUMMY_NO_HS	0.2851978	0.1066157	0.0074728
DUMMY_HS	0.3610971	0.0810916	0.0000085
DUMMY_Clerical	0.3037345	0.0980939	0.0019591
DUMMY_Doctor	-0.4287064	0.2213810	0.0528054
DUMMY_Manager	-0.6401350	0.1102225	0.0000000
DUMMY_Blue_Collar	0.2728819	0.0913106	0.0028035
DUMMY_URBANICITY	2.3665305	0.1134790	0.0000000
DUMMY_CAR_USE	0.6723707	0.0764122	0.0000000
DUMMY_MINI_VAN	-0.6767584	0.0796769	0.0000000
DUMMY_Pickup	-0.1215004	0.0833888	0.1451068
DUMMY_Sports_Car	0.2622784	0.0961850	0.0063949
DUMMY_REVOKED	0.9351259	0.0928902	0.0000000

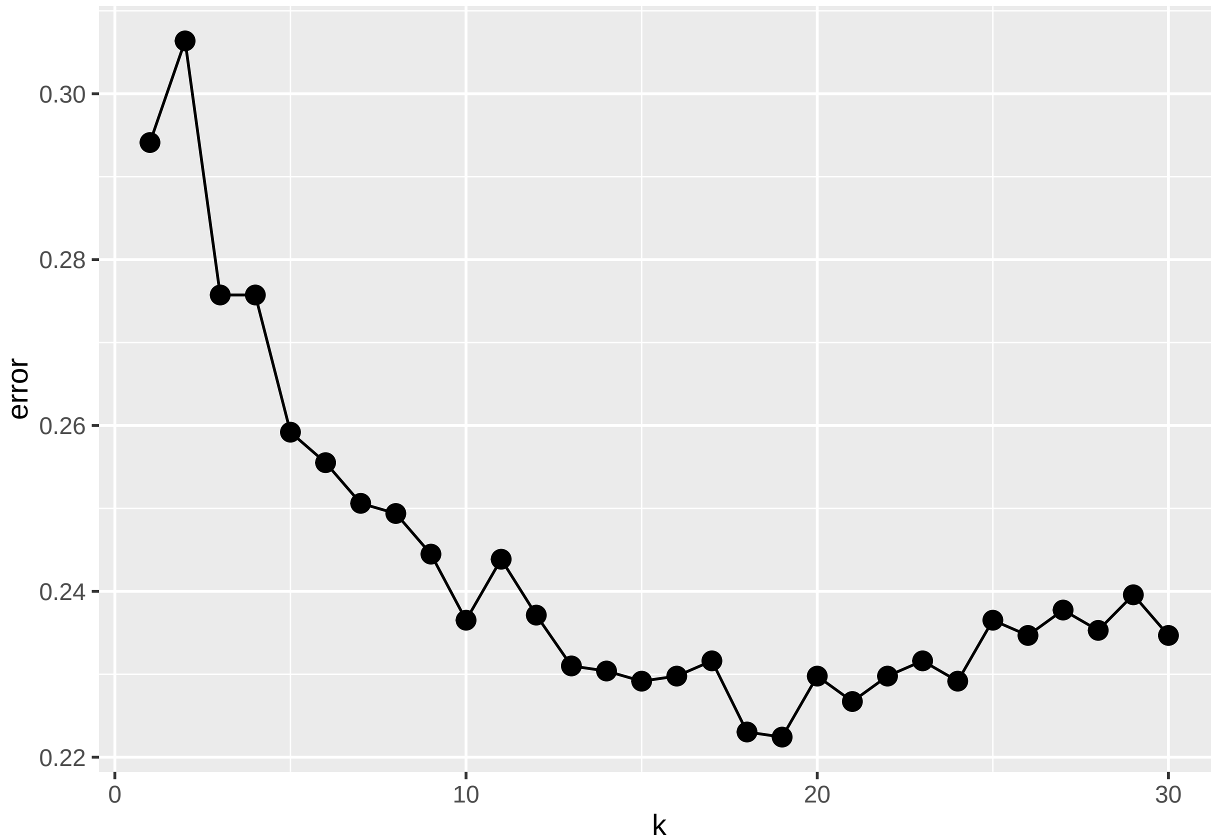
Unlike *Model-1*, *Model-2* cannot be easily interpreted, since *Model-2* has quadratic terms. So we are not presenting the effect of a variable on the probability of claim for *Model-2*. However we will still test *Model-2* for performance using ROC and 5-fold Cross Validation technique.

Building *Model-3*

We will use K Nearest Neighbors(KNN) to build the third model. KNN works by finding K nearest observations for the input value, and calculating the probability that the input observation belongs to a specific class based on the identified K neighbors. The optimal number of neighbors is found by dividing the given data set into two data sets: training and test data. The training data set is used to predict the output of the test dataset's observations for various values of K. The value of K at which we get the minimum test error, will be selected as the optimal value of K.

We first randomly divide our training data frame(train_df_mod) into two data frames, knn_train and knn_test such that knn_train will have 80% of the observations from train_df_mod, while the knn_test will have the remaining 20% of the observations from our training data (train_df_mod). Also the input variables are scaled before finding the optimal k value (Scaling refers to subtraction of variable's mean from the variable's value, and dividing the result by the variable's standard deviation). As the value of k increases, the flexibility of the model decreases. In other words as the value of K increases,the bias of the model increases, and the variance of the model decreases. We therefore must select an optimal value of K at which both the bias and variance are minimum. The following figure displays the error rate at various values of K, when KNN algorithm is applied on the training data set. From this figure, we can find that at K=19, we have the minimum error.

Figure-8: Finding the optimal value of K for KNN algorithm



Sine KNN is a non-parametric model, we cannot interpret the model.

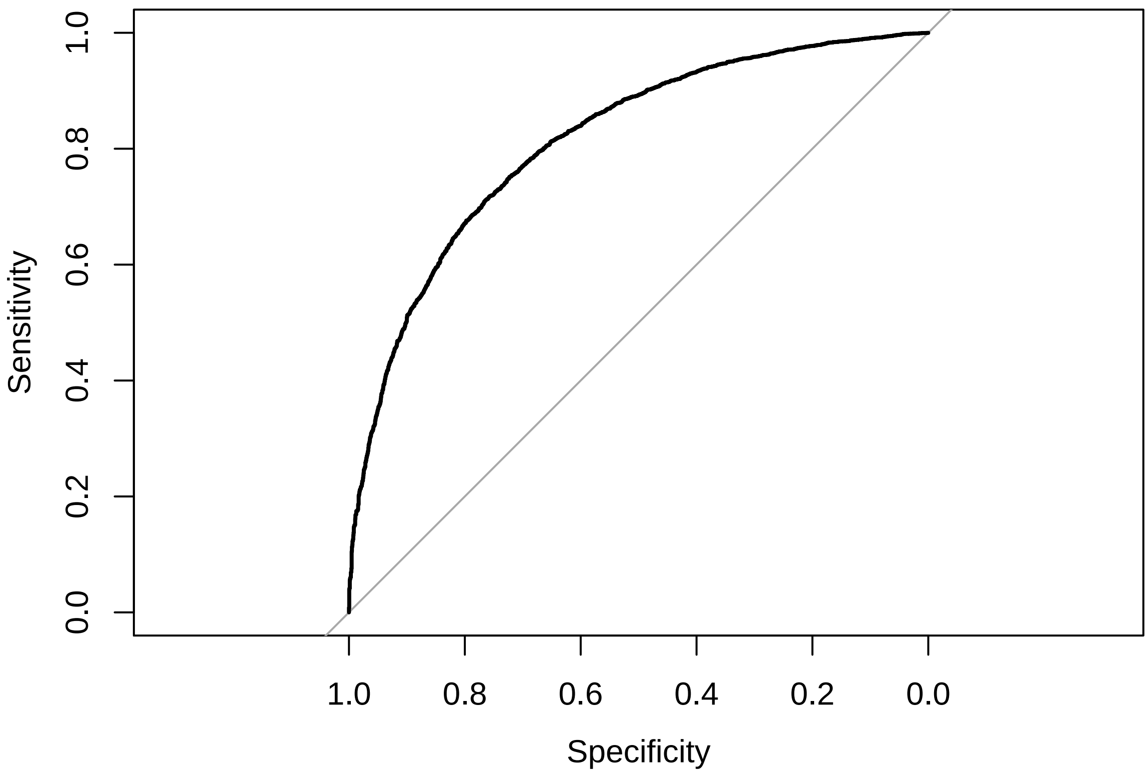
Model evaluation

We will evaluate *Model-1*, *Model-2* and *Model-3* using ROC.

Evaluating ***Model-1*** performance

The following figure shows the ROC curve for *Model-1.* The Area Under the Curve (AUC) for this model is 0.8134.

Figure-9: ROC curve for *Model-1*

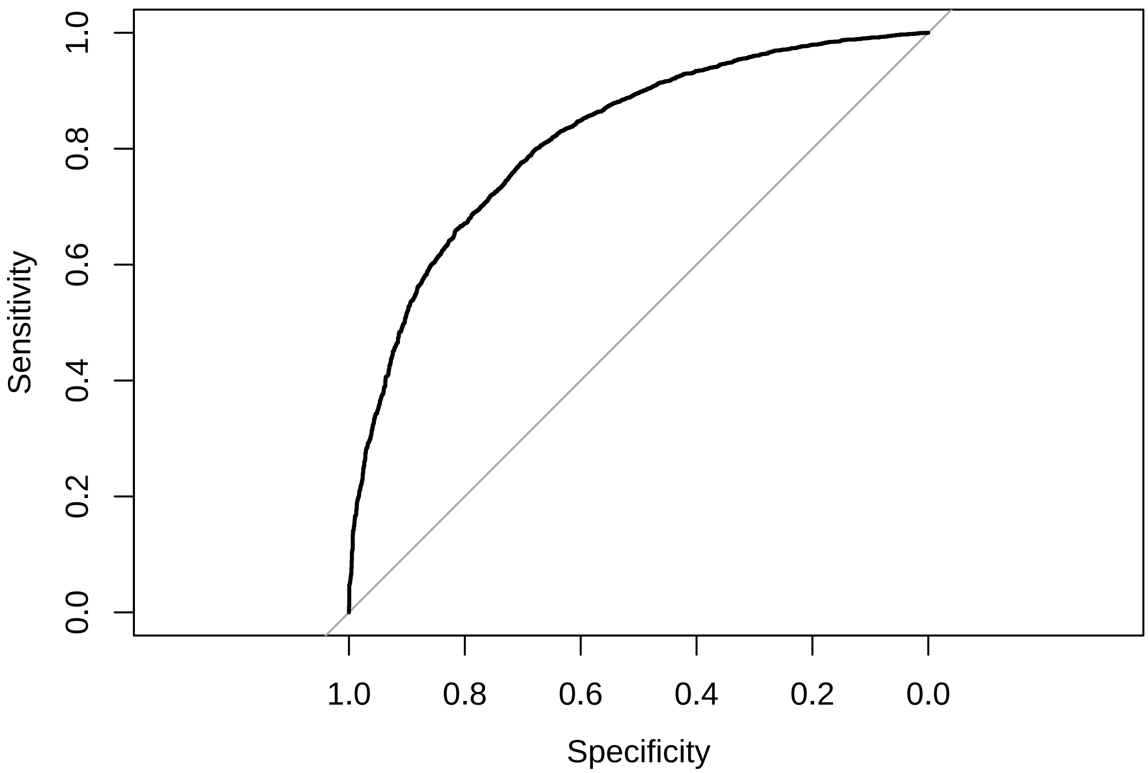



```
##
## Call:
## roc.default(response = train_df_mod$TARGET_FLAG, predictor = prob,     levels = rev(levels(as.factor
##
## Data: prob in 2152 controls (train_df_mod$TARGET_FLAG 1) > 6008 cases (train_df_mod$TARGET_FLAG 0).
## Area under the curve: 0.8134
```

Evaluating *Model-2* performance

The following figure shows the ROC curve for *Model-2*. The Area Under the Curve (AUC) for this model is 0.8167.

Figure-10: ROC curve for *Model-2*

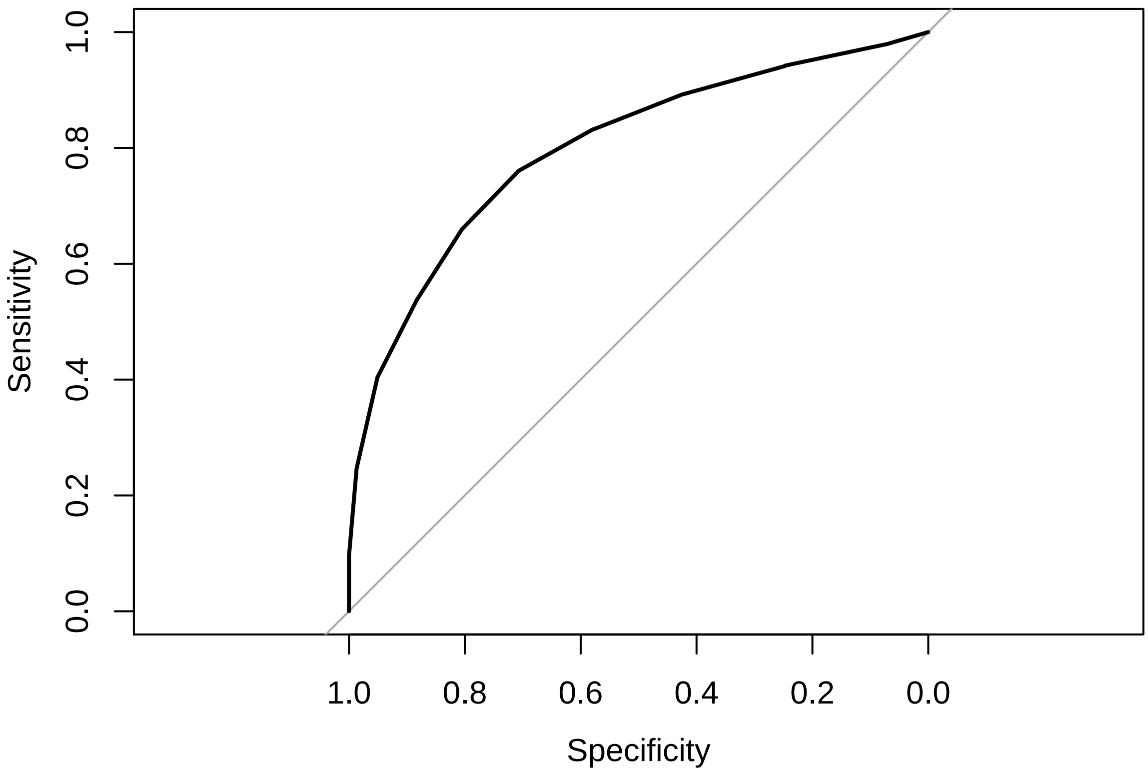



```
##
## Call:
## roc.default(response = train_df_mod$TARGET_FLAG, predictor = prob,     levels = rev(levels(as.factor
##
## Data: prob in 2152 controls (train_df_mod$TARGET_FLAG 1) > 6008 cases (train_df_mod$TARGET_FLAG 0).
## Area under the curve: 0.8168
```

Evaluating *Model-3* performance

The following figure shows the ROC curve for *Model-3*. The Area Under the Curve (AUC) for this model is 0.7991.

Figure-11: ROC curve for *Model-3*




```
##
## Call:
## roc.default(response = actual, predictor = prob, levels = rev(levels(as.factor(actual))))
##
## Data: prob in 2152 controls (actual 1) < 6008 cases (actual 0).
## Area under the curve: 0.7993
```

The AUC of the three models is almost same (*Model-3* has the least AUC). Since *Model-3* is based on non-parametric model, we drop this model from further consideration, since non-parametric models are complex in nature and have high variance. Since the AUC for *Model-1* and *Model-2* are approximately same, we may drop *Model-2* since, *Model-2* is a quadratic model (more complex than *Model-1*). However we will check the cross validation error, and determine which model should be considered.

A 5 fold cross validation for *Model-1* and *Model-2* has given approximately the same error (*Model-1* Cross validation error is 0.1463688, while *Model-2* Cross Validation error is 0.1455384.

Since the AUC and Cross Validation errors are similar for both *Model-1* and *Model-2*, we reject *Model-2*, since this model is complex, when compared to *Model-1* (*Model-1* does not have any quadratic terms).

Computing the optimal threshold for P(TARGET_FLAG=1)

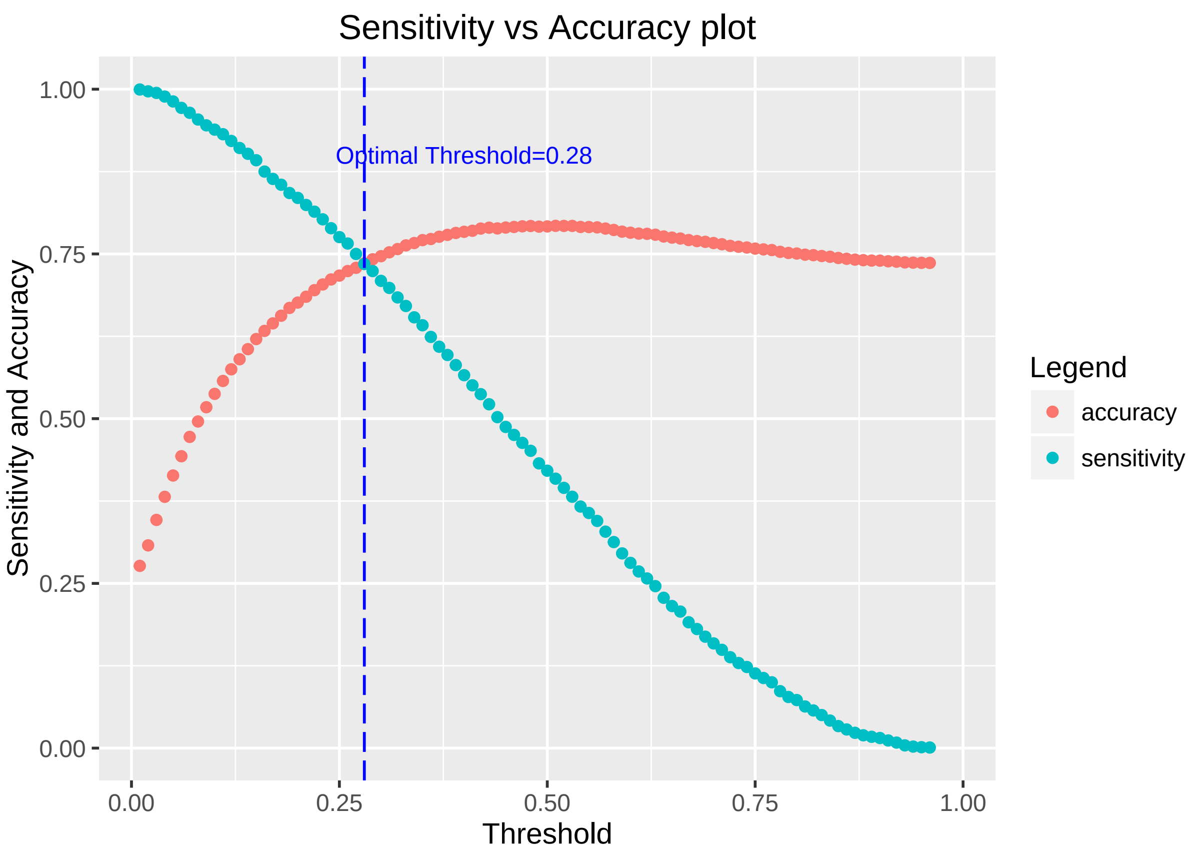
Let us find the accuracy, sensitivity and other performance details of *Model-1*. The model uses 0.5 as the threshold i.e., if the predicted probability of TARGET_FLAG = 1 is greater than or equal to 0.5, then the model predicts the target class as 1, else 0.

Figure-12: *Model-1* performance details with threshold=0.5

```
## Confusion Matrix and Statistics
##
##           actual
## predicted    0    1
##         0 5555 1246
##         1  453  906
##
##                Accuracy : 0.7918
##                  95% CI : (0.7828, 0.8006)
##     No Information Rate : 0.7363
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.392
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4210
##             Specificity : 0.9246
##          Pos Pred Value : 0.6667
##          Neg Pred Value : 0.8168
##              Prevalence : 0.2637
##          Detection Rate : 0.1110
##    Detection Prevalence : 0.1665
##       Balanced Accuracy : 0.6728
##
##        'Positive' Class : 1
##
```

The accuracy of the model is 79%, while the sensitivity is just 42%. This means our model is not doing well in predicting the true positive cases. But this model is better than the NULL model (which predicts all the cases as 0. The NULL model has 0 sensitivity and 0.74 accuracy. See *Figure-3*). The low sensitivity of *Model-1* can be fixed by reducing the threshold to a value lesser than 0.5. Reducing the threshold will decrease the model's accruacy, but it will increase the sensitivity of the model. We have to identify an optimal threshold at which both the sensitivity and accruacy are maximized. We identify the optimal threshold value by plotting the accuracy and sensitivity of the model at various threshold points.

Figure-13: Threshold vs sensitivity and accuracy plot



From *Figure-13* we can infer that at a threshold value of 0.28, we are getting an accuracy of approximately 0.74 and sensitivity of approximately 0.74. The accuracy is equal to NULL model, but the sensitivity is way above NULL model. The confusion matrix and other performance metrics of *Model-1* at 0.28 threshold value is displayed below in *Figure-14*.

Figure-14: *Model-1* performance details with threshold=0.28

```
## Confusion Matrix and Statistics
##
##          actual
## predicted    0    1
##         0 4422  571
##         1 1586 1581
##
##                Accuracy : 0.7357
##                  95% CI : (0.7259, 0.7452)
##     No Information Rate : 0.7363
##     P-Value [Acc > NIR] : 0.5557
##
##                   Kappa : 0.4088
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7347
##             Specificity : 0.7360
```

```
##          Pos Pred Value : 0.4992
##          Neg Pred Value : 0.8856
##              Prevalence : 0.2637
##          Detection Rate : 0.1938
##    Detection Prevalence : 0.3881
##       Balanced Accuracy : 0.7353
##
##        'Positive' Class : 1
##
```

Figure-14 shows that the accuracy of the model has dcreased but its sensitivity has increased.

Model building for TARGET_AMT

To predict the TARGET_AMT, we will create two new variables (PROB and TARGET_FLAG_PRED) in train_df_mod data frame. The PROB variable will contain the value probability that TARGET_FLAG=1, and the other variable TARGET_FLAG_PRED will contain the predicted TARGET_FLAG value. We cannot use the TARGET_FLAG variable in our model since this variable needs to be predicted first, and based on this value, the TARGET_AMT variable's value should be predicted.

Our strategy is to fit a linear regression model for TARGET_AMT variable, using all the variables except the TARGET_FLAG variable. In the place of TARGET_FLAG we will use TARGET_FLAG_PRED variable. From the linear model, we will identify the important variables using variables selection method (*stepAIC()*) function. Then the model is created using just the important variables identified by *stepAIC()* function. We call that model as *Model-reg-1*. Based on the residual plots of *Model-reg-1* we will transform perform any required transformations and build other models.

Building _Model-reg-1 regression model

Using *glm()* function of R, followed by the *stepAIC()* function, the following linear model is produced:

Figure-15: *Model-reg-1* coefficients, std. deviations and p-values

Variable	Coefficient	Std_Error	P_value
(Intercept)	-413.148599	170.7492159	0.0155584
BLUEBOOK	0.029125	0.0061786	0.0000025
MVR_PTS	46.757773	26.2247966	0.0746300
CAR_AGE	-17.680651	10.0887369	0.0797231
DUMMY_SEX	170.434198	100.7692784	0.0908118
DUMMY_HS	-187.786190	123.3013607	0.1278011
DUMMY_REVOKED	-300.711046	161.5878133	0.0627835
PROB	5835.286984	279.8599611	0.0000000

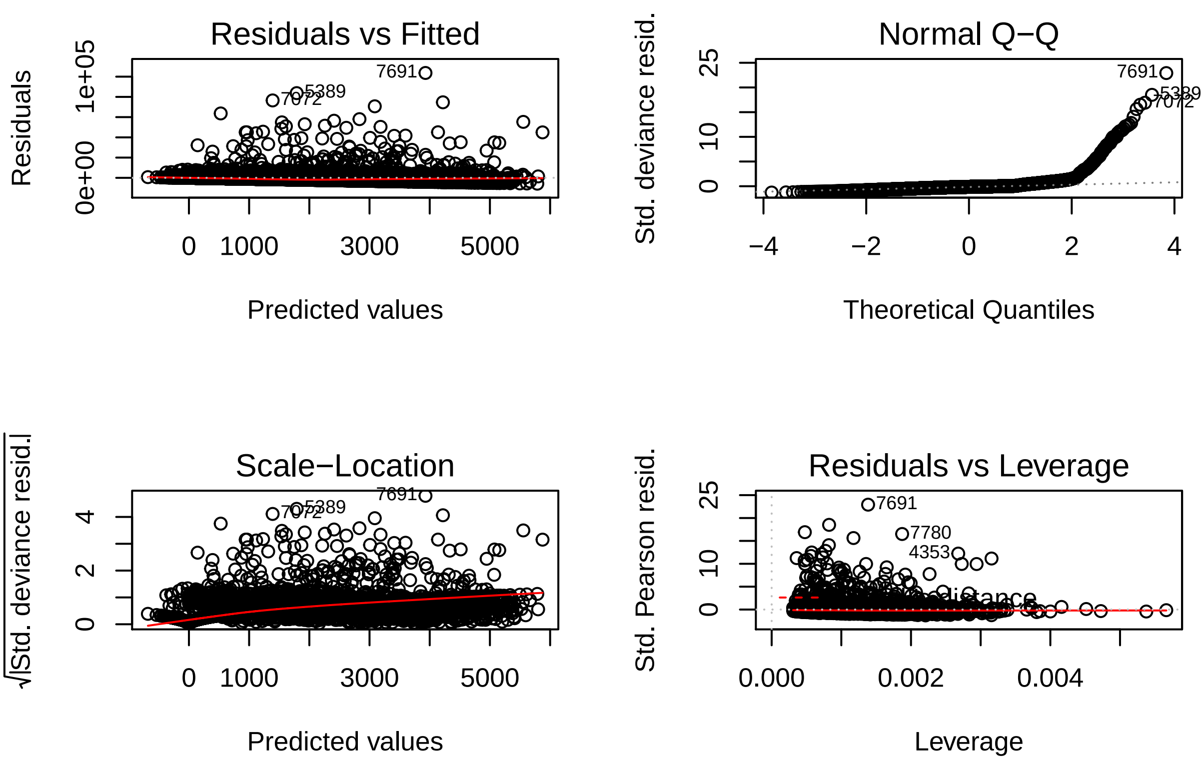
Figure-15 shows that, except the PROB and BLUEBOOK variables p-values, all other variable's p-values are pretty high (greater than 0.05). However we will consider all the variables in the model, since these are the significant variables identified by *stepAIC()* variable selection method.

Model_reg-1 is formally defined as: TARGET_AMT = TARGET_FLAG_PRED(-413.148 + 0.029125BLUEBOOK+ 46.757773MVR_PTS -17.680651CAR_AGE+ 170.434198DUMMY_SEX -187.786190DUMMY_HS-300.711046DUMMY_REVOKED+ 5835.0180201PROB)

We are multiplying the regression model obtained with TARGET_FLAG_PROD, since we want to make the TARGET_AMT as 0, whenever we predict the TARGET_FLAG as 0. Also from the model we can infer that the PROB coefficient is very huge when compared to the other coefficients. Surprisingly if someone's license is revoked, then the claim amount might decrease (since DUMMY_REVOKED has a coefficient of -300). But if MVR_PTS are high, then the claim amount will increase. The claim amount for males is also higher (by about 170$) than females, since DUMMY_SEX has a coefficient of 170. If the license is revoked, then the claim amount will decrease. If the driver has a high school degree then the claim amount will decrease.

Let us plot the residual plots of the *Model-reg-1* model. The residual plots and the R^2 calculated for this model does not consider the TARGET_FLAG_PRED (i.e., we will consider the TARGET_FLAG_PRED=1 always, to plot the residual plot and while computing the model's R^2).

Figure-16: Residual plots of *Model-reg-1*



The residual plot does not have any specific pattern, but clearly it has non-constant variance. The Q-Q plot shows that the errors are not normally distributed. Perhaps applying a log transformation to the TARGET_AMT might make the errors normally distributed. The R^2 of *Model-reg-1* is approximately 0.075 (which is pretty low). But given that the predicted TARGET_AMT is based on the predicted probability of TARGET_FLAG=1, we believe this is the best possible model we can obtain in the given time frame. In future we would like to evaluate if we can improve the predictions using non-parametric methods such as neural networks, Random forests.

Conclusion

In summary we will use the following model to predict the P(TARGET_FLAG=1):

$$P(TARGET\_FLAG = 1) = \frac{e^{f(x)}}{1 + e^{f(x)}}$$

where

$$f(x) = -2.9873679 + 0.3840843KIDSDRIV + 2.0853961NA\_AGE+$$
$$0.0536735HOMEKIDS + -0.0119663YOJ - 0.0000049INCOME$$
$$-0.2864266NA\_HOME\_VAL + 0.014592TRAVTIME - 0.0000243BLUEBOOK$$
$$-0.0555523TIF - 0.0000139OLDCLAIM + 0.1976401CLM\_FREQ$$
$$0.1121727MVR\_PTS - 0.3359639DUMMY\_HOME\_OWNER - 0.467941DUMMY\_MSTATUS+$$
$$0.3651455DUMMY\_PARENT1 + 0.3579248DUMMY\_NO\_HS + 0.3929838DUMMY\_HS+$$
$$0.2569926DUMMY\_Clerical + -0.4239305DUMMY\_Doctor - 0.6878651DUMMY\_Manager+$$
$$0.1749694DUMMY\_Blue\_Collar+2.394212DUMMY\_URBANICITY+0.6983876DUMMY\_CAR\_USE$$
$$-0.6897985DUMMY\_MINI\_VAN+-0.1189946DUMMY\_Pickup+0.2805493DUMMY\_Sports\_Car+$$
$$0.8862538DUMMY\_REVOKED$$

Using the threshold value of 0.28, we will classify TARGET_FLAG as 1 whenever the P(TARGET_CLAG) is greater than or equal to 0.28, else TARGET_FLAG is predicted as 0. Once the TARGET_FLAG is determined, we will use the following model to predict the TARGET_AMT:

$$TARGET\_AMT = TARGET\_FLAG\_PRED(-413.148+0.029125BLUEBOOK+46.757773MVR\_PTS$$
$$-17.680651CAR\_AGE+170.434198DUMMY\_SEX-187.786190DUMMY\_HS-300.711046DUMMY\_REVOKED+$$
$$5835.0180201PROB)$$

where PROB=P(TARGET_FLAG=1)

Using the above two models, we predicted the TARGET_FLAG and TARGET_AMT for the test data set, and the data set is submitted along with this project report for evaluation.

Appendix-A

The summary information of the modified training data set is displayed in the below figure (Figure A.1)

Figure-A.1: Summary of all the varibles in the modified training data set

```
##   TARGET_FLAG       TARGET_AMT        KIDSDRIV           AGE
##  Min.   :0.0000   Min.   :     0   Min.   :0.0000   Min.   :16.00
##  1st Qu.:0.0000   1st Qu.:     0   1st Qu.:0.0000   1st Qu.:39.00
##  Median :0.0000   Median :     0   Median :0.0000   Median :45.00
##  Mean   :0.2637   Mean   :  1504   Mean   :0.1711   Mean   :44.79
##  3rd Qu.:1.0000   3rd Qu.:  1036   3rd Qu.:0.0000   3rd Qu.:51.00
##  Max.   :1.0000   Max.   :107586   Max.   :4.0000   Max.   :81.00
##     HOMEKIDS            YOJ            INCOME          HOME_VAL
##  Min.   :0.0000   Min.   : 0.00   Min.   :     0   Min.   :     0
##  1st Qu.:0.0000   1st Qu.: 9.00   1st Qu.: 29706   1st Qu.:     0
##  Median :0.0000   Median :11.00   Median : 53529   Median :151943
##  Mean   :0.7213   Mean   :10.53   Mean   : 61443   Mean   :146054
##  3rd Qu.:1.0000   3rd Qu.:13.00   3rd Qu.: 83307   3rd Qu.:233366
##  Max.   :5.0000   Max.   :23.00   Max.   :367030   Max.   :885282
##     TRAVTIME         BLUEBOOK          TIF            OLDCLAIM
##  Min.   :  5.00   Min.   : 1500   Min.   : 1.000   Min.   :    0
##  1st Qu.: 22.00   1st Qu.: 9280   1st Qu.: 1.000   1st Qu.:    0
##  Median : 33.00   Median :14440   Median : 4.000   Median :    0
##  Mean   : 33.48   Mean   :15710   Mean   : 5.351   Mean   : 4033
##  3rd Qu.: 44.00   3rd Qu.:20850   3rd Qu.: 7.000   3rd Qu.: 4634
##  Max.   :142.00   Max.   :69740   Max.   :25.000   Max.   :57037
##     CLM_FREQ          MVR_PTS          CAR_AGE           NA_AGE
##  Min.   :0.0000   Min.   : 0.000   Min.   : 0.000   Min.   :0.0000000
##  1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.: 4.000   1st Qu.:0.0000000
##  Median :0.0000   Median : 1.000   Median : 8.000   Median :0.0000000
##  Mean   :0.7983   Mean   : 1.696   Mean   : 8.309   Mean   :0.0007353
##  3rd Qu.:2.0000   3rd Qu.: 3.000   3rd Qu.:12.000   3rd Qu.:0.0000000
##  Max.   :5.0000   Max.   :13.000   Max.   :28.000   Max.   :1.0000000
##      NA_YOJ           NA_INCOME         NA_HOME_VAL      DUMMY_HOME_OWNER
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :1.000
##  Mean   :0.05564   Mean   :0.05453   Mean   :0.05686   Mean   :0.662
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.000
##  DUMMY_MSTATUS      DUMMY_SEX     DUMMY_PARENT1     NA_CAR_AGE
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.0000
##  Median :1.0000   Median :0.000   Median :0.000   Median :0.0000
##  Mean   :0.5998   Mean   :0.464   Mean   :0.132   Mean   :0.0625
##  3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:0.000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.000   Max.   :1.000   Max.   :1.0000
##   DUMMY_NO_HS        DUMMY_HS       DUMMY_BACHELOR   DUMMY_MASTERS
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.1474   Mean   :0.2855   Mean   :0.2746   Mean   :0.2032
```

```
## 3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
## Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
## DUMMY_Clerical    DUMMY_Doctor     DUMMY_Home_Maker   DUMMY_Lawyer
## Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
## 1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
## Median :0.0000   Median :0.00000   Median :0.00000   Median :0.0000
## Mean   :0.1558   Mean   :0.03015   Mean   :0.07855   Mean   :0.1023
## 3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
## Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
## DUMMY_Manager    DUMMY_Professional DUMMY_Student     DUMMY_Blue_Collar
## Min.   :0.0000   Min.   :0.0000     Min.   :0.00000   Min.   :0.0000
## 1st Qu.:0.0000   1st Qu.:0.0000     1st Qu.:0.00000   1st Qu.:0.0000
## Median :0.0000   Median :0.0000     Median :0.00000   Median :0.0000
## Mean   :0.1211   Mean   :0.1368     Mean   :0.08725   Mean   :0.2237
## 3rd Qu.:0.0000   3rd Qu.:0.0000     3rd Qu.:0.00000   3rd Qu.:0.0000
## Max.   :1.0000   Max.   :1.0000     Max.   :1.00000   Max.   :1.0000
## DUMMY_URBANICITY DUMMY_CAR_USE    DUMMY_MINI_VAN   DUMMY_Panel_Truck
## Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
## 1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
## Median :1.0000   Median :0.0000   Median :0.0000   Median :0.00000
## Mean   :0.7955   Mean   :0.3712   Mean   :0.2629   Mean   :0.08284
## 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.00000
## Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  DUMMY_Pickup    DUMMY_Sports_Car   DUMMY_Van        DUMMY_RED_CAR
## Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
## 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000
## Median :0.0000   Median :0.0000   Median :0.00000   Median :0.0000
## Mean   :0.1701   Mean   :0.1112   Mean   :0.09191   Mean   :0.2914
## 3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:1.0000
## Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000
## DUMMY_REVOKED         PROB         TARGET_FLAG_PRED
## Min.   :0.0000   Min.   :0.00249   Min.   :0.0000
## 1st Qu.:0.0000   1st Qu.:0.07749   1st Qu.:0.0000
## Median :0.0000   Median :0.20026   Median :0.0000
## Mean   :0.1224   Mean   :0.26373   Mean   :0.3881
## 3rd Qu.:0.0000   3rd Qu.:0.40133   3rd Qu.:1.0000
## Max.   :1.0000   Max.   :0.96668   Max.   :1.0000
```

Appendix-B

The R code to implement and test the models is given below:

```
###*Figure-2: Summary of the training data set*
summary(train_df)
```

```
###*Figure-3: TARGET_FLAG classes proportion*

display_df <- data.frame(table(train_df$TARGET_FLAG))
display_df$perc <- as.character
(round(100*display_df$Freq/sum(display_df$Freq),2))
#display_df$perc <- round(100*display_df$Freq/sum(display_df$Freq),2)
display_df$perc <- paste(display_df$perc,"%",sep="")

names(display_df) <- c("TARGET_FLAG", "Count", "Percentage")

ggplot(data = display_df, aes(x=TARGET_FLAG, y=Count,
                              fill = TARGET_FLAG, label = Percentage)) +
  geom_bar(stat = "identity", position = "dodge",width=0.5) +
  geom_text(position = position_dodge(width = .9),vjust=0)
```

```
###*Figure-4: Modified training data set's variables*

#Add an indicator variable, to distinguish the data sets
train_df <- cbind(indicator="Train",train_df)
test_df <- cbind(indicator="Test",test_df)

#Combine  the test and train data sets
df <- rbind(train_df,test_df)
#summary(df)

#AGE DUMMY Variable
df$NA_AGE <- df$AGE
df$NA_AGE[!is.na(df$NA_AGE)] <- 0
df$NA_AGE[is.na(df$NA_AGE)] <- 1

df$AGE[is.na(df$AGE)] <- median(df$AGE,na.rm=TRUE)

#YOJ Dummy variable
df$NA_YOJ <- df$YOJ
df$NA_YOJ[!is.na(df$NA_YOJ)] <- 0
df$NA_YOJ[is.na(df$NA_YOJ)] <- 1
df$YOJ[is.na(df$YOJ)] <- median(df$YOJ,na.rm=TRUE)

#Clean INCOME Data
df$INCOME <- gsub(",","",df$INCOME)
df$INCOME <- as.numeric(gsub("$","",df$INCOME,fixed = TRUE))

#Income data has NA values, so create NA_INCOME variable also
df$NA_INCOME <- ifelse(is.na(df$INCOME),1,0)

df$INCOME[is.na(df$INCOME)] <- median(df$INCOME,na.rm=TRUE)
```

```r
df$HOME_VAL <- gsub(",","",df$HOME_VAL)

df$HOME_VAL <- as.numeric(gsub("$","",df$HOME_VAL,fixed = TRUE))

df$NA_HOME_VAL <- ifelse(is.na(df$HOME_VAL),1,0)



#df$HOME_VAL[is.na(df$HOME_VAL)] <- median(df$HOME_VAL,na.rm=TRUE)
df$HOME_VAL[is.na(df$HOME_VAL)] <- 0

df$DUMMY_HOME_OWNER <- ifelse(df$HOME_VAL> 0,1,0)

df$BLUEBOOK <- gsub(",","",df$BLUEBOOK)
df$BLUEBOOK <- as.numeric(gsub("$","",df$BLUEBOOK,fixed = TRUE))


df$OLDCLAIM  <- gsub(",","",df$OLDCLAIM )
df$OLDCLAIM  <- as.numeric(gsub("$","",df$OLDCLAIM,fixed = TRUE))

df$DUMMY_MSTATUS <- ifelse(df$MSTATUS=="z_No",0,1)

df$DUMMY_SEX <- ifelse(df$SEX=="z_F",0,1)

df$DUMMY_PARENT1 <- ifelse(df$PARENT1=="No",0,1)

df$NA_CAR_AGE <- df$CAR_AGE
df$NA_CAR_AGE[!is.na(df$NA_CAR_AGE)] <- 0
df$NA_CAR_AGE[is.na(df$NA_CAR_AGE)] <- 1
df$CAR_AGE[is.na(df$CAR_AGE)] <- median(df$CAR_AGE,na.rm=TRUE)
df <- df[df$CAR_AGE>=0,]
#df$CAR_AGE[df$CAR_AGE < 0] <- -1*df$CAR_AGE[df$CAR_AGE < 0]



df$DUMMY_NO_HS <- ifelse(df$EDUCATION== "<High School",1,0)
df$DUMMY_HS <- ifelse(df$EDUCATION== "z_High School",1,0)
df$DUMMY_BACHELOR <- ifelse(df$EDUCATION== "Bachelors",1,0)
df$DUMMY_MASTERS <- ifelse(df$EDUCATION== "Masters",1,0)

df$DUMMY_Clerical <- ifelse(df$JOB== "Clerical",1,0)
df$DUMMY_Doctor <- ifelse(df$JOB== "Doctor",1,0)
df$DUMMY_Home_Maker <- ifelse(df$JOB== "Home Maker",1,0)
df$DUMMY_Lawyer <- ifelse(df$JOB== "Lawyer",1,0)
df$DUMMY_Manager <- ifelse(df$JOB== "Manager",1,0)
df$DUMMY_Professional <- ifelse(df$JOB== "Professional",1,0)
df$DUMMY_Student <- ifelse(df$JOB== "Student",1,0)
df$DUMMY_Blue_Collar <- ifelse(df$JOB== "z_Blue Collar",1,0)

df$DUMMY_URBANICITY <- ifelse(df$URBANICITY == "Highly Urban/ Urban",1,0)
```

```
df$DUMMY_CAR_USE <- ifelse(df$CAR_USE == "Commercial",1,0)


df$DUMMY_MINI_VAN <- ifelse(df$CAR_TYPE == "Minivan",1,0)

df$DUMMY_Panel_Truck <- ifelse(df$CAR_TYPE == "Panel Truck",1,0)
df$DUMMY_Pickup <- ifelse(df$CAR_TYPE == "Pickup",1,0)
df$DUMMY_Sports_Car <- ifelse(df$CAR_TYPE == "Sports Car",1,0)
df$DUMMY_Van <- ifelse(df$CAR_TYPE == "Van",1,0)

df$DUMMY_RED_CAR <- ifelse(df$RED_CAR == "yes",1,0)
df$DUMMY_REVOKED <- ifelse(df$REVOKED == "Yes",1,0)

#summary(df)
#head(df)

train_df <- df[df$indicator == "Train",c(-1)]
test_df <- df[df$indicator == "Test",c(-1)]

#head(train_df)

#names(train_df)

#Let us prepare another data frame to build models
train_df_mod <- train_df[,c("TARGET_FLAG",
"TARGET_AMT",
"KIDSDRIV",
"AGE",
"HOMEKIDS",
"YOJ",
"INCOME",
"HOME_VAL",
"TRAVTIME",
"BLUEBOOK",
"TIF",
"OLDCLAIM",
"CLM_FREQ",
"MVR_PTS",
"CAR_AGE",
"NA_AGE",
"NA_YOJ",
"NA_INCOME",
"NA_HOME_VAL",
"DUMMY_HOME_OWNER",
"DUMMY_MSTATUS",
"DUMMY_SEX",
"DUMMY_PARENT1",
"NA_CAR_AGE",
"DUMMY_NO_HS",
"DUMMY_HS",
"DUMMY_BACHELOR",
"DUMMY_MASTERS",
"DUMMY_Clerical",
```

```r
"DUMMY_Doctor",
"DUMMY_Home_Maker",
"DUMMY_Lawyer",
"DUMMY_Manager",
"DUMMY_Professional",
"DUMMY_Student",
"DUMMY_Blue_Collar",
"DUMMY_URBANICITY",
"DUMMY_CAR_USE",
"DUMMY_MINI_VAN",
"DUMMY_Panel_Truck",
"DUMMY_Pickup",
"DUMMY_Sports_Car",
"DUMMY_Van",
"DUMMY_RED_CAR",
"DUMMY_REVOKED"
                             )]

#head(train_df_mod)
names(train_df_mod)
```

```r
glm.fit1 <- glm(data=train_df_mod[,-2],TARGET_FLAG~KIDSDRIV+
AGE*NA_AGE+
HOMEKIDS+
YOJ*NA_YOJ+
INCOME*NA_INCOME+
HOME_VAL*NA_HOME_VAL+
TRAVTIME+
BLUEBOOK+
TIF+
OLDCLAIM+
CLM_FREQ+
MVR_PTS+
CAR_AGE*NA_CAR_AGE+
DUMMY_HOME_OWNER*NA_HOME_VAL+
DUMMY_MSTATUS+
DUMMY_SEX+
DUMMY_PARENT1+
DUMMY_NO_HS+
DUMMY_HS+
DUMMY_BACHELOR+
DUMMY_MASTERS+
DUMMY_Clerical+
DUMMY_Doctor+
DUMMY_Home_Maker+
DUMMY_Lawyer+
DUMMY_Manager+
DUMMY_Professional+
DUMMY_Student+
DUMMY_Blue_Collar+
DUMMY_URBANICITY +
DUMMY_CAR_USE+
DUMMY_MINI_VAN+
```

```
DUMMY_Panel_Truck+
DUMMY_Pickup+
DUMMY_Sports_Car+
DUMMY_Van+
DUMMY_RED_CAR+
DUMMY_REVOKED,family="binomial")

#names(summary(glm.fit1))
#stepAIC(glm.fit1)
display_df <- data.frame(summary(glm.fit1)$coefficients)
display_df <- display_df[,-3]
Variable <- rownames(display_df)
display_df <- cbind(Variable,display_df)
rownames(display_df) <- NULL
names(display_df) <- c("Variable","Coefficient","Std_Error","P_value")
kable(display_df)
```

```
###_Figure-6: Model-1 (Logistic regression) variables,
##which have p-values greater than 0.05_

display_df <- display_df[display_df$P_value<= 0.05,]
rownames(display_df) <- NULL
kable(display_df)
```

```
###_Figure-7: Model-1, built using significant variables only_

glm.fit1 <- glm(formula = TARGET_FLAG ~ KIDSDRIV + NA_AGE + HOMEKIDS + YOJ +
    INCOME + NA_HOME_VAL + TRAVTIME + BLUEBOOK + TIF + OLDCLAIM +
    CLM_FREQ + MVR_PTS + DUMMY_HOME_OWNER + DUMMY_MSTATUS + DUMMY_PARENT1 +
    DUMMY_NO_HS + DUMMY_HS + DUMMY_Clerical + DUMMY_Doctor +
    DUMMY_Manager + DUMMY_Blue_Collar + DUMMY_URBANICITY + DUMMY_CAR_USE +
    DUMMY_MINI_VAN + DUMMY_Pickup + DUMMY_Sports_Car + DUMMY_REVOKED,
    family = "binomial", data = train_df_mod[, -2])

#summary(glm.fit1)
display_df <- data.frame(summary(glm.fit1)$coefficients)
display_df <- display_df[,-3]
Variable <- rownames(display_df)
display_df <- cbind(Variable,display_df)
rownames(display_df) <- NULL
names(display_df) <- c("Variable","Coefficient","Std_Error","P_value")
kable(display_df)
```

```
###_Figure-8: Model-2, built using Model-1 variables raised to the power of 2_

glm.fit2 <- glm(formula = TARGET_FLAG ~ poly(KIDSDRIV,2) +
                  NA_AGE + poly(HOMEKIDS,2) + poly(YOJ,2) +
    poly(INCOME,2) + NA_HOME_VAL + poly(TRAVTIME,2) +
      poly(BLUEBOOK,2) + poly(TIF,2) + poly(OLDCLAIM,2) +
    poly(CLM_FREQ,2) + poly(MVR_PTS,2) + DUMMY_HOME_OWNER +
      DUMMY_MSTATUS + DUMMY_PARENT1 +
    DUMMY_NO_HS + DUMMY_HS + DUMMY_Clerical + DUMMY_Doctor +
    DUMMY_Manager + DUMMY_Blue_Collar + DUMMY_URBANICITY +
```

```
      DUMMY_CAR_USE +
    DUMMY_MINI_VAN + DUMMY_Pickup + DUMMY_Sports_Car +
      DUMMY_REVOKED,
    family = "binomial", data = train_df_mod[, -2])

#summary(glm.fit2)
#stepAIC(glm.fit2)

glm.fit2 <- glm(formula = TARGET_FLAG ~
                  poly(KIDSDRIV, 2) + NA_AGE + poly(YOJ,
    2) + poly(INCOME, 2) + NA_HOME_VAL +
      poly(TRAVTIME, 2) +
    poly(BLUEBOOK, 2) + poly(TIF, 2) + poly(OLDCLAIM, 2) +
      poly(CLM_FREQ,
    2) + poly(MVR_PTS, 2) + DUMMY_HOME_OWNER +
      DUMMY_MSTATUS +
    DUMMY_PARENT1 + DUMMY_NO_HS + DUMMY_HS +
      DUMMY_Clerical +
    DUMMY_Doctor + DUMMY_Manager + DUMMY_Blue_Collar +
      DUMMY_URBANICITY +
    DUMMY_CAR_USE + DUMMY_MINI_VAN + DUMMY_Pickup +
      DUMMY_Sports_Car +
    DUMMY_REVOKED, family = "binomial", data = train_df_mod[,
    -2])

display_df <- data.frame(summary(glm.fit2)$coefficients)
display_df <- display_df[,-3]
Variable <- rownames(display_df)
display_df <- cbind(Variable,display_df)
rownames(display_df) <- NULL
names(display_df) <- c("Variable","Coefficient","Std_Error","P_value")
kable(display_df)
```

```
###_Figure-8: Finding the optimal value of K for KNN algorithm_
set.seed(123)

knn_test_ind <- sample(1:8160,round(.2*8160))

knn_test <- train_df_mod[knn_test_ind,]

knn_train <- train_df_mod[-knn_test_ind,]

knn_test_actual <- knn_test$TARGET_FLAG
knn_train_actual <- knn_train$TARGET_FLAG

knn_test_actual <- as.factor(knn_test_actual)
knn_train_actual <- as.factor(knn_train_actual)

knn_test <- scale(train_df_mod[knn_test_ind,c(-1,-2)])
knn_train <- scale(train_df_mod[-knn_test_ind,c(-1,-2)])

error <- vector(length=30)
```

```
for(i in 1:30)
  {
     k <- knn(knn_train,knn_test,knn_train_actual,k=i)

     error[i] <- mean(k!=knn_test_actual)
}

display_df <- data.frame(k=1:30,error=error)


ggplot(data=display_df,aes(x=k,y=error))+
  geom_point(size=3)+
  geom_line()
```

```
###Figure-9: ROC curve for _Model-1_

#actual <- train_df_mod$TARGET_FLAG
prob <- predict(glm.fit1,type="response")
#predicted <- ifelse(prob>=0.28,1,0)
#conf_matrix <- table(predicted,actual)
#conf_matrix

#confusionMatrix(conf_matrix,positive = "1")

roc_obj = roc(response=train_df_mod$TARGET_FLAG,predictor=prob,
levels=rev(levels(as.factor(train_df_mod$TARGET_FLAG))))
plot.roc(roc_obj)
```

```
###Figure-10: ROC curve for _Model-2_

#actual <- train_df_mod$TARGET_FLAG
prob <- predict(glm.fit2,type="response")
#predicted <- ifelse(prob>=0.28,1,0)
#conf_matrix <- table(predicted,actual)
#conf_matrix

#confusionMatrix(conf_matrix,positive = "1")

roc_obj = roc(response=train_df_mod$TARGET_FLAG,predictor=prob,
levels=rev(levels(as.factor(train_df_mod$TARGET_FLAG))))
plot.roc(roc_obj)
```

```
knn_train <- train_df_mod
actual <- knn_train$TARGET_FLAG
knn_train <- scale(knn_train[,c(-1,-2)])
prob <- knn(knn_train,knn_train,actual,k=20,prob=TRUE)

prob <- attributes(.Last.value)$prob

#prob <- as.vector(prob)

roc_obj = roc(response=actual,predictor=prob,
levels=rev(levels(as.factor(actual))))
```

```
plot.roc(roc_obj)
```

```
###Figure-12: _Model-1_ performance details with threshold=0.5

actual <- train_df_mod$TARGET_FLAG
prob <- predict(glm.fit1,type="response")
predicted <- ifelse(prob>=0.5,1,0)
conf_matrix <- table(predicted,actual)
#conf_matrix
confusionMatrix(conf_matrix,positive = "1")
```

```
###Figure-13: Threshold vs sensitivity and accuracy plot
threshold <- seq(from=0.01,to=.99,by=.01)

actual <- train_df_mod$TARGET_FLAG
prob <- predict(glm.fit1,type="response")
acc <- vector()
sens <- vector()
for(i in 1:length(threshold))
{

predicted <- ifelse(prob>=threshold[i],1,0)
conf_matrix <- table(predicted,actual)
if(nrow(conf_matrix) == 1) break()

cnf <- confusionMatrix(conf_matrix,positive = "1")
#names(cnf)
acc[i] <- cnf$overall["Accuracy"]
sens[i] <- cnf$byClass["Sensitivity"]

}

display_df <- data.frame(Legend="accuracy",
                         value=acc[1:length(threshold)],threshold=threshold)
display_df <- rbind(display_df,data.frame(Legend="sensitivity",
                                          value=sens[1:length(threshold)],threshold=threshold))

ggplot(data=display_df,aes(x=threshold,y=value,color=Legend))+
  geom_point()+
  geom_vline(xintercept = .28,colour="blue", linetype = "longdash")+
  annotate("text",label="Optimal Threshold=0.28", x = .4,
           y = .9, size = 3, colour = "blue")+
  labs(title="Sensitivity vs Accuracy plot",x="Threshold",
       y="Sensitivity and Accuracy")
```

```
###Figure-14: _Model-1_ performance details with threshold=0.28

actual <- train_df_mod$TARGET_FLAG
prob <- predict(glm.fit1,type="response")
predicted <- ifelse(prob>=0.28,1,0)
conf_matrix <- table(predicted,actual)
#conf_matrix
confusionMatrix(conf_matrix,positive = "1")
```

```r
train_df_mod$PROB <- predict(glm.fit1,type="response")

train_df_mod$TARGET_FLAG_PRED <- ifelse(train_df_mod$PROB>=0.28,1,0)

glm.reg.fit1 <- glm(data = train_df_mod[,-1],TARGET_AMT~.)
#stepAIC(glm.reg.fit1)
glm.reg.fit1 <- glm(formula = TARGET_AMT ~ BLUEBOOK + MVR_PTS + CAR_AGE + DUMMY_SEX +
    DUMMY_HS + DUMMY_REVOKED + PROB, data = train_df_mod[, -1])

display_df <- data.frame(summary(glm.reg.fit1)$coefficients)
display_df <- display_df[,-3]
Variable <- rownames(display_df)
display_df <- cbind(Variable,display_df)
rownames(display_df) <- NULL

names(display_df) <- c("Variable","Coefficient","Std_Error","P_value")
#names(display_df) <- c("Coefficient","Std_Error","P_value")
kable(display_df)
```

```r
###Figure-16: Residual plots of _Model-reg-1_

par(mfrow=c(2,2))
plot(glm.reg.fit1)
```

```r
#glm.fit1
prob <- predict(glm.fit1,test_df,type="response")
test_df$PROB <- predict(glm.fit1,test_df,type="response")
test_df$TARGET_FLAG <- ifelse(prob >= 0.28, 1, 0)
test_df$TARGET_FLAG_PRED <- ifelse(prob >= 0.28, 1, 0)
test_df$TARGET_AMT <- test_df$TARGET_FLAG_PRED * (predict(glm.reg.fit1,test_df))
#head(test_df)
write_test_df <- read.csv("insurance-evaluation-data.csv")
write_test_df$TARGET_FLAG <- test_df$TARGET_FLAG
write_test_df$TARGET_AMT <- test_df$TARGET_AMT
write.csv(write_test_df,file="test_result.csv",row.names = FALSE)
```

```r
###Figure-A.1: Summary of all the varibles in the modified training data set

summary(train_df_mod)
```