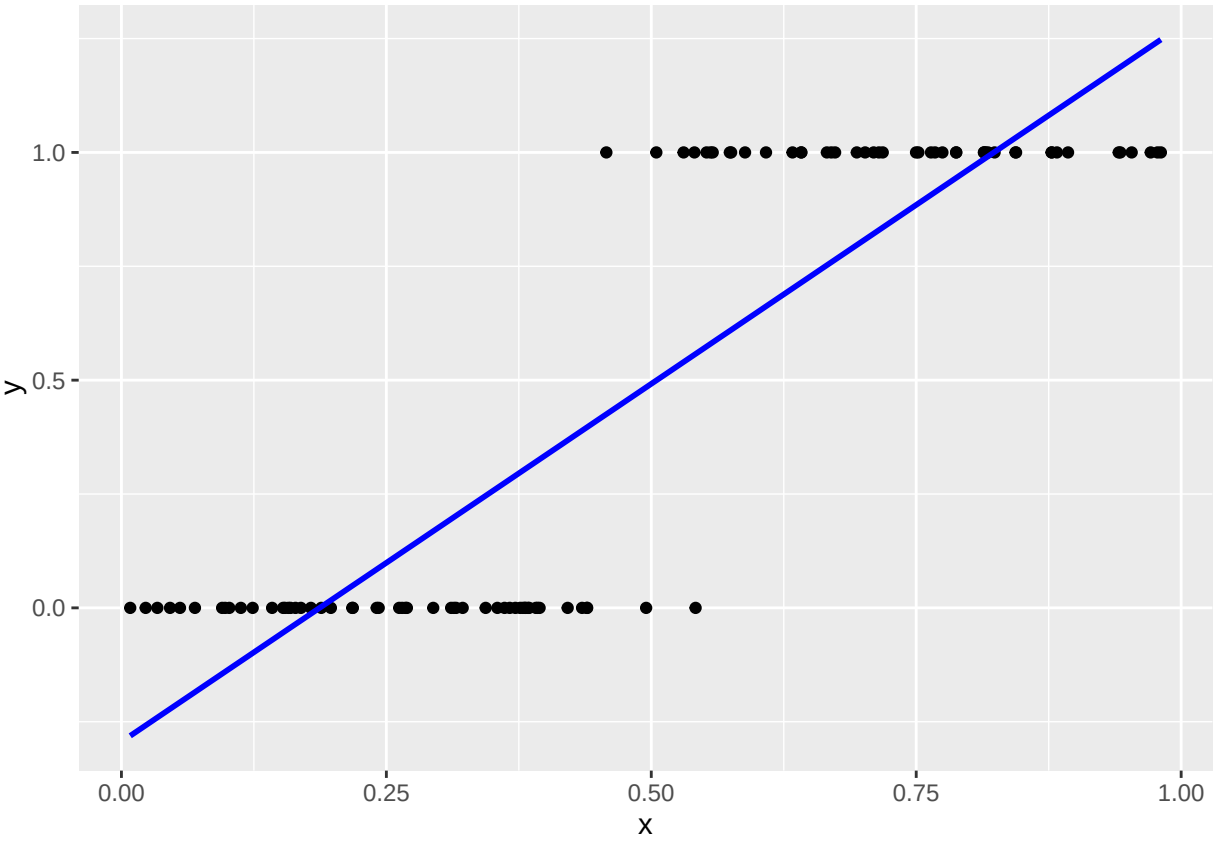
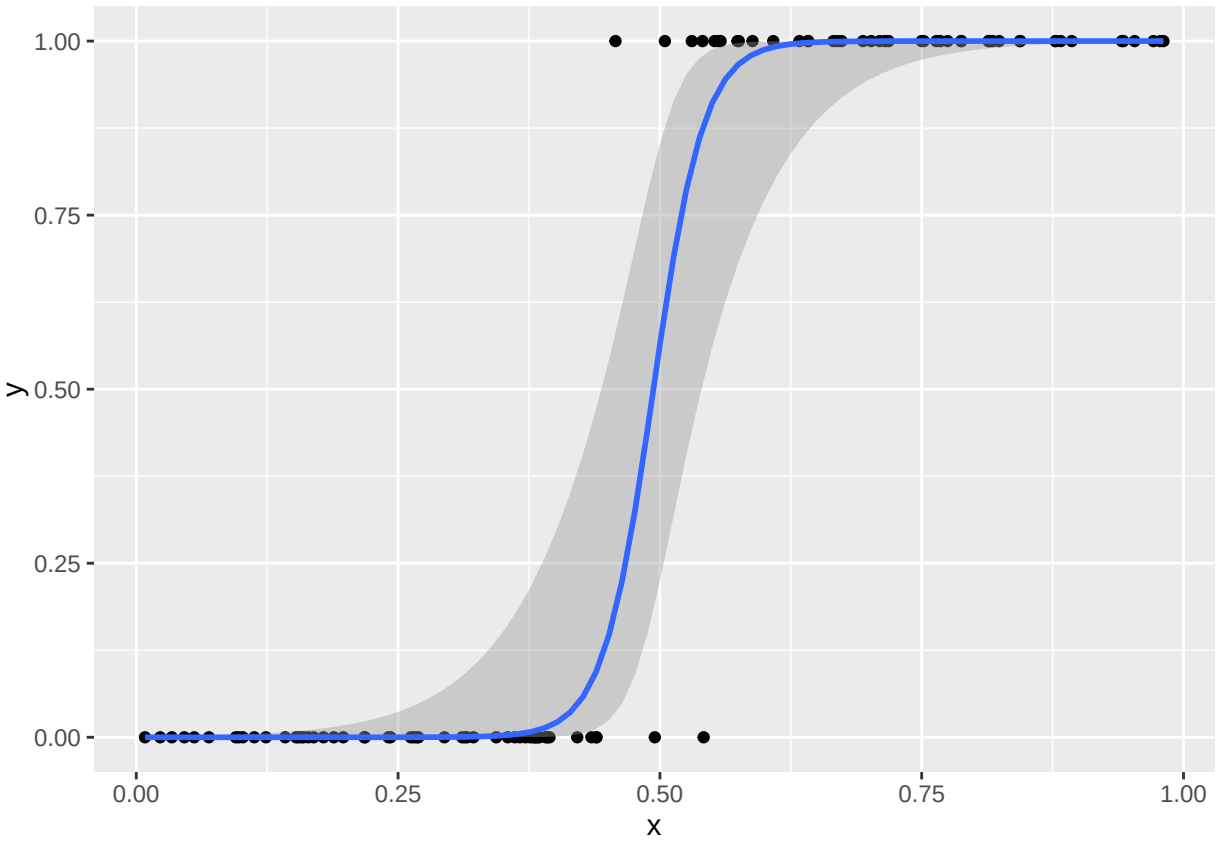
Logistic Regression

You have seen how to model a continuous numeric response with linear regression technique. But in many business scenarios our target is binary. For example whether someone will buy my product , whether someone will default on the loan they have taken. Answer to all these and many other such questions is yes/no. We can convert that 1/0 and then try to model them with linear regression technique but that doesn't result in good results. Have a look.

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```



You can see how badly this fails. what we really need is something like this.



Although to reach there we need to first develop some understanding regarding this. So far we have seen that linear regression approach fails at many levels. Lets look at these kind of problems from a different perspective. Consider this data regarding a hypothetical situation where we asked children of various age whether they are afraid of ghosts or not. Here are the results:

Age	Response
4	yes
4	yes
4	yes
4	yes
4	no
5	yes
5	yes
5	yes
5	no
5	no
6	yes
6	yes
6	no
6	no
6	no
7	yes
7	no
7	no
7	no
7	no

Now if someone asked you what might the response be if the child's age is 7. By looking at the table your guess would be "no". What you did there was to look at probability of response being "no" when age is 7.And you naturally guessed for "no" because that had higher probability *[chances]*

So instead of modeling y we should model P(y="yes") or P(y="no"). lets denote that by just p. Instead of $y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 .... \beta_p * x_p$. we'd model this:

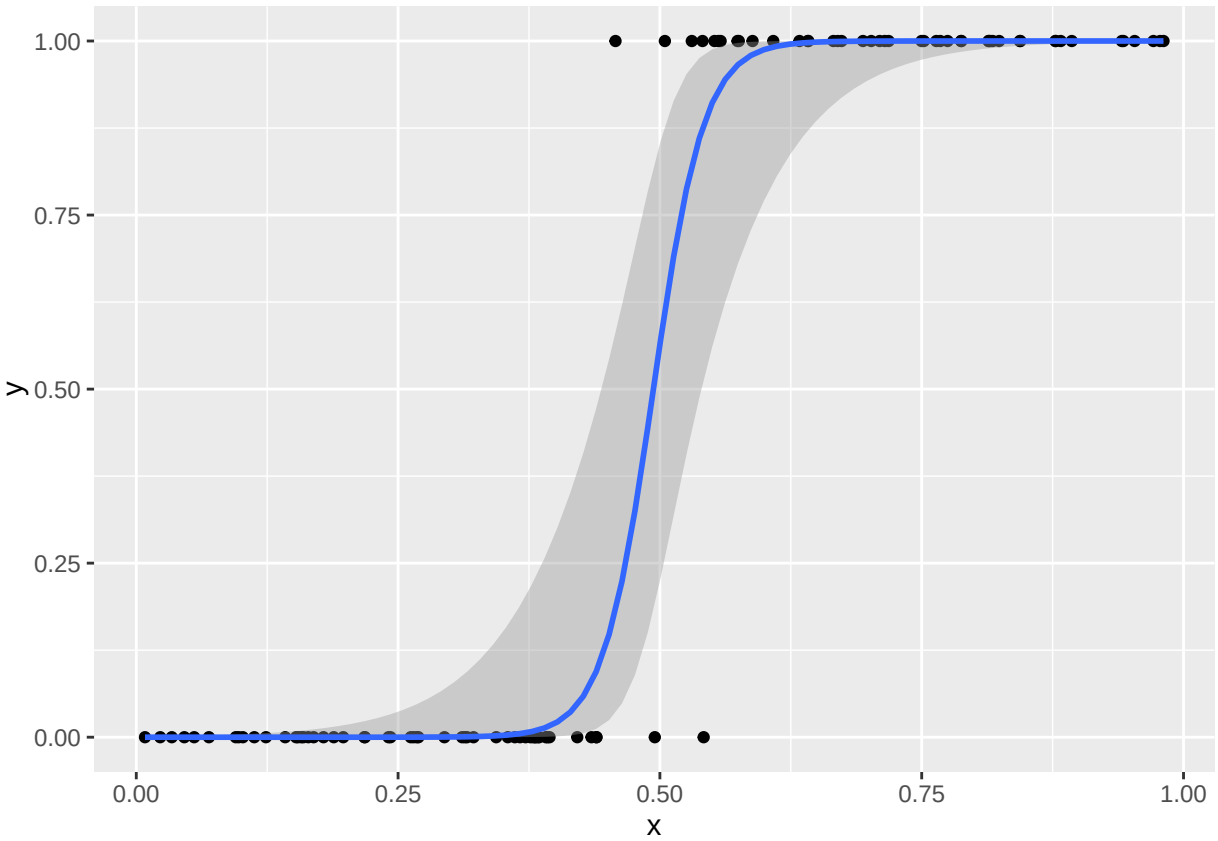
$$p = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 .... \beta_p * x_p$$

but this is problematic because right hand side in equation above can take values in the interval $(-\infty, +\infty)$ .where as probability p can take values in [0,1]. We need to transform this so that ranges match on both sides.

$$log(\frac{p}{1-p}) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 .... \beta_p * x_p$$

This takes care of the range mismatch issue. This transformation also provides one important characteristic to the probability p which you'll get as result. Remember in our hypothetical example we said you looked at high probability outcome and decided that'd be your prediction. Now imagine this to be at a large scale. Your prediction will still be binary, but you'll have to come up with a cutoff for this "p", so that above that cutoff your prediction will be 1 and below that your prediction will be 0.

You'd like this cutoff to be such that it enables you to make as least as possible miss-classifications. For this to happen , your "p" should be consistently high for one class and low for another , so that you can choose a nice cutoff for "p" in between which very well devides both the classes. Look at this curve again. Due to transformation mentioned above, we get our probabilities p to lie on this curve which enables us to get a good cutoff.



Now lets talk about those parameter estimation. The objective here is not to "correctly" estimate $log(\frac{p}{1-p})$. No. We want our parameters to take values which result in such a score *[probabilities or p]* which enables us to have a good cutoff. Meaning this "score" should be high for one class and low for another.

Lets say $P(y_i = 1|x_i) = M_i$

Now consider this :

$$L_i = M_i^{y_i} * (1 - M_i)^{1-y_i}$$

whenever $y_i = 1 : L_i = M_i$ and when $y_i = 0 : L_i = 1 - M_i$. Which means that when $y_i = 1$, L_i equals to probability of y being 1, when $y_i = 0$, L_i equals to probability of y being 0. You'd want your probabilities to match with real outcome. In other words you'd like to maximise L_i. Again we'd maximise a collective form of these L_i.

$$L = \prod_{i=1}^{n} L_i$$

Corresponding parameter estimates for this maximization are obtained through numerical methods, which we'll not discuss because of mathematical complexity involved.

Parameter Interpretation

Once we have our model we can write probabilites as this

$$\frac{p}{(1-p)} = e^{\beta_0} * e^{\beta_1 * x_1} * e^{\beta_2 * x_2} ..... * e^{\beta_p * x_p}$$

this $\frac{p}{(1-p)}$ is nothing but odds in favour of y=1 *[or y=0]*. Now we can say that when x_i goes up by one unit, odds change by e^{β_i}. If β_i is +ve , odds go up and if its -ve, odds go down.

Miss-Classification Metrices and Other Things

We'll understand how to get the cutoff out from our probability score with the help of different metrices. First two of which relate directly to miss-classification.

We understand that no ultimate model and cutoff is going to result in perfect predictions. There are going to be cases where you predicted class to be 1 and in realty it'll be 0 and vice versa. We can make a cross table for predicted and real results like this:

\\ | Predicted 1 | Predicted 0

———|————-|———— Real 1 | TP | FN Real 0 | FP | TN

Here TP = True Positive , Count of cases where real outcome was 1 and prediction was also 1 FP = False Postive , Count of cases where real outcome was 0 but prediction was 1 TN = True Negative , Count of cases where real outcome was 0 and prediction was also 0 FN = False Negative, Count of case Where real outcome was 1 but prediction was 0 T = TP + FN = Count of all cases where real outcome was 1 N = TN + FP = Count of all cases where real outcome was 0

On the basis of these counts we can define follwoing metrices

$$Accuracy = \frac{(TP + TN)}{(T + N)}$$

$$Miss - ClassificationError = \frac{(FP + FN)}{(T + N)} = 1 - Accuracy$$

Two other important measures are sensitivity and specificity. Which are defined as follows:

Sensitivity = Ability of the model to capture all positives.

$$Sensitivity = \frac{TP}{(TP + FN)}$$

Specificity = Ability of the model to capture all negatives.

$$Specificity = \frac{TN}{TN + FP}$$

For an ideal model , predictions will be perfect. And values of Accuracy , Sensitivity and Specificity will all be 1 where as Miss-Classification Error will be 0. In practical scenarios You'd like to Your Sensitivity and Specificity as close to 1 as Possible.

ROC Curve

We can consider many cutoffs across our score range of 0 to 1 and calculate Sensitivity and Specificity For all these cutoffs. When we plot these pairs of (Sensitivity , 1-Specificity) , the curve obtained is called ROC curve. The point which is closest to upper left corner *[which corresponds to Sensitivity=1 & Specificity=1]* is chosen as the cutoff.

We'll look at how to plot ROC curve with a prepared model. Once we do please refer to following note.

Note: You can imagine that higher your curve goes w.r.t. to straight line in the middle, better your cutoff will be *[Closer to upper left corner]* . Or in other words higher is the Area Under Curve *[aka AUC]* better will be your model. This AUC can be used to compare multiple models.

Lift

We'll try to understand the concept of lift from a marketing campaign perspective. Lets say you did not have any model to pre check whether which one of your prospective 10000 customers are going to respond to your email campaign. You dropped mail to all 10000 and 3000 of those responded. This means , if you randomly send mails to say 1000 people 30% of that will respond. Now with your model in hand you are able to predict before hand whether someone will respond to it or not. Of course your prediction is not perfect and each "bin" has some miss-classification associated with it.

Consider you make 10 bins out of 10000 people where 1 correspond to responders and 0 corresponds to non responders. If you dropped mails to every one who falls in bin 1 according to the score. you'll be able to capture say 30% of the responders by dropping mail to only 10% of the population. if you dropped mail to both bin1 and bin2 people, you'll be able to capture say 65% of the responder by dropping mail to 20% of the population. These percentages of responders that you are able to capture are called gains. ratio between gains and the population percentage that you have to drop mail to is called lift.

Here in the context of the campign you can decide your cutoff on the basis of how much lift you want for your next campaign.

Case Study

A financial institution is planning to roll out a stock market trading faciliation service for their existing account holders. This service costs significant amount of money for the bank in terms of infra, licensing and people cost. To make the serive offering profitable, they charge a percentage base comission on every trade transaction. However this is not a unique service offered by them, many of their other competitors are offering the same service and at lesser commission some times. To retain or attract people who trade heavily

on stock market and in turn generate a good commission for institution, they are planning to offer discounts as they roll out the service to entire customer base.

Problem is , that this discount, hampers profits coming from the customers who do not trade in large quantities . To tackle this issue , company wants to offer discounts selectively. To be able to do so, they need to know which of their customers are going to be heavy traders or money makers for them.

To be able to do this, they decided to do a beta run of their service to a small chunk of their customer base [approx 10000 people]. For these customers they have manually divided them into two revenue categories 1 and 2. Revenue one category is the one which are moeny makers for the bank, revenue category 2 are the ones which need to be kept out of discount offers.

We need to use this study's data to build a prediction model which should be able to identify if a customer is potentially eligible for discounts [falls In revnue grid category 1]. Lets get the data and begin.

```
rg=read.csv("Existing Base.csv",stringsAsFactors = FALSE)
library(dplyr)
glimpse(rg)
```

```
## Observations: 10,155
## Variables: 32
## $ REF_NO                           (int) 1, 2, 3, 5, 6, 7, 8, 9, 10, 11...
## $ children                         (chr) "Zero", "Zero", "Zero", "Zero"...
## $ age_band                         (chr) "51-55", "55-60", "26-30", "18...
## $ status                           (chr) "Partner", "Single/Never Marri...
## $ occupation                       (chr) "Manual Worker", "Retired", "P...
## $ occupation_partner               (chr) "Secretarial/Admin", "Retired"...
## $ home_status                      (chr) "Own Home", "Own Home", "Own H...
## $ family_income                    (chr) "<17,500, >=15,000", "<27,500,...
## $ self_employed                    (chr) "No", "No", "Yes", "No", "No",...
## $ self_employed_partner            (chr) "No", "No", "No", "No", "No", ...
## $ year_last_moved                  (int) 1972, 1998, 1996, 1997, 1995, ...
## $ TVarea                           (chr) "HTV", "Granada", "Tyne Tees",...
## $ post_code                        (chr) "NP4 9HS", "M41 0QH", "NE30 1J...
## $ post_area                        (chr) "NP4", "M41", "NE30", "HR9", "...
## $ Average.Credit.Card.Transaction (dbl) 148.44, 0.00, 0.00, 0.00, 73.4...
## $ Balance.Transfer                 (dbl) 142.95, 74.98, 166.44, 0.00, 5...
## $ Term.Deposit                     (dbl) 0.00, 0.00, 20.99, 0.00, 0.00,...
## $ Life.Insurance                   (dbl) 81.96, 25.99, 291.37, 20.49, 1...
## $ Medical.Insurance                (dbl) 0.00, 0.00, 11.48, 0.00, 41.95...
## $ Average.A.C.Balance              (dbl) 29.99, 0.00, 166.94, 39.46, 39...
## $ Personal.Loan                    (dbl) 0.00, 0.00, 0.00, 0.00, 10.97,...
## $ Investment.in.Mutual.Fund        (dbl) 61.95, 0.00, 15.99, 45.44, 212...
## $ Investment.Tax.Saving.Bond       (dbl) 19.99, 0.00, 0.00, 0.00, 0.00,...
## $ Home.Loan                        (dbl) 0.00, 0.00, 3.49, 0.00, 45.91,...
## $ Online.Purchase.Amount           (dbl) 0.00, 0.00, 0.00, 0.00, 25.98,...
## $ Revenue.Grid                     (int) 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, ...
## $ gender                           (chr) "Female", "Female", "Male", "F...
## $ region                           (chr) "Wales", "North West", "North"...
## $ Investment.in.Commudity          (dbl) 74.67, 20.19, 98.06, 4.10, 70....
## $ Investment.in.Equity             (dbl) 18.66, 0.00, 31.07, 14.15, 55....
## $ Investment.in.Derivative         (dbl) 32.32, 4.33, 80.96, 17.57, 80....
## $ Portfolio.Balance                (dbl) 89.43, 22.78, 171.78, -41.70, ...
```

As we saw in linear regression , a good amount of the modelling process will go into getting our data ready for eventual statistical operations. Lets start with looking at our first predictor variable in the data which is "children".

```
table(rg$children)
```

```
##
##    1    2    3   4+ Zero
## 1848 1607  473   19 6208
```

We can easily convert this, to numeric data without any concern.

```
rg = rg %>%
  mutate(children=ifelse(children=="Zero",0,substr(children,1,1)),
         children=as.numeric(children))
```

Lets look at age band variable , we can possibly convert this to numeric by taking average of age ranges. Lets look at the frequency table any way to find if there are any non-numeric fields.

```
table(rg$age_band)
```

```
##
##   18-21   22-25   26-30   31-35   36-40   41-45   45-50   51-55   55-60
##      63     456     927    1061    1134    1112    1359    1052    1047
##   61-65   65-70     71+ Unknown
##     881     598     410      55
```

We can try two iterations here, we can create dummy variables for these categories present. In this case we are loosing some information by not utilising the fact that each range contains some numbers and there is certainly an order to them. Another possiblity is what we discussed earlier, converting these ranges to continuos numeric values by taking average of the range given. Be aware however that we are inflating infromation here. Assuming that variable can take all possible values in the range given.

For our discussion here we are going to try second type, leaving the first option to try to you. Now in that we can either drop the obs where category value is unknown or we can check the response rates acroos categories and see which one is closesto "unknown" and supply that value instead. Lets see:

```
prop.table(table(rg$age_band,rg$Revenue.Grid),1)
```

```
##
##                    1          2
##   18-21   0.17460317 0.82539683
##   22-25   0.10964912 0.89035088
##   26-30   0.10679612 0.89320388
##   31-35   0.10556079 0.89443921
##   36-40   0.12610229 0.87389771
##   41-45   0.11061151 0.88938849
##   45-50   0.10154525 0.89845475
##   51-55   0.09980989 0.90019011
##   55-60   0.11461318 0.88538682
##   61-65   0.09421112 0.90578888
##   65-70   0.09531773 0.90468227
##   71+     0.10243902 0.89756098
##   Unknown 0.05454545 0.94545455
```

It turns out that the behaviour is really different from the rest of the categories , so the option for value imputation to unknown is gone. It would have been an insteresting dummy variable possibly , for now we are going to drop those observations. [All these subjective choices we are making, signify the need for running multiple iterations with different choices and then selecting the path which results in best model].

```
rg=rg %>%
mutate(a1=as.numeric(substr(age_band,1,2)),
```

```
       a2=as.numeric(substr(age_band,4,5)),
      age=ifelse(substr(age_band,1,2)=="71",71,ifelse(age_band=="Unknown",NA,0.5*(a1+a2)))
       ) %>%
  select(-a1,-a2,-age_band) %>%
  na.omit()
```

Next we'll be looking at various categorical variables and taking decision on which dummy variables to create.

```
table(rg$status)
```

```
##
##   Divorced/Separated              Partner Single/Never Married
##                  678                 7692                 1099
##              Unknown              Widowed
##                   17                  614
```

```
rg = rg %>%
  mutate(status_div=as.numeric(status=="Divorced/Separated"),
         status_partner=as.numeric(status=="Partner"),
         status_single=as.numeric(status=="Single/Never Married")) %>%
  select(-status)

table(rg$occupation)
```

```
##
##  Business Manager         Housewife     Manual Worker             Other
##               732              1253               556               537
##      Professional           Retired Secretarial/Admin           Student
##              2436              2198              1796                56
##           Unknown
##               536
```

We can either chose to make n-1 dummy variable here or check if response behaviour is similar across few of these categories and merge them. Lets see:

```
round(prop.table(table(rg$occupation,rg$Revenue.Grid),1),2)
```

```
##
##                         1    2
##   Business Manager  0.12 0.88
##   Housewife         0.09 0.91
##   Manual Worker     0.11 0.89
##   Other             0.11 0.89
##   Professional      0.12 0.88
##   Retired           0.10 0.90
##   Secretarial/Admin 0.11 0.89
##   Student           0.11 0.89
##   Unknown           0.11 0.89
```

```
rg=rg %>%
  mutate(occ_BM_prof=as.numeric(occupation %in% c("Business Manager","Professional")),
         occ_Retired=as.numeric(occupation=="Retired"),
         occ_HW=as.numeric(occupation=="Housewife")) %>%
  select(-occupation)

round(prop.table(table(rg$occupation_partner,rg$Revenue.Grid),1),2)
```

```
##
##                          1    2
##   Business Manager  0.11 0.89
##   Housewife         0.11 0.89
##   Manual Worker     0.11 0.89
##   Other             0.10 0.90
##   Professional      0.11 0.89
##   Retired           0.10 0.90
##   Secretarial/Admin 0.12 0.88
##   Student           0.12 0.88
##   Unknown           0.10 0.90
```

```
rg=rg %>%
  mutate(op_1=as.numeric(occupation_partner %in% c("Other","Retired","Unknown")),
         op_2=as.numeric(occupation_partner %in% c("Student","Secretarial/Admin"))) %>%
  select(-occupation_partner)

table(rg$home_status)
```

```
##
## Live in Parental Hom             Own Home Rent from Council/HA
##                  109                 9390                  321
##       Rent Privately         Unclassified
##                  259                   21
```

```
unique(rg$home_status)
```

```
## [1] "Own Home"             "Rent from Council/HA" "Rent Privately"
## [4] "Live in Parental Hom" "Unclassified"
```

```
rg=rg %>%
  mutate(hs_livein=as.numeric(home_status=="Live in Parental Hom"),
         hs_own=as.numeric(home_status=="Own Home"),
         hs_rent_private=as.numeric(home_status=="Rent Privately"),
         hs_rent_council=as.numeric(home_status=="Rent from Council/HA")) %>%
  select(-home_status)

round(prop.table(table(rg$family_income,rg$Revenue.Grid),1),2)
```

```
##
##                          1    2
##   < 4,000           0.08 0.92
##   < 8,000, >= 4,000 0.08 0.92
##   <10,000, >= 8,000 0.11 0.89
##   <12,500, >=10,000 0.10 0.90
##   <15,000, >=12,500 0.11 0.89
##   <17,500, >=15,000 0.12 0.88
##   <20,000, >=17,500 0.11 0.89
##   <22,500, >=20,000 0.12 0.88
##   <25,000, >=22,500 0.10 0.90
##   <27,500, >=25,000 0.10 0.90
##   <30,000, >=27,500 0.12 0.88
##   >=35,000          0.11 0.89
##   Unknown           0.07 0.93
```

```
rg=rg %>%
```

```
  mutate(fi_1=as.numeric(family_income %in%
              c("< 4,000","< 8,000, >= 4,000")),
         fi_2=as.numeric(family_income %in%
              c("<12,500, >=10,000","<25,000, >=22,500","<27,500, >=25,000")),
         fi_3=as.numeric(family_income %in%
              c("<10,000, >= 8,000","<15,000, >=12,500","<20,000, >=17,500",">=35,000")),
         fi_4=as.numeric(family_income %in%
              c("<17,500, >=15,000","<22,500, >=20,000","<30,000, >=27,500"))
        ) %>%
  select(-family_income)

table(rg$self_employed)
```

```
##
##   No  Yes
## 9385  715
```

```
table(rg$self_employed_partner)
```

```
##
##   No  Yes
## 8973 1127
```

```
table(rg$gender)
```

```
##
##  Female    Male Unknown
##    7596    2469      35
```

```
rg=rg %>%
  mutate(self_emp_yes=as.numeric(self_employed=="Yes"),
         self_emp_part_yes=as.numeric(self_employed_partner=="Yes"),
         gender_f=as.numeric(gender=="Female"),
         gender_m=as.numeric(gender=="Male")) %>%
  select(-self_employed,-self_employed_partner,-gender)
```

We are dropping variables `post_code`, `post_area`. They take too many distinct values for these variables to be useful in modeling process. We are also dropping variables `TVarea` and `region`. You can process them as we have done for categorical variables with many categories and see if makes performance of your model better.

```
rg=rg %>%
  select(-TVarea,-post_code,-post_area,-region)
```

Now our data has all numeric vars and ready for modelling process.Lets break it into train and test.

```
set.seed(2)
s=sample(1:nrow(rg),0.7*nrow(rg))
rg_train=rg[s,]
rg_test=rg[-s,]
```

First thing that we'll be looking to eliminate is severe cases of multi-collinearity. [mild cases of multicollinearity is not an issue in logistic regression]. To examine VIF, we can run a linear regression. We are not concerned with the output of this linear regression model, we are only interested in VIF values of the predictor.

```
library(car)
for_vif=lm(Revenue.Grid~.-REF_NO,data=rg_train)
vif(for_vif)
```

```
##                        children                year_last_moved
##                     1.406039e+00                    1.149760e+00
## Average.Credit.Card.Transaction                Balance.Transfer
##                     1.392445e+07                    3.469292e+07
##                     Term.Deposit                  Life.Insurance
##                     1.506211e+07                    8.369697e+07
##                Medical.Insurance             Average.A.C.Balance
##                     9.445974e+06                    1.333791e+07
##                    Personal.Loan       Investment.in.Mutual.Fund
##                     5.462973e+07                    2.861701e+07
##      Investment.Tax.Saving.Bond                       Home.Loan
##                     1.103905e+06                    3.784454e+05
##          Online.Purchase.Amount          Investment.in.Commudity
##                     2.488567e+07                    2.461260e+08
##             Investment.in.Equity         Investment.in.Derivative
##                     1.469331e+08                    2.154981e+08
##                Portfolio.Balance                             age
##                     1.356355e+01                    2.349850e+00
##                       status_div                  status_partner
##                     2.066407e+00                    3.693805e+00
##                    status_single                     occ_BM_prof
##                     3.083577e+00                    1.458711e+00
##                     occ_Retired                          occ_HW
##                     2.203752e+00                    1.268323e+00
##                             op_1                             op_2
##                     1.663551e+00                    1.115622e+00
##                        hs_livein                          hs_own
##                     7.375955e+00                    3.643886e+01
##                  hs_rent_private                 hs_rent_council
##                     1.439103e+01                    1.800098e+01
##                             fi_1                             fi_2
##                     1.038826e+01                    2.930104e+01
##                             fi_3                             fi_4
##                     3.674891e+01                    2.591935e+01
##                     self_emp_yes                self_emp_part_yes
##                     1.095195e+00                    1.137885e+00
##                         gender_f                         gender_m
##                     5.112265e+01                    5.114283e+01
```

You can see there are few cases of insanely high VIF values , lets eliminate those variables one by one.

```
for_vif=lm(Revenue.Grid~.-REF_NO-Investment.in.Commudity,data=rg_train)
vif(for_vif)
```

```
##                        children                year_last_moved
##                     1.405880e+00                    1.149343e+00
## Average.Credit.Card.Transaction                Balance.Transfer
##                     1.342111e+00                    2.005925e+00
##                     Term.Deposit                  Life.Insurance
##                     1.575316e+00                    3.407520e+07
##                Medical.Insurance             Average.A.C.Balance
##                     3.845709e+06                    1.333444e+07
##                    Personal.Loan       Investment.in.Mutual.Fund
##                     5.461551e+07                    2.860955e+07
##      Investment.Tax.Saving.Bond                       Home.Loan
```

```
##                    1.103618e+06                    3.783750e+05
##           Online.Purchase.Amount             Investment.in.Equity
##                    2.488102e+07                    1.469057e+08
##         Investment.in.Derivative                Portfolio.Balance
##                    2.154855e+08                    1.356330e+01
##                             age                      status_div
##                    2.349567e+00                    2.066288e+00
##                  status_partner                   status_single
##                    3.693799e+00                    3.083305e+00
##                     occ_BM_prof                     occ_Retired
##                    1.458626e+00                    2.203357e+00
##                          occ_HW                             op_1
##                    1.267072e+00                    1.663062e+00
##                             op_2                        hs_livein
##                    1.115591e+00                    7.375860e+00
##                          hs_own                  hs_rent_private
##                    3.643782e+01                    1.439017e+01
##                 hs_rent_council                             fi_1
##                    1.799963e+01                    1.038556e+01
##                             fi_2                             fi_3
##                    2.929936e+01                    3.674248e+01
##                             fi_4                    self_emp_yes
##                    2.591646e+01                    1.095194e+00
##               self_emp_part_yes                        gender_f
##                    1.137881e+00                    5.111900e+01
##                        gender_m
##                    5.113938e+01
```

```
for_vif=lm(Revenue.Grid~.-REF_NO-Investment.in.Commudity-Investment.in.Derivative ,data=rg_train)
vif(for_vif)
```

```
##                          children                   year_last_moved
##                    1.405798e+00                    1.149300e+00
## Average.Credit.Card.Transaction                  Balance.Transfer
##                    1.341637e+00                    2.005498e+00
##                    Term.Deposit                   Life.Insurance
##                    1.575305e+00                    3.443909e+00
##                Medical.Insurance               Average.A.C.Balance
##                    1.694007e+00                    7.725666e+06
##                   Personal.Loan         Investment.in.Mutual.Fund
##                    3.164266e+07                    1.657565e+07
##       Investment.Tax.Saving.Bond                       Home.Loan
##                    6.394085e+05                    3.728427e+05
##           Online.Purchase.Amount             Investment.in.Equity
##                    2.451703e+07                    1.447566e+08
##                Portfolio.Balance                             age
##                    1.356288e+01                    2.349439e+00
##                      status_div                  status_partner
##                    2.065876e+00                    3.693072e+00
##                     status_single                    occ_BM_prof
##                    3.082914e+00                    1.458191e+00
##                      occ_Retired                          occ_HW
##                    2.202954e+00                    1.267063e+00
##                             op_1                             op_2
##                    1.662305e+00                    1.115570e+00
```

```
##                      hs_livein                       hs_own
##                   7.375657e+00                  3.643776e+01
##                hs_rent_private                hs_rent_council
##                   1.439016e+01                  1.799962e+01
##                           fi_1                           fi_2
##                   1.038510e+01                  2.929897e+01
##                           fi_3                           fi_4
##                   3.674189e+01                  2.591571e+01
##                   self_emp_yes             self_emp_part_yes
##                   1.095049e+00                  1.137874e+00
##                       gender_f                       gender_m
##                   5.111619e+01                  5.113756e+01
```

```
for_vif=lm(Revenue.Grid~.-REF_NO-Investment.in.Commudity-Investment.in.Derivative
           -Investment.in.Equity,data=rg_train)
vif(for_vif)
```

```
##                         children                year_last_moved
##                         1.405722                      1.149299
## Average.Credit.Card.Transaction               Balance.Transfer
##                         1.341636                      2.005167
##                     Term.Deposit                 Life.Insurance
##                         1.574566                      3.442981
##                Medical.Insurance            Average.A.C.Balance
##                         1.693944                      1.843127
##                    Personal.Loan      Investment.in.Mutual.Fund
##                         2.395305                      2.270952
##      Investment.Tax.Saving.Bond                     Home.Loan
##                         1.263203                      1.221573
##          Online.Purchase.Amount               Portfolio.Balance
##                         1.299196                     13.561316
##                             age                    status_div
##                         2.348895                      2.064304
##                  status_partner                  status_single
##                         3.689496                      3.078125
##                     occ_BM_prof                   occ_Retired
##                         1.458087                      2.202512
##                          occ_HW                           op_1
##                         1.266982                      1.662285
##                             op_2                     hs_livein
##                         1.115494                      7.375465
##                          hs_own                hs_rent_private
##                        36.432436                     14.387391
##                hs_rent_council                           fi_1
##                        17.998118                     10.385043
##                             fi_2                           fi_3
##                        29.298957                     36.741892
##                             fi_4                   self_emp_yes
##                        25.915696                      1.095040
##                self_emp_part_yes                      gender_f
##                         1.137854                     51.103380
##                         gender_m
##                        51.125480
```

```
for_vif=lm(Revenue.Grid~.-REF_NO-Investment.in.Commudity-Investment.in.Derivative
          -Investment.in.Equity-gender_m,data=rg_train)
vif(for_vif)
```

```
##                         children                 year_last_moved
##                         1.405367                        1.149291
## Average.Credit.Card.Transaction                Balance.Transfer
##                         1.341379                        2.004942
##                     Term.Deposit                  Life.Insurance
##                         1.574511                        3.442969
##                Medical.Insurance             Average.A.C.Balance
##                         1.693944                        1.841890
##                    Personal.Loan       Investment.in.Mutual.Fund
##                         2.395237                        2.270913
##      Investment.Tax.Saving.Bond                       Home.Loan
##                         1.263151                        1.221472
##          Online.Purchase.Amount                Portfolio.Balance
##                         1.299192                       13.560564
##                              age                      status_div
##                         2.348895                        2.064272
##                   status_partner                   status_single
##                         3.689004                        3.077448
##                     occ_BM_prof                     occ_Retired
##                         1.458087                        2.202256
##                           occ_HW                            op_1
##                         1.266949                        1.662104
##                            op_2                       hs_livein
##                         1.113936                        7.375069
##                           hs_own                  hs_rent_private
##                        36.432402                       14.387214
##                  hs_rent_council                            fi_1
##                        17.997940                       10.385043
##                            fi_2                            fi_3
##                        29.298649                       36.741783
##                            fi_4                     self_emp_yes
##                        25.915302                        1.094770
##                self_emp_part_yes                        gender_f
##                         1.137553                        1.192562
```

```
for_vif=lm(Revenue.Grid~.-REF_NO-Investment.in.Commudity-Investment.in.Derivative
          -Investment.in.Equity-gender_m-hs_own,data=rg_train)
vif(for_vif)
```

```
##                         children                 year_last_moved
##                         1.405146                        1.066514
## Average.Credit.Card.Transaction                Balance.Transfer
##                         1.341096                        2.004413
##                     Term.Deposit                  Life.Insurance
##                         1.574511                        3.442709
##                Medical.Insurance             Average.A.C.Balance
##                         1.693861                        1.841777
##                    Personal.Loan       Investment.in.Mutual.Fund
##                         2.395216                        2.270610
##      Investment.Tax.Saving.Bond                       Home.Loan
```

```
##                        1.263143                        1.221401
##          Online.Purchase.Amount               Portfolio.Balance
##                        1.299186                        13.556397
##                             age                      status_div
##                        2.343841                        2.063786
##                  status_partner                    status_single
##                        3.688109                        3.077424
##                     occ_BM_prof                     occ_Retired
##                        1.458086                        2.202057
##                          occ_HW                             op_1
##                        1.266846                        1.662103
##                             op_2                      hs_livein
##                        1.113908                        1.077322
##                 hs_rent_private                  hs_rent_council
##                        1.017966                        1.075867
##                             fi_1                             fi_2
##                        9.796221                       27.391396
##                             fi_3                             fi_4
##                       34.353851                       24.254057
##                    self_emp_yes                self_emp_part_yes
##                        1.094578                        1.137522
##                        gender_f
##                        1.191991
```

```
for_vif=lm(Revenue.Grid~.-REF_NO-Investment.in.Commudity-Investment.in.Derivative
           -Investment.in.Equity-gender_m-hs_own-fi_3,data=rg_train)
vif(for_vif)
```

```
##                        children                year_last_moved
##                        1.404964                        1.008359
## Average.Credit.Card.Transaction                Balance.Transfer
##                        1.340889                        2.004342
##                    Term.Deposit                  Life.Insurance
##                        1.573748                        3.441857
##                Medical.Insurance             Average.A.C.Balance
##                        1.692738                        1.841753
##                   Personal.Loan       Investment.in.Mutual.Fund
##                        2.395186                        2.270290
##      Investment.Tax.Saving.Bond                       Home.Loan
##                        1.263143                        1.221326
##          Online.Purchase.Amount               Portfolio.Balance
##                        1.299163                        13.556162
##                             age                      status_div
##                        2.342182                        2.063155
##                  status_partner                    status_single
##                        3.687170                        3.074666
##                     occ_BM_prof                     occ_Retired
##                        1.457704                        2.201859
##                          occ_HW                             op_1
##                        1.266776                        1.660900
##                             op_2                      hs_livein
##                        1.113896                        1.076795
##                 hs_rent_private                  hs_rent_council
##                        1.016312                        1.075867
##                             fi_1                             fi_2
```

```
##                         1.326843                         1.200390
##                             fi_4                      self_emp_yes
##                         1.202264                         1.094573
##                self_emp_part_yes                         gender_f
##                         1.137465                         1.191764
```

```
for_vif=lm(Revenue.Grid~.-REF_NO-Investment.in.Commudity-Investment.in.Derivative
           -Investment.in.Equity-gender_m-hs_own-fi_3-Portfolio.Balance,data=rg_train)
vif(for_vif)
```

```
##                         children                 year_last_moved
##                         1.404774                         1.008351
## Average.Credit.Card.Transaction                 Balance.Transfer
##                         1.237463                         1.703295
##                      Term.Deposit                   Life.Insurance
##                         1.437386                         2.079784
##                 Medical.Insurance              Average.A.C.Balance
##                         1.517226                         1.617689
##                     Personal.Loan       Investment.in.Mutual.Fund
##                         1.408566                         1.708482
##       Investment.Tax.Saving.Bond                        Home.Loan
##                         1.238099                         1.219443
##           Online.Purchase.Amount                              age
##                         1.115391                         2.342106
##                       status_div                   status_partner
##                         2.063155                         3.687170
##                     status_single                      occ_BM_prof
##                         3.074019                         1.457652
##                      occ_Retired                          occ_HW
##                         2.198683                         1.266773
##                             op_1                             op_2
##                         1.659915                         1.112140
##                       hs_livein                  hs_rent_private
##                         1.075737                         1.016303
##                  hs_rent_council                             fi_1
##                         1.075831                         1.326734
##                             fi_2                             fi_4
##                         1.200389                         1.202239
##                      self_emp_yes                self_emp_part_yes
##                         1.094573                         1.137290
##                         gender_f
##                         1.191741
```

All VIF values now are less than 10. This is good enough for logistic regression , Lets move to build our classification model now. [We'll keep on excluding the variables with high VIF values, which we identified]

```
rg_fit=rg_train %>%
  select(-REF_NO,-Investment.in.Commudity,-Investment.in.Derivative,
           -Investment.in.Equity,-gender_m,-hs_own,-fi_3,-Portfolio.Balance)
fit=glm(Revenue.Grid~.,family = "binomial",data=rg_fit)
```

```
## Error in eval(expr, envir, enclos): y values must be 0 <= y <= 1
```

You get an error that y values or your response should be 0 and 1. In our data they are 1 and 2, lets do that conversion and move ahead. [We'll have to redo sampling for this effect to appear across all data]

```
rg$Revenue.Grid=as.numeric(rg$Revenue.Grid==1)
set.seed(2)
s=sample(1:nrow(rg),0.7*nrow(rg))
rg_train=rg[s,]
rg_test=rg[-s,]

rg_fit=rg_train %>%
  select(-REF_NO,-Investment.in.Commudity,-Investment.in.Derivative,
           -Investment.in.Equity,-gender_m,-hs_own,-fi_3,-Portfolio.Balance)
fit=glm(Revenue.Grid~.,family = "binomial",data=rg_fit)
```

We can now look at summary for this fit and start dropping variables based on p-values, one by one. That is one option or we can use function `step` on the model object `fit`. Function step will exclude variables from the model one by one based AIC score.

```
fit=step(fit)
```

```
## Start:  AIC=2098.39
## Revenue.Grid ~ children + year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     age + status_div + status_partner + status_single + occ_BM_prof +
##     occ_Retired + occ_HW + op_1 + op_2 + hs_livein + hs_rent_private +
##     hs_rent_council + fi_1 + fi_2 + fi_4 + self_emp_yes + self_emp_part_yes +
##     gender_f
##
##                                  Df Deviance    AIC
## - gender_f                        1   2034.4 2096.4
## - occ_BM_prof                     1   2034.4 2096.4
## - occ_HW                          1   2034.4 2096.4
## - hs_livein                       1   2034.5 2096.5
## - occ_Retired                     1   2034.5 2096.5
## - status_single                   1   2034.5 2096.5
## - op_1                            1   2034.5 2096.5
## - hs_rent_private                 1   2034.5 2096.5
## - status_partner                  1   2034.6 2096.6
## - age                             1   2034.7 2096.7
## - fi_2                            1   2035.2 2097.2
## - status_div                      1   2035.5 2097.5
## - children                        1   2035.6 2097.6
## - self_emp_yes                    1   2035.8 2097.8
## - hs_rent_council                 1   2036.0 2098.0
## - self_emp_part_yes               1   2036.1 2098.1
## <none>                                2034.4 2098.4
## - fi_1                            1   2036.4 2098.4
## - op_2                            1   2037.0 2099.0
## - fi_4                            1   2037.1 2099.1
## - year_last_moved                 1   2038.1 2100.1
## - Investment.in.Mutual.Fund       1   2039.5 2101.5
## - Balance.Transfer                1   2062.6 2124.6
## - Medical.Insurance               1   2076.1 2138.1
## - Average.A.C.Balance             1   2087.0 2149.0
## - Home.Loan                       1   2117.5 2179.5
```

```
## - Term.Deposit                     1   2151.7 2213.7
## - Life.Insurance                   1   2365.6 2427.6
## - Investment.Tax.Saving.Bond       1   2457.2 2519.2
## - Personal.Loan                    1   2524.8 2586.8
## - Average.Credit.Card.Transaction  1   2576.3 2638.3
## - Online.Purchase.Amount           1   3585.6 3647.6
##
## Step:  AIC=2096.39
## Revenue.Grid ~ children + year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     age + status_div + status_partner + status_single + occ_BM_prof +
##     occ_Retired + occ_HW + op_1 + op_2 + hs_livein + hs_rent_private +
##     hs_rent_council + fi_1 + fi_2 + fi_4 + self_emp_yes + self_emp_part_yes
##
##                                   Df Deviance    AIC
## - occ_HW                           1   2034.4 2094.4
## - occ_BM_prof                      1   2034.4 2094.4
## - hs_livein                        1   2034.5 2094.5
## - occ_Retired                      1   2034.5 2094.5
## - status_single                    1   2034.5 2094.5
## - op_1                             1   2034.5 2094.5
## - hs_rent_private                  1   2034.5 2094.5
## - status_partner                   1   2034.6 2094.6
## - age                              1   2034.7 2094.7
## - fi_2                             1   2035.2 2095.2
## - status_div                       1   2035.5 2095.5
## - children                         1   2035.6 2095.6
## - self_emp_yes                     1   2035.8 2095.8
## - hs_rent_council                  1   2036.0 2096.0
## - self_emp_part_yes                1   2036.1 2096.1
## <none>                                 2034.4 2096.4
## - fi_1                             1   2036.5 2096.5
## - fi_4                             1   2037.1 2097.1
## - op_2                             1   2037.1 2097.1
## - year_last_moved                  1   2038.1 2098.1
## - Investment.in.Mutual.Fund        1   2039.5 2099.5
## - Balance.Transfer                 1   2062.6 2122.6
## - Medical.Insurance                1   2076.2 2136.2
## - Average.A.C.Balance              1   2087.0 2147.0
## - Home.Loan                        1   2117.5 2177.5
## - Term.Deposit                     1   2151.7 2211.7
## - Life.Insurance                   1   2365.8 2425.8
## - Investment.Tax.Saving.Bond       1   2457.2 2517.2
## - Personal.Loan                    1   2525.8 2585.8
## - Average.Credit.Card.Transaction  1   2577.0 2637.0
## - Online.Purchase.Amount           1   3585.7 3645.7
##
## Step:  AIC=2094.44
## Revenue.Grid ~ children + year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
```

```
##     age + status_div + status_partner + status_single + occ_BM_prof +
##     occ_Retired + op_1 + op_2 + hs_livein + hs_rent_private +
##     hs_rent_council + fi_1 + fi_2 + fi_4 + self_emp_yes + self_emp_part_yes
##
##                                   Df Deviance    AIC
## - occ_Retired                      1   2034.5 2092.5
## - hs_livein                        1   2034.5 2092.5
## - status_single                    1   2034.5 2092.5
## - op_1                             1   2034.5 2092.5
## - occ_BM_prof                      1   2034.5 2092.5
## - hs_rent_private                  1   2034.6 2092.6
## - status_partner                   1   2034.7 2092.7
## - age                              1   2034.7 2092.7
## - fi_2                             1   2035.2 2093.2
## - status_div                       1   2035.6 2093.6
## - children                         1   2035.6 2093.6
## - self_emp_yes                     1   2035.9 2093.9
## - hs_rent_council                  1   2036.0 2094.0
## - self_emp_part_yes                1   2036.1 2094.1
## <none>                                 2034.4 2094.4
## - fi_1                             1   2036.5 2094.5
## - fi_4                             1   2037.2 2095.2
## - op_2                             1   2037.2 2095.2
## - year_last_moved                  1   2038.1 2096.1
## - Investment.in.Mutual.Fund        1   2039.6 2097.6
## - Balance.Transfer                 1   2062.7 2120.7
## - Medical.Insurance                1   2076.2 2134.2
## - Average.A.C.Balance              1   2087.1 2145.1
## - Home.Loan                        1   2117.6 2175.6
## - Term.Deposit                     1   2151.7 2209.7
## - Life.Insurance                   1   2366.2 2424.2
## - Investment.Tax.Saving.Bond       1   2457.3 2515.3
## - Personal.Loan                    1   2525.8 2583.8
## - Average.Credit.Card.Transaction  1   2577.1 2635.1
## - Online.Purchase.Amount           1   3586.0 3644.0
##
## Step:  AIC=2092.48
## Revenue.Grid ~ children + year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     age + status_div + status_partner + status_single + occ_BM_prof +
##     op_1 + op_2 + hs_livein + hs_rent_private + hs_rent_council +
##     fi_1 + fi_2 + fi_4 + self_emp_yes + self_emp_part_yes
##
##                                   Df Deviance    AIC
## - hs_livein                        1   2034.6 2090.6
## - op_1                             1   2034.6 2090.6
## - status_single                    1   2034.6 2090.6
## - occ_BM_prof                      1   2034.6 2090.6
## - hs_rent_private                  1   2034.7 2090.7
## - age                              1   2034.7 2090.7
## - status_partner                   1   2034.7 2090.7
## - fi_2                             1   2035.3 2091.3
```

```
## - status_div                        1   2035.6 2091.6
## - children                          1   2035.7 2091.7
## - self_emp_yes                      1   2036.0 2092.0
## - hs_rent_council                   1   2036.0 2092.0
## - self_emp_part_yes                 1   2036.3 2092.3
## <none>                                  2034.5 2092.5
## - fi_1                              1   2036.6 2092.6
## - op_2                              1   2037.2 2093.2
## - fi_4                              1   2037.2 2093.2
## - year_last_moved                   1   2038.1 2094.1
## - Investment.in.Mutual.Fund         1   2039.6 2095.6
## - Balance.Transfer                  1   2062.8 2118.8
## - Medical.Insurance                 1   2076.3 2132.3
## - Average.A.C.Balance               1   2087.1 2143.1
## - Home.Loan                         1   2117.8 2173.8
## - Term.Deposit                      1   2151.8 2207.8
## - Life.Insurance                    1   2366.3 2422.3
## - Investment.Tax.Saving.Bond        1   2457.3 2513.3
## - Personal.Loan                     1   2525.9 2581.9
## - Average.Credit.Card.Transaction   1   2577.2 2633.2
## - Online.Purchase.Amount            1   3586.2 3642.2
##
## Step:  AIC=2090.55
## Revenue.Grid ~ children + year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     age + status_div + status_partner + status_single + occ_BM_prof +
##     op_1 + op_2 + hs_rent_private + hs_rent_council + fi_1 +
##     fi_2 + fi_4 + self_emp_yes + self_emp_part_yes
##
##                                    Df Deviance    AIC
## - op_1                              1   2034.6 2088.6
## - status_single                     1   2034.6 2088.6
## - occ_BM_prof                       1   2034.7 2088.7
## - hs_rent_private                   1   2034.7 2088.7
## - status_partner                    1   2034.8 2088.8
## - age                               1   2034.8 2088.8
## - fi_2                              1   2035.4 2089.4
## - status_div                        1   2035.7 2089.7
## - children                          1   2035.8 2089.8
## - self_emp_yes                      1   2036.1 2090.1
## - hs_rent_council                   1   2036.1 2090.1
## - self_emp_part_yes                 1   2036.4 2090.4
## <none>                                  2034.6 2090.6
## - fi_1                              1   2036.6 2090.6
## - op_2                              1   2037.3 2091.3
## - fi_4                              1   2037.3 2091.3
## - year_last_moved                   1   2038.2 2092.2
## - Investment.in.Mutual.Fund         1   2039.6 2093.6
## - Balance.Transfer                  1   2062.8 2116.8
## - Medical.Insurance                 1   2076.4 2130.4
## - Average.A.C.Balance               1   2087.1 2141.1
## - Home.Loan                         1   2118.0 2172.0
```

```
## - Term.Deposit                      1   2151.8 2205.8
## - Life.Insurance                    1   2366.3 2420.3
## - Investment.Tax.Saving.Bond        1   2457.3 2511.3
## - Personal.Loan                     1   2526.0 2580.0
## - Average.Credit.Card.Transaction   1   2577.5 2631.5
## - Online.Purchase.Amount            1   3586.2 3640.2
##
## Step:  AIC=2088.62
## Revenue.Grid ~ children + year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     age + status_div + status_partner + status_single + occ_BM_prof +
##     op_2 + hs_rent_private + hs_rent_council + fi_1 + fi_2 +
##     fi_4 + self_emp_yes + self_emp_part_yes
##
##                                    Df Deviance    AIC
## - status_single                     1   2034.7 2086.7
## - occ_BM_prof                       1   2034.8 2086.8
## - hs_rent_private                   1   2034.8 2086.8
## - status_partner                    1   2034.8 2086.8
## - age                               1   2035.0 2087.0
## - fi_2                              1   2035.4 2087.4
## - status_div                        1   2035.7 2087.7
## - children                          1   2035.8 2087.8
## - self_emp_yes                      1   2036.2 2088.2
## - hs_rent_council                   1   2036.2 2088.2
## - self_emp_part_yes                 1   2036.4 2088.4
## <none>                                  2034.6 2088.6
## - fi_1                              1   2036.7 2088.7
## - op_2                              1   2037.3 2089.3
## - fi_4                              1   2037.4 2089.4
## - year_last_moved                   1   2038.3 2090.3
## - Investment.in.Mutual.Fund         1   2039.8 2091.8
## - Balance.Transfer                  1   2062.9 2114.9
## - Medical.Insurance                 1   2076.5 2128.5
## - Average.A.C.Balance               1   2087.2 2139.2
## - Home.Loan                         1   2118.0 2170.0
## - Term.Deposit                      1   2151.8 2203.8
## - Life.Insurance                    1   2366.7 2418.7
## - Investment.Tax.Saving.Bond        1   2457.7 2509.7
## - Personal.Loan                     1   2526.2 2578.2
## - Average.Credit.Card.Transaction   1   2577.5 2629.5
## - Online.Purchase.Amount            1   3586.3 3638.3
##
## Step:  AIC=2086.7
## Revenue.Grid ~ children + year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     age + status_div + status_partner + occ_BM_prof + op_2 +
##     hs_rent_private + hs_rent_council + fi_1 + fi_2 + fi_4 +
##     self_emp_yes + self_emp_part_yes
##
```

```
##                                   Df Deviance    AIC
## - occ_BM_prof                      1   2034.8 2084.8
## - hs_rent_private                  1   2034.9 2084.9
## - age                              1   2035.3 2085.3
## - fi_2                             1   2035.5 2085.5
## - status_partner                   1   2035.9 2085.9
## - status_div                       1   2035.9 2085.9
## - children                         1   2036.0 2086.0
## - hs_rent_council                  1   2036.2 2086.2
## - self_emp_yes                     1   2036.3 2086.3
## - self_emp_part_yes                1   2036.5 2086.5
## - fi_1                             1   2036.7 2086.7
## <none>                                 2034.7 2086.7
## - op_2                             1   2037.4 2087.4
## - fi_4                             1   2037.5 2087.5
## - year_last_moved                  1   2038.4 2088.4
## - Investment.in.Mutual.Fund        1   2039.9 2089.9
## - Balance.Transfer                 1   2063.1 2113.1
## - Medical.Insurance                1   2076.5 2126.5
## - Average.A.C.Balance              1   2087.4 2137.4
## - Home.Loan                        1   2118.0 2168.0
## - Term.Deposit                     1   2151.8 2201.8
## - Life.Insurance                   1   2366.8 2416.8
## - Investment.Tax.Saving.Bond       1   2457.7 2507.7
## - Personal.Loan                    1   2526.2 2576.2
## - Average.Credit.Card.Transaction  1   2577.5 2627.5
## - Online.Purchase.Amount           1   3586.3 3636.3
##
## Step:  AIC=2084.83
## Revenue.Grid ~ children + year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     age + status_div + status_partner + op_2 + hs_rent_private +
##     hs_rent_council + fi_1 + fi_2 + fi_4 + self_emp_yes + self_emp_part_yes
##
##                                   Df Deviance    AIC
## - hs_rent_private                  1   2035.0 2083.0
## - age                              1   2035.3 2083.3
## - fi_2                             1   2035.6 2083.6
## - status_partner                   1   2036.0 2084.0
## - status_div                       1   2036.0 2084.0
## - children                         1   2036.2 2084.2
## - hs_rent_council                  1   2036.3 2084.3
## - self_emp_yes                     1   2036.4 2084.4
## - self_emp_part_yes                1   2036.6 2084.6
## <none>                                 2034.8 2084.8
## - fi_1                             1   2037.0 2085.0
## - fi_4                             1   2037.5 2085.5
## - op_2                             1   2037.6 2085.6
## - year_last_moved                  1   2038.5 2086.5
## - Investment.in.Mutual.Fund        1   2040.0 2088.0
## - Balance.Transfer                 1   2063.2 2111.2
## - Medical.Insurance                1   2076.7 2124.7
```

```
## - Average.A.C.Balance              1   2087.4 2135.4
## - Home.Loan                        1   2118.5 2166.5
## - Term.Deposit                     1   2152.0 2200.0
## - Life.Insurance                   1   2367.5 2415.5
## - Investment.Tax.Saving.Bond       1   2458.4 2506.4
## - Personal.Loan                    1   2526.3 2574.3
## - Average.Credit.Card.Transaction  1   2577.6 2625.6
## - Online.Purchase.Amount           1   3586.9 3634.9
##
## Step:  AIC=2083
## Revenue.Grid ~ children + year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     age + status_div + status_partner + op_2 + hs_rent_council +
##     fi_1 + fi_2 + fi_4 + self_emp_yes + self_emp_part_yes
##
##                                   Df Deviance    AIC
## - age                              1   2035.5 2081.5
## - fi_2                             1   2035.8 2081.8
## - status_partner                   1   2036.2 2082.2
## - status_div                       1   2036.2 2082.2
## - children                         1   2036.3 2082.3
## - hs_rent_council                  1   2036.4 2082.4
## - self_emp_yes                     1   2036.6 2082.6
## - self_emp_part_yes                1   2036.7 2082.7
## <none>                                 2035.0 2083.0
## - fi_1                             1   2037.0 2083.0
## - fi_4                             1   2037.7 2083.7
## - op_2                             1   2037.8 2083.8
## - year_last_moved                  1   2038.7 2084.7
## - Investment.in.Mutual.Fund        1   2040.2 2086.2
## - Balance.Transfer                 1   2063.4 2109.4
## - Medical.Insurance                1   2077.0 2123.0
## - Average.A.C.Balance              1   2087.6 2133.6
## - Home.Loan                        1   2118.7 2164.7
## - Term.Deposit                     1   2152.3 2198.3
## - Life.Insurance                   1   2367.5 2413.5
## - Investment.Tax.Saving.Bond       1   2458.7 2504.7
## - Personal.Loan                    1   2526.7 2572.7
## - Average.Credit.Card.Transaction  1   2577.6 2623.6
## - Online.Purchase.Amount           1   3587.0 3633.0
##
## Step:  AIC=2081.46
## Revenue.Grid ~ children + year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     status_div + status_partner + op_2 + hs_rent_council + fi_1 +
##     fi_2 + fi_4 + self_emp_yes + self_emp_part_yes
##
##                                   Df Deviance    AIC
## - fi_2                             1   2036.2 2080.2
## - children                         1   2036.4 2080.4
```

```
## - status_div                       1   2036.5 2080.5
## - hs_rent_council                  1   2036.9 2080.9
## - status_partner                   1   2037.1 2081.1
## - self_emp_yes                     1   2037.1 2081.1
## - self_emp_part_yes                1   2037.1 2081.1
## - fi_1                             1   2037.2 2081.2
## <none>                                 2035.5 2081.5
## - op_2                             1   2038.0 2082.0
## - fi_4                             1   2038.1 2082.1
## - year_last_moved                  1   2038.8 2082.8
## - Investment.in.Mutual.Fund        1   2040.6 2084.6
## - Balance.Transfer                 1   2064.2 2108.2
## - Medical.Insurance                1   2077.8 2121.8
## - Average.A.C.Balance              1   2088.3 2132.3
## - Home.Loan                        1   2119.1 2163.1
## - Term.Deposit                     1   2152.6 2196.6
## - Life.Insurance                   1   2368.7 2412.7
## - Investment.Tax.Saving.Bond       1   2459.1 2503.1
## - Personal.Loan                    1   2528.5 2572.5
## - Average.Credit.Card.Transaction  1   2578.7 2622.7
## - Online.Purchase.Amount           1   3587.0 3631.0
##
## Step:  AIC=2080.19
## Revenue.Grid ~ children + year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     status_div + status_partner + op_2 + hs_rent_council + fi_1 +
##     fi_4 + self_emp_yes + self_emp_part_yes
##
##                                   Df Deviance    AIC
## - children                         1   2037.0 2079.0
## - status_div                       1   2037.2 2079.2
## - hs_rent_council                  1   2037.6 2079.6
## - self_emp_yes                     1   2037.7 2079.7
## - status_partner                   1   2037.8 2079.8
## - self_emp_part_yes                1   2037.8 2079.8
## - fi_4                             1   2038.2 2080.2
## <none>                                 2036.2 2080.2
## - fi_1                             1   2038.6 2080.6
## - op_2                             1   2038.8 2080.8
## - year_last_moved                  1   2039.6 2081.6
## - Investment.in.Mutual.Fund        1   2041.4 2083.4
## - Balance.Transfer                 1   2064.9 2106.9
## - Medical.Insurance                1   2078.2 2120.2
## - Average.A.C.Balance              1   2089.0 2131.0
## - Home.Loan                        1   2120.0 2162.0
## - Term.Deposit                     1   2153.3 2195.3
## - Life.Insurance                   1   2369.2 2411.2
## - Investment.Tax.Saving.Bond       1   2460.5 2502.5
## - Personal.Loan                    1   2529.8 2571.8
## - Average.Credit.Card.Transaction  1   2578.7 2620.7
## - Online.Purchase.Amount           1   3588.3 3630.3
##
```

```
## Step:  AIC=2079.03
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     status_div + status_partner + op_2 + hs_rent_council + fi_1 +
##     fi_4 + self_emp_yes + self_emp_part_yes
##
##                                   Df Deviance    AIC
## - status_div                       1   2037.9 2077.9
## - hs_rent_council                  1   2038.5 2078.5
## - self_emp_yes                     1   2038.6 2078.6
## - self_emp_part_yes                1   2038.9 2078.9
## <none>                                 2037.0 2079.0
## - fi_4                             1   2039.1 2079.1
## - status_partner                   1   2039.3 2079.3
## - fi_1                             1   2039.5 2079.5
## - op_2                             1   2039.7 2079.7
## - year_last_moved                  1   2040.7 2080.7
## - Investment.in.Mutual.Fund        1   2042.1 2082.1
## - Balance.Transfer                 1   2065.6 2105.6
## - Medical.Insurance                1   2079.4 2119.4
## - Average.A.C.Balance              1   2090.4 2130.4
## - Home.Loan                        1   2120.8 2160.8
## - Term.Deposit                     1   2154.2 2194.2
## - Life.Insurance                   1   2370.2 2410.2
## - Investment.Tax.Saving.Bond       1   2461.5 2501.5
## - Personal.Loan                    1   2532.2 2572.2
## - Average.Credit.Card.Transaction  1   2579.6 2619.6
## - Online.Purchase.Amount           1   3588.4 3628.4
##
## Step:  AIC=2077.9
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     status_partner + op_2 + hs_rent_council + fi_1 + fi_4 + self_emp_yes +
##     self_emp_part_yes
##
##                                   Df Deviance    AIC
## - hs_rent_council                  1   2039.3 2077.3
## - self_emp_yes                     1   2039.4 2077.4
## - self_emp_part_yes                1   2039.7 2077.7
## <none>                                 2037.9 2077.9
## - fi_4                             1   2039.9 2077.9
## - fi_1                             1   2040.3 2078.3
## - op_2                             1   2040.6 2078.6
## - year_last_moved                  1   2041.5 2079.5
## - status_partner                   1   2042.4 2080.4
## - Investment.in.Mutual.Fund        1   2042.9 2080.9
## - Balance.Transfer                 1   2066.8 2104.8
## - Medical.Insurance                1   2080.2 2118.2
## - Average.A.C.Balance              1   2091.2 2129.2
## - Home.Loan                        1   2121.6 2159.6
```

```
## - Term.Deposit                      1   2154.9 2192.9
## - Life.Insurance                    1   2371.0 2409.0
## - Investment.Tax.Saving.Bond        1   2462.4 2500.4
## - Personal.Loan                     1   2535.2 2573.2
## - Average.Credit.Card.Transaction   1   2580.9 2618.9
## - Online.Purchase.Amount            1   3588.5 3626.5
##
## Step:  AIC=2077.29
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     status_partner + op_2 + fi_1 + fi_4 + self_emp_yes + self_emp_part_yes
##
##                                    Df Deviance    AIC
## - self_emp_yes                      1   2040.7 2076.7
## - fi_1                              1   2040.9 2076.9
## - self_emp_part_yes                 1   2041.1 2077.1
## <none>                                  2039.3 2077.3
## - fi_4                              1   2041.3 2077.3
## - op_2                              1   2042.0 2078.0
## - year_last_moved                   1   2043.0 2079.0
## - status_partner                    1   2043.4 2079.4
## - Investment.in.Mutual.Fund         1   2044.2 2080.2
## - Balance.Transfer                  1   2067.8 2103.8
## - Medical.Insurance                 1   2081.6 2117.6
## - Average.A.C.Balance               1   2092.2 2128.2
## - Home.Loan                         1   2123.0 2159.0
## - Term.Deposit                      1   2155.9 2191.9
## - Life.Insurance                    1   2371.5 2407.5
## - Investment.Tax.Saving.Bond        1   2463.1 2499.1
## - Personal.Loan                     1   2537.0 2573.0
## - Average.Credit.Card.Transaction   1   2583.3 2619.3
## - Online.Purchase.Amount            1   3588.8 3624.8
##
## Step:  AIC=2076.74
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     status_partner + op_2 + fi_1 + fi_4 + self_emp_part_yes
##
##                                    Df Deviance    AIC
## - fi_1                              1   2042.4 2076.4
## - fi_4                              1   2042.7 2076.7
## <none>                                  2040.7 2076.7
## - self_emp_part_yes                 1   2043.4 2077.4
## - op_2                              1   2043.6 2077.6
## - year_last_moved                   1   2044.5 2078.5
## - status_partner                    1   2044.9 2078.9
## - Investment.in.Mutual.Fund         1   2045.8 2079.8
## - Balance.Transfer                  1   2068.8 2102.8
## - Medical.Insurance                 1   2082.9 2116.9
## - Average.A.C.Balance               1   2093.6 2127.6
```

```
## - Home.Loan                          1   2123.5 2157.5
## - Term.Deposit                       1   2157.6 2191.6
## - Life.Insurance                     1   2372.5 2406.5
## - Investment.Tax.Saving.Bond         1   2464.2 2498.2
## - Personal.Loan                      1   2538.0 2572.0
## - Average.Credit.Card.Transaction    1   2584.0 2618.0
## - Online.Purchase.Amount             1   3589.6 3623.6
##
## Step:  AIC=2076.44
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     status_partner + op_2 + fi_4 + self_emp_part_yes
##
##                                    Df Deviance    AIC
## <none>                                  2042.4 2076.4
## - fi_4                              1   2045.0 2077.0
## - self_emp_part_yes                 1   2045.3 2077.3
## - op_2                              1   2045.5 2077.5
## - year_last_moved                   1   2046.7 2078.7
## - Investment.in.Mutual.Fund         1   2047.5 2079.5
## - status_partner                    1   2047.9 2079.9
## - Balance.Transfer                  1   2070.2 2102.2
## - Medical.Insurance                 1   2084.4 2116.4
## - Average.A.C.Balance               1   2095.4 2127.4
## - Home.Loan                         1   2124.9 2156.9
## - Term.Deposit                      1   2159.9 2191.9
## - Life.Insurance                    1   2374.8 2406.8
## - Investment.Tax.Saving.Bond        1   2466.3 2498.3
## - Personal.Loan                     1   2538.3 2570.3
## - Average.Credit.Card.Transaction   1   2586.8 2618.8
## - Online.Purchase.Amount            1   3594.7 3626.7
```

If you look at summary(fit), you'll find there are still some variable with high p-values. This is because dropping variables based on AIC scores has different inbuilt threshhold, which might not match with p-value decision for boundary cases. We can still run our logistic regression model with variables selected by step function and now drop variabe based on p-values on our own from the remaining bunch.

```
formula(fit)
```

```
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     status_partner + op_2 + fi_4 + self_emp_part_yes
```

We can use this to now run our model and drop variables based on p-values too.

```
fit=glm(Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +
          Term.Deposit + Life.Insurance + Medical.Insurance + Average.A.C.Balance +
          Personal.Loan + Investment.in.Mutual.Fund + Investment.Tax.Saving.Bond +
          Home.Loan + Online.Purchase.Amount + fi_2 + fi_4 + self_emp_part_yes +
          gender_f, data=rg_train, family = "binomial")
summary(fit)
```

```
##
## Call:
## glm(formula = Revenue.Grid ~ Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     fi_2 + fi_4 + self_emp_part_yes + gender_f, family = "binomial",
##     data = rg_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9957  -0.2367  -0.1536  -0.0716   4.1935
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -4.0759704  0.1631680 -24.980  < 2e-16 ***
## Average.Credit.Card.Transaction  0.0232186  0.0010707  21.686  < 2e-16 ***
## Balance.Transfer                -0.0055499  0.0011138  -4.983 6.26e-07 ***
## Term.Deposit                    -0.0161979  0.0017671  -9.166  < 2e-16 ***
## Life.Insurance                   0.0142174  0.0008324  17.079  < 2e-16 ***
## Medical.Insurance               -0.0146012  0.0023741  -6.150 7.73e-10 ***
## Average.A.C.Balance             -0.0128271  0.0018781  -6.830 8.51e-12 ***
## Personal.Loan                   -0.0304855  0.0021562 -14.139  < 2e-16 ***
## Investment.in.Mutual.Fund       -0.0027606  0.0011914  -2.317   0.0205 *
## Investment.Tax.Saving.Bond       0.0875062  0.0045132  19.389  < 2e-16 ***
## Home.Loan                       -0.0704791  0.0088897  -7.928 2.22e-15 ***
## Online.Purchase.Amount           0.0510862  0.0020721  24.655  < 2e-16 ***
## fi_2                             0.1702710  0.1427819   1.193   0.2331
## fi_4                             0.3106944  0.1476898   2.104   0.0354 *
## self_emp_part_yes                0.3646082  0.1794558   2.032   0.0422 *
## gender_f                        -0.0996661  0.1373543  -0.726   0.4681
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4821.0  on 7069  degrees of freedom
## Residual deviance: 2053.5  on 7054  degrees of freedom
## AIC: 2085.5
##
## Number of Fisher Scoring iterations: 8
```

```
fit=glm(Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +
          Term.Deposit + Life.Insurance + Medical.Insurance + Average.A.C.Balance +
          Personal.Loan + Investment.in.Mutual.Fund + Investment.Tax.Saving.Bond +
          Home.Loan + Online.Purchase.Amount +  fi_4 + self_emp_part_yes +
          gender_f , data=rg_train, family = "binomial")
summary(fit)
```

```
##
## Call:
## glm(formula = Revenue.Grid ~ Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
```

```
##     fi_4 + self_emp_part_yes + gender_f, family = "binomial",
##     data = rg_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9960  -0.2374  -0.1535  -0.0717   4.1769
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -4.0181760  0.1550525 -25.915  < 2e-16 ***
## Average.Credit.Card.Transaction  0.0231870  0.0010698  21.673  < 2e-16 ***
## Balance.Transfer                -0.0055413  0.0011142  -4.973 6.58e-07 ***
## Term.Deposit                    -0.0162241  0.0017663  -9.185  < 2e-16 ***
## Life.Insurance                   0.0142161  0.0008325  17.076  < 2e-16 ***
## Medical.Insurance               -0.0145099  0.0023716  -6.118 9.46e-10 ***
## Average.A.C.Balance             -0.0128265  0.0018780  -6.830 8.50e-12 ***
## Personal.Loan                   -0.0304836  0.0021563 -14.137  < 2e-16 ***
## Investment.in.Mutual.Fund       -0.0027662  0.0011896  -2.325   0.0201 *
## Investment.Tax.Saving.Bond       0.0876437  0.0045094  19.436  < 2e-16 ***
## Home.Loan                       -0.0705619  0.0088874  -7.940 2.03e-15 ***
## Online.Purchase.Amount           0.0510368  0.0020724  24.627  < 2e-16 ***
## fi_4                             0.2493729  0.1378994   1.808   0.0705 .
## self_emp_part_yes                0.3609578  0.1795009   2.011   0.0443 *
## gender_f                        -0.0942710  0.1373364  -0.686   0.4924
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4821.0  on 7069  degrees of freedom
## Residual deviance: 2054.9  on 7055  degrees of freedom
## AIC: 2084.9
##
## Number of Fisher Scoring iterations: 8
```

```
fit=glm(Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +
          Term.Deposit + Life.Insurance + Medical.Insurance + Average.A.C.Balance +
          Personal.Loan + Investment.in.Mutual.Fund + Investment.Tax.Saving.Bond +
          Home.Loan + Online.Purchase.Amount +self_emp_part_yes +
          gender_f, data=rg_train, family = "binomial")
summary(fit)
```

```
##
## Call:
## glm(formula = Revenue.Grid ~ Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     self_emp_part_yes + gender_f, family = "binomial", data = rg_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9746  -0.2364  -0.1545  -0.0726   4.1616
##
## Coefficients:
```

```
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -3.9624737  0.1512250 -26.203  < 2e-16 ***
## Average.Credit.Card.Transaction  0.0232085  0.0010717  21.655  < 2e-16 ***
## Balance.Transfer                -0.0055184  0.0011117  -4.964 6.91e-07 ***
## Term.Deposit                    -0.0162188  0.0017667  -9.180  < 2e-16 ***
## Life.Insurance                   0.0142476  0.0008335  17.093  < 2e-16 ***
## Medical.Insurance               -0.0145367  0.0023761  -6.118 9.49e-10 ***
## Average.A.C.Balance             -0.0128568  0.0018769  -6.850 7.39e-12 ***
## Personal.Loan                   -0.0303716  0.0021571 -14.080  < 2e-16 ***
## Investment.in.Mutual.Fund       -0.0027268  0.0011914  -2.289   0.0221 *
## Investment.Tax.Saving.Bond       0.0876990  0.0045077  19.455  < 2e-16 ***
## Home.Loan                       -0.0702954  0.0088648  -7.930 2.20e-15 ***
## Online.Purchase.Amount           0.0509011  0.0020660  24.637  < 2e-16 ***
## self_emp_part_yes                0.3647376  0.1790785   2.037   0.0417 *
## gender_f                        -0.0964252  0.1372215  -0.703   0.4822
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4821.0  on 7069  degrees of freedom
## Residual deviance: 2058.2  on 7056  degrees of freedom
## AIC: 2086.2
##
## Number of Fisher Scoring iterations: 8
```

```
formula(fit)
```

```
## Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +
##     Term.Deposit + Life.Insurance + Medical.Insurance + Average.A.C.Balance +
##     Personal.Loan + Investment.in.Mutual.Fund + Investment.Tax.Saving.Bond +
##     Home.Loan + Online.Purchase.Amount + self_emp_part_yes +
##     gender_f
```

```
fit_final=glm(Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +
                Term.Deposit + Life.Insurance + Medical.Insurance + Average.A.C.Balance +
                Personal.Loan + Investment.in.Mutual.Fund + Investment.Tax.Saving.Bond +
                Home.Loan + Online.Purchase.Amount + self_emp_part_yes +
                gender_f,
              family = "binomial",data=rg_train)
summary(fit_final)
```

```
##
## Call:
## glm(formula = Revenue.Grid ~ Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.in.Mutual.Fund +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     self_emp_part_yes + gender_f, family = "binomial", data = rg_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9746  -0.2364  -0.1545  -0.0726   4.1616
##
## Coefficients:
```

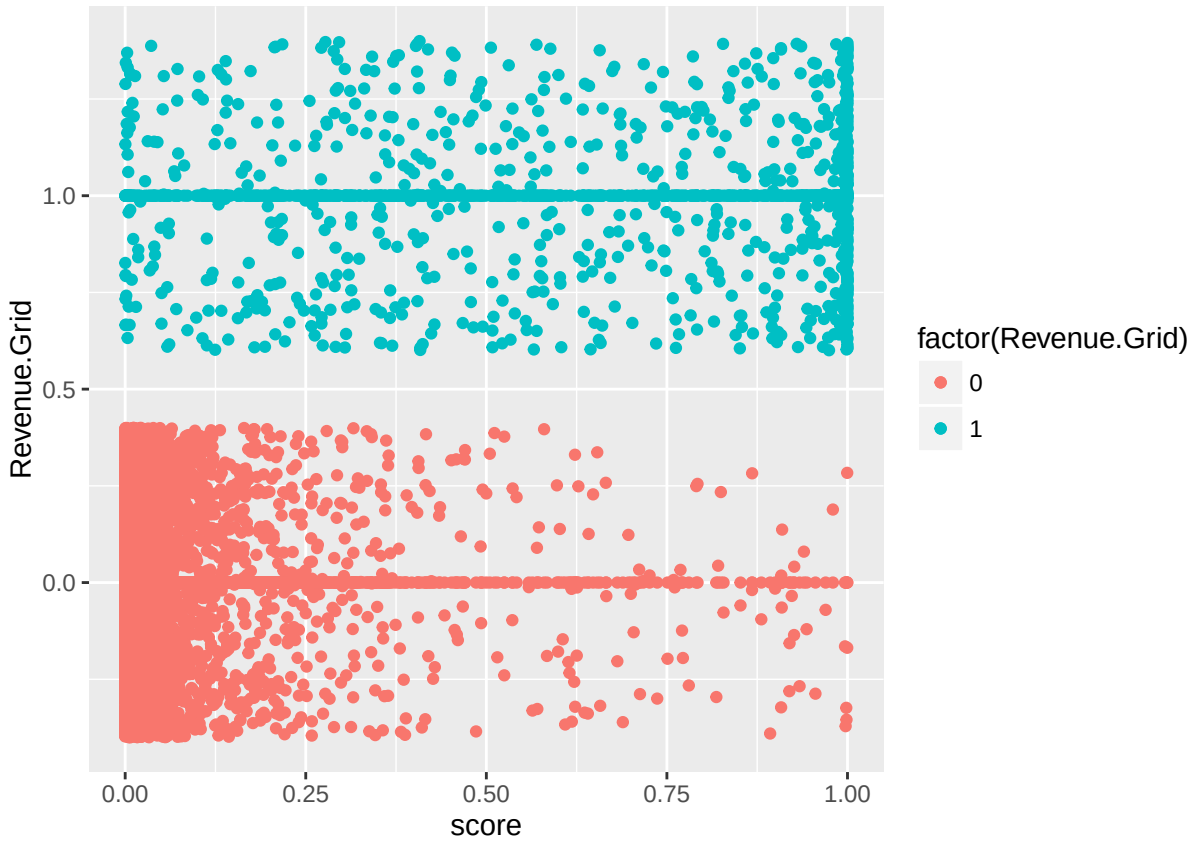
```
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -3.9624737  0.1512250 -26.203  < 2e-16 ***
## Average.Credit.Card.Transaction  0.0232085  0.0010717  21.655  < 2e-16 ***
## Balance.Transfer                -0.0055184  0.0011117  -4.964 6.91e-07 ***
## Term.Deposit                    -0.0162188  0.0017667  -9.180  < 2e-16 ***
## Life.Insurance                   0.0142476  0.0008335  17.093  < 2e-16 ***
## Medical.Insurance               -0.0145367  0.0023761  -6.118 9.49e-10 ***
## Average.A.C.Balance             -0.0128568  0.0018769  -6.850 7.39e-12 ***
## Personal.Loan                   -0.0303716  0.0021571 -14.080  < 2e-16 ***
## Investment.in.Mutual.Fund       -0.0027268  0.0011914  -2.289   0.0221 *
## Investment.Tax.Saving.Bond       0.0876990  0.0045077  19.455  < 2e-16 ***
## Home.Loan                       -0.0702954  0.0088648  -7.930 2.20e-15 ***
## Online.Purchase.Amount           0.0509011  0.0020660  24.637  < 2e-16 ***
## self_emp_part_yes                0.3647376  0.1790785   2.037   0.0417 *
## gender_f                        -0.0964252  0.1372215  -0.703   0.4822
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4821.0  on 7069  degrees of freedom
## Residual deviance: 2058.2  on 7056  degrees of freedom
## AIC: 2086.2
##
## Number of Fisher Scoring iterations: 8
```

You should realise that this final model that we have is not yet predicting the binary outcome. It is merely predicting probabilities yet.

```
rg_train$score=predict(fit_final,newdata=rg_train,type = "response")
```

Modelled probability is P(y=1) by default. Meaning, score should be high when outcome is 1 and low when otucome it 0. Lets visualise how is our eventual binary response is behaving w.r.t. score that we obtained

```
library(ggplot2)
ggplot(rg_train,aes(y=Revenue.Grid,x=score,color=factor(Revenue.Grid)))+
  geom_point()+geom_jitter()
```



You can see that response 0 is bunched around low scores and response 1 is bunched around high scores, However there is overlap as well across score values. We need to find a cutoff in this score so as to reach our business goal. Now remember there are various wasy to consider our business goals, each of them will result in different cutoffs. Lesson is, that there is not one formula to get a fixed cutoff, it depends on what you want to achieve with final model [+ cutoff].

We need to consider a set performance metrics to arrive at the cutoff. This performance criterion can be one of the standard ones or it could be something very specific to your business. One thing to note here is that all these performance metrics will be relying on in one or the other way on missclassification of 1s and 0s. Or in other words , on TP,TN, FP and FN. We'll first see how to get these values for an arbitrary cutoff, then we'll extend that idea to many cutoffs.

Before we go ahead and do this , you need to realise following properties about the `cutoff`

- All the predicted values above cutoff will be 1
- All the predicted values below cutoff will be 0
- Response values above cutoff(predicted 1) which are 1 in reality will be noted as TP
- Response values above cutoff(predicted 1) which are 0 in reality will be noted as FP
- Response values below cutoff(predicted 0) which are 1 in reality will be noted as FN
- Response values below cutoff(predicted 0) which are 0 in reality will be noted as TN

Using these four basic numbers , you will be able to calculate many performance metrics and choose any of them to decide cutoff .

```
cutoff=0.2
predicted=as.numeric(rg_train$score>cutoff)
TP=sum(predicted==1 & rg_train$Revenue.Grid==1)
FP=sum(predicted==1 & rg_train$Revenue.Grid==0)
FN=sum(predicted==0 & rg_train$Revenue.Grid==1)
TN=sum(predicted==0 & rg_train$Revenue.Grid==0)
```

```
# lets also calculate total number of real positives and negatives in the data
P=TP+FN
N=TN+FP

# total number of observations
total=P+N
```

We can calculate all these numbers for a larger sequence of cutoffs also and store these in a data frame using for loops. Lets do that.

```
cutoff_data=data.frame(cutoff=0,TP=0,FP=0,FN=0,TN=0)
cutoffs=seq(0,1,length=100)

for (cutoff in cutoffs){
  predicted=as.numeric(rg_train$score>cutoff)

  TP=sum(predicted==1 & rg_train$Revenue.Grid==1)
  FP=sum(predicted==1 & rg_train$Revenue.Grid==0)
  FN=sum(predicted==0 & rg_train$Revenue.Grid==1)
  TN=sum(predicted==0 & rg_train$Revenue.Grid==0)
  cutoff_data=rbind(cutoff_data,c(cutoff,TP,FP,FN,TN))
}
# lets remove the dummy data cotaining top row
cutoff_data=cutoff_data[-1,]
```

For all these cutoffs now we can consider and calculate many performance metrics. Lets list down some of them.

- *Sensitivity:* Defined as total percentage of positives crorrectly captured by the model or $\frac{TP}{P}$
- *Specificity:* Defined as total percentage of negatives correctly captured by the model or $\frac{TN}{N}$
- *KS:* Defined as absolute difference between cumulative percentage of real postives and real negatives captured by the model as positives or $|\frac{TP}{P} - \frac{FP}{N}|$
- *Accuracy:* Defined as percentage of case classified correctly or $\frac{TP+TN}{total}$
- *Lift:* Defined as ratio of percetnage of positive captured and total percent of population classified as positives by the model or $\frac{\frac{TP}{P}}{\frac{TP+FP}{total}}$

How can we use above metrics to get cutoff ? There are some standard ways to get cutoff associated with these.

- When no explicit business rule is given, it is customary to use KS to decide cutoff. cutoff with maximum value of KS is chosen as cutoff
- In some cases where event and non events are balanced [approximately equal number of 1s and 0s in population], you can decide the value of cutoff which gives you pair of sensitivity and specificity nearest to point (1,1). Or in other words cutoff for which $\sqrt{(1-S_n)^2 + (1-S_p)^2}$ is minimum.
- If for your business restricts percentage of population to be classified as positives , you can use lift as a measure to choose between models. Lift for a stand alone model is not used to decide cutoff.

We'll also discuss a hypothetical business performance metrics in order to emphasize that you can use cusomt performance metrics to get cutoff as well. Lets say for your business it is relatively much more costly to missclassify 1s in comparison to 0s. According to prior experience you'd need to minimise following performance metrics to get cutoff

- *M:* $\frac{(8*FN+2*FP)}{total}$

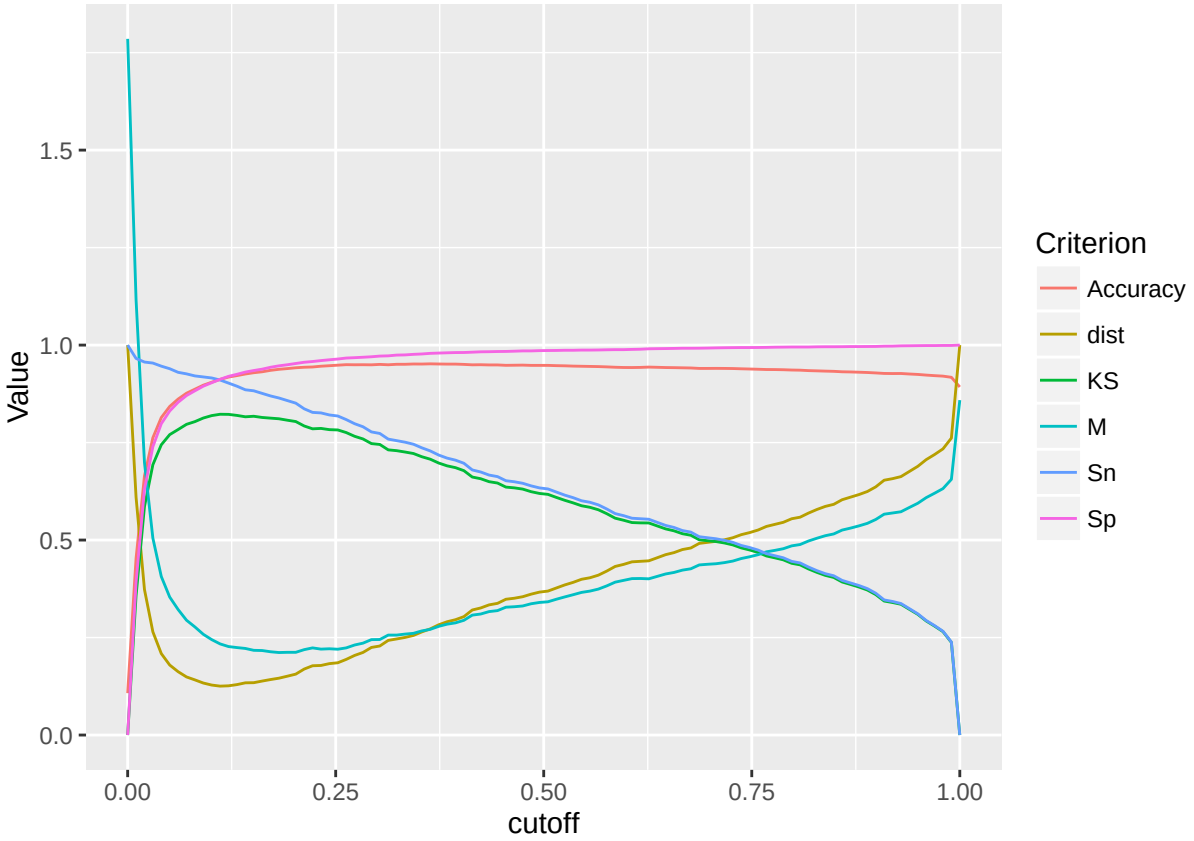
We will calculate all those performance metrics and get cutoff according to them. Lets get to work.

```
cutoff_data=cutoff_data %>%
  mutate(Sn=TP/P, Sp=TN/N,dist=sqrt((1-Sn)**2+(1-Sp)**2),P=FN+TP,N=TN+FP) %>%
  mutate(KS=abs((TP/P)-(FP/N))) %>%
  mutate(Accuracy=(TP+TN)/(P+N)) %>%
  mutate(Lift=(TP/P)/((TP+FP)/(P+N))) %>%
  mutate(M=(8*FN+2*FP)/(P+N)) %>%
  select(-P,-N)
```

Lets visualise these numbers , this will tell you how they behave across differnt cutoff values.

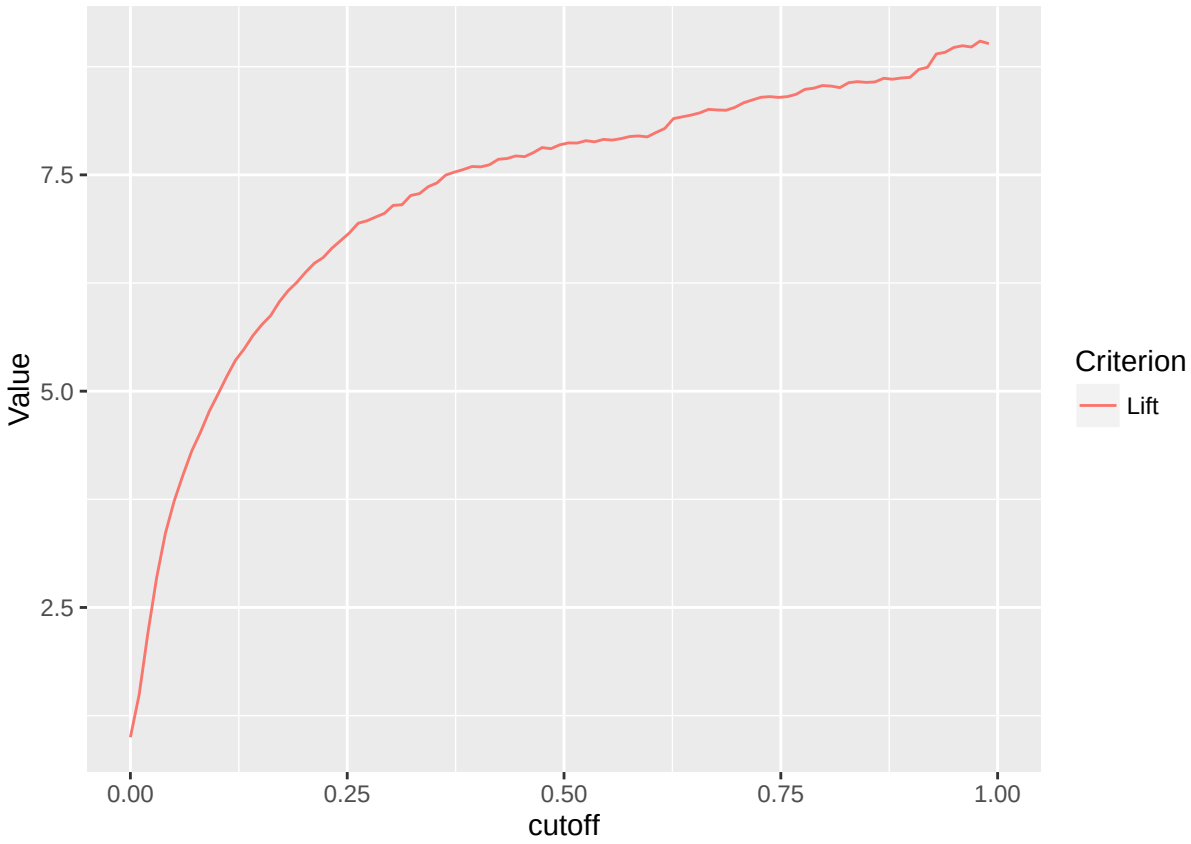
```
library(tidyr)
cutoff_viz=cutoff_data %>%
  select(cutoff,Sn,Sp,dist,KS,Accuracy,Lift,M) %>%
  gather(Criterion,Value,Sn:M)

ggplot(filter(cutoff_viz,Criterion!="Lift"),aes(x=cutoff,y=Value,color=Criterion))+
  geom_line()
```




```
#We'll visualise lift separately because of its scale

cutoff_viz %>%
  filter(Criterion=="Lift") %>%
  ggplot(aes(x=cutoff,y=Value,color=Criterion))+geom_line()
```



You can see that according to different criterion, you'll get different cutoffs. Lets get some get cutoffs and confusion matrix for those cutoffs on test data.

First we'll get scores on the test data.

```
rg_test$score=predict(fit_final,newdata = rg_test,type = "response")
```

Now lets get various cutoffs and corresponding confusion matrix for test data.

```
#Cutoff with minimum KS:
KS_cutoff=cutoff_data$cutoff[which(cutoff_data$KS==max(cutoff_data$KS))][1]
KS_cutoff
```

```
## [1] 0.1111111
```

```
table(rg_test$Revenue.Grid,as.numeric(rg_test$score>KS_cutoff))
```

```
##
##        0    1
##   0 2452  254
##   1   42  282
```

```
#Cutoff with minimum distance
dist_cutoff=cutoff_data$cutoff[which(cutoff_data$dist==min(cutoff_data$dist))][1]
dist_cutoff
```

```
## [1] 0.1111111
```

```
table(rg_test$Revenue.Grid,as.numeric(rg_test$score>dist_cutoff))
```

```
##
##        0    1
```

```
##   0 2452  254
##   1   42  282
```

These numbers are incidentally same, this might not be the case always.

```
#Cutoff with max Accuracy
Acc_cutoff=cutoff_data$cutoff[which(cutoff_data$Accuracy==max(cutoff_data$Accuracy))][1]
Acc_cutoff
```

```
## [1] 0.3636364
```

```
table(rg_test$Revenue.Grid,as.numeric(rg_test$score>Acc_cutoff))
```

```
##
##        0    1
##   0 2641   65
##   1  101  223
```

```
# Cutoff with minimum M ( The hypothetical business criterion)
M_cutoff=cutoff_data$cutoff[which(cutoff_data$M==min(cutoff_data$M))][1]
M_cutoff
```

```
## [1] 0.1818182
```

```
table(rg_test$Revenue.Grid,as.numeric(rg_test$score>M_cutoff))
```

```
##
##        0    1
##   0 2545  161
##   1   60  264
```

Similarly you can get cutoff on score depending on whatever business criterion you need to consider. Remember that if no specific business criterion is given then use KS method to get cutoff.

We'll conclude here.

Prepared by : Lalit Sachan

Contact : lalit.sachan@edvancer.in

In case of any doubts/question regarding content of reading material, please post on QA forum in LMS