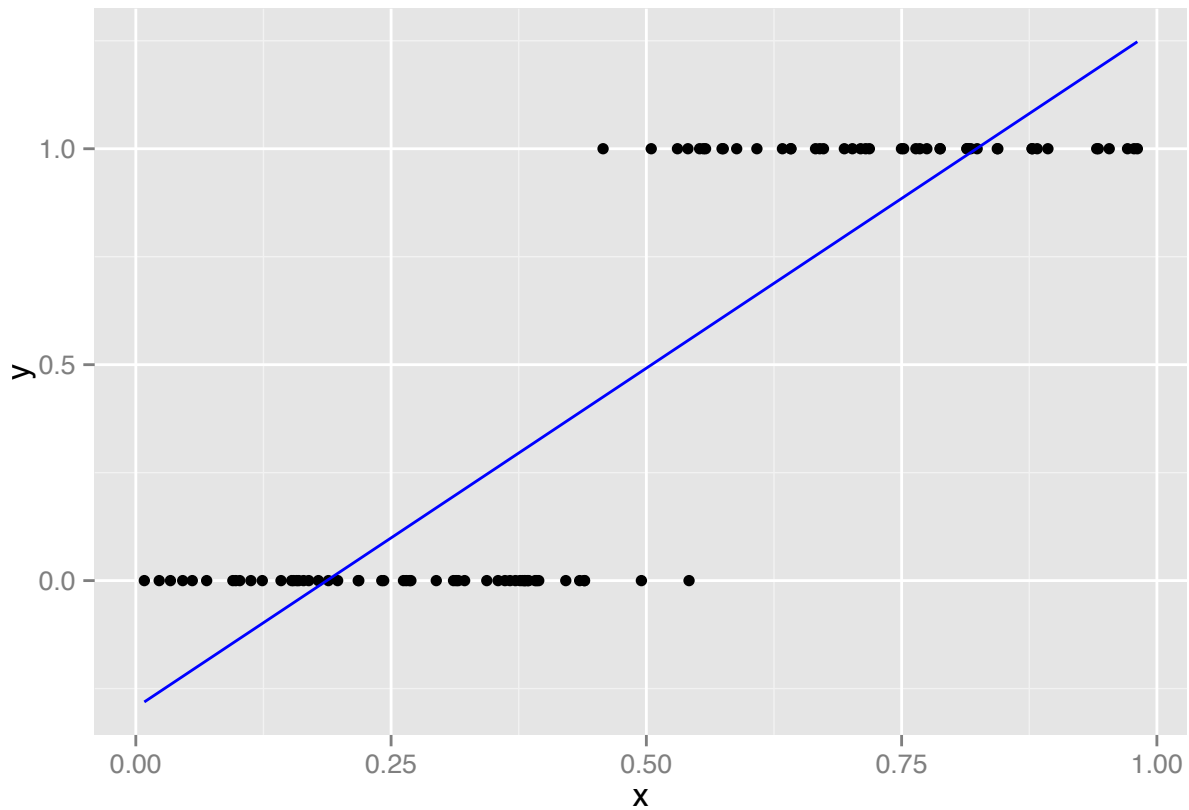
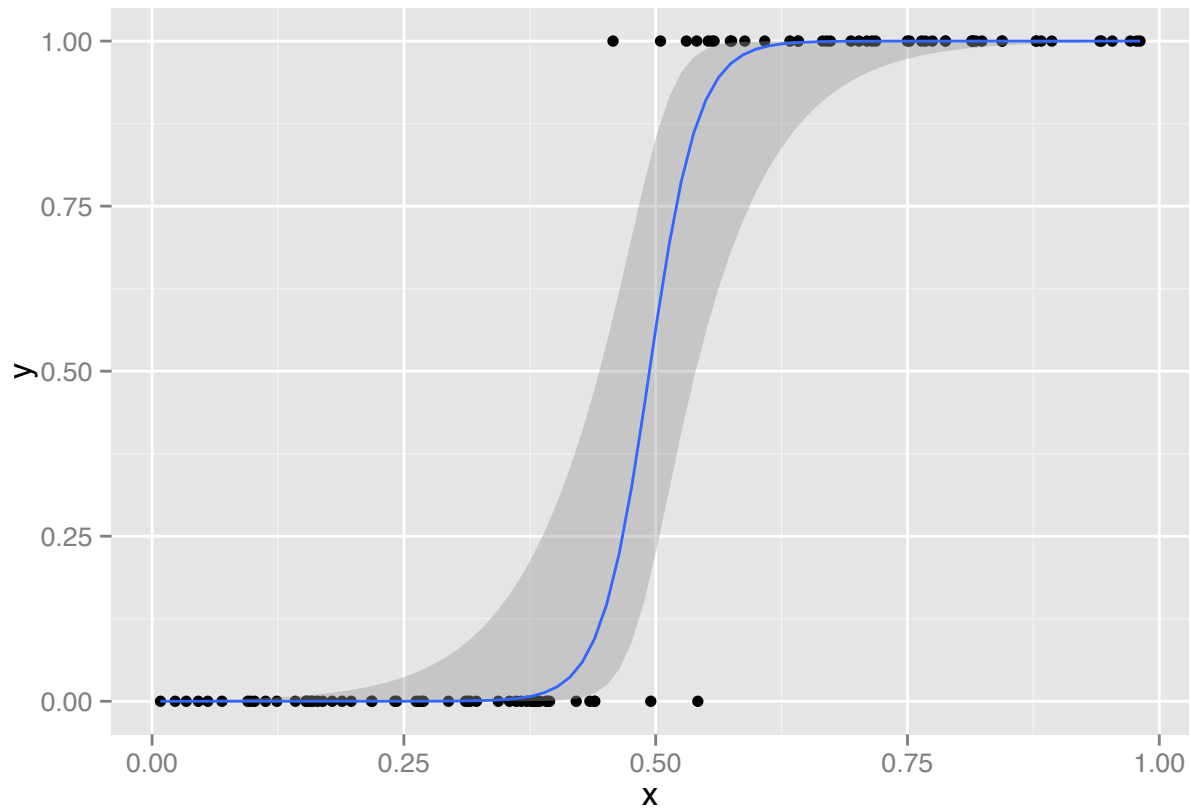


## Logistic Regression

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.



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  $P(y=\text{"yes"})$  or  $P(y=\text{"no"})$ . Let's denote that by just  $p$ . Instead of  $y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 \dots \beta_p * x_p$  we'd model this:

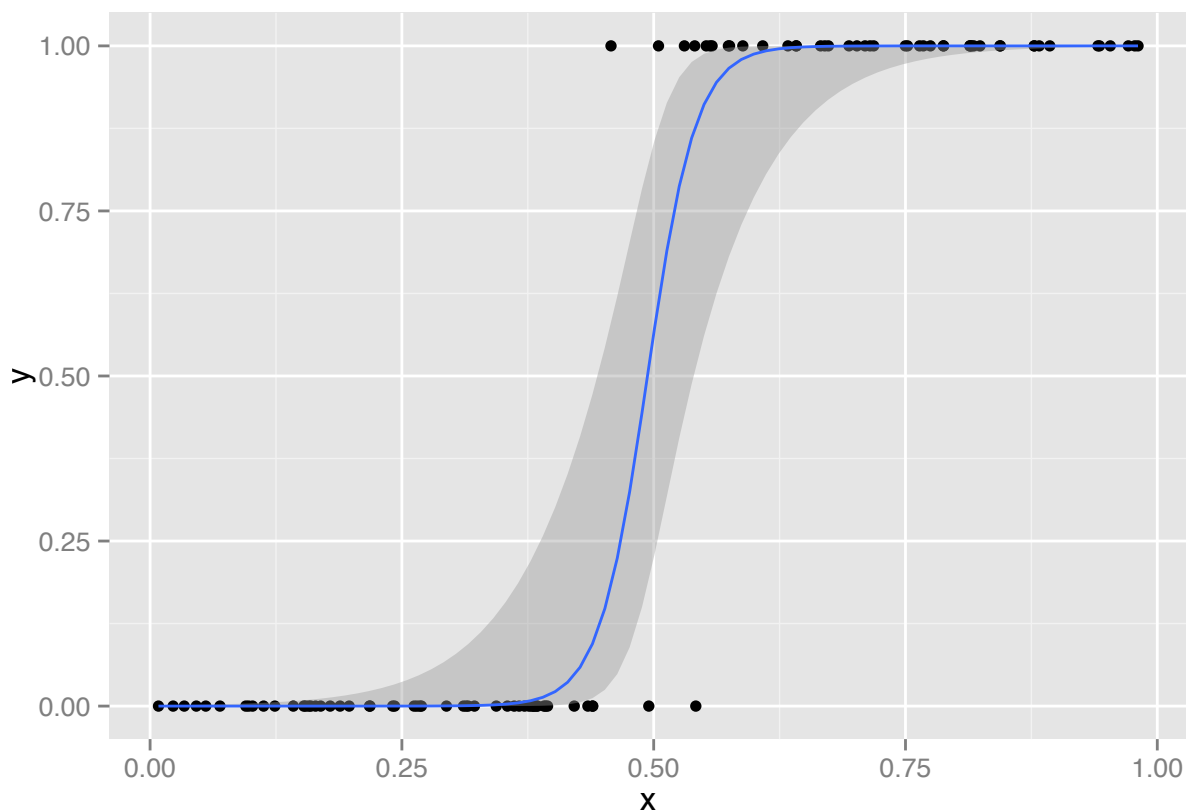
$$p = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 \dots \beta_p * x_p$$

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

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

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 divides 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 let's 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  $[\text{probabilities or } p]$  which enables us to have a good cutoff. Meaning this “score” should be high for one class and low for another.

Let's 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 probabilities as this

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

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  $e^{\beta_i}$ . If  $\beta_i$  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 metrics. 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 reality 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 Positive, 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 following metrics

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

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

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.

**Another way to decide cutoffs** Deciding a cutoff based on ROC curve only works or makes sense when both classes 1s and 0s occur with equal probability in the population, which is never really the case with real data. In that case instead of looking at raw counts , we need to look at percentages of 1s and 0s. To make that point clearer lets look at a hypothetical scenario.

Lets say we were working on building a model to identify whether someone will default on their loan or not. We have build a logistic regression model and have come up with predicted probabilities for an outcome to be 1 [*or score*] for each obseravtion. Next we have sorted our entire data by that score. We had total 1000 observations , out of which 300 obseravtions have outcome 1 and 700 have outcome 0. [*clearly you can see both class outcome dont occur with equal probability in the population*]

Next we divide our data into 10 equal bins containing 100 observations each. And calculate what percenatges of 1s and 0s these bins contain. Lets first look at the raw counts by each bin.

bin	ones	zeroes
1	100	0
2	90	10
3	70	30
4	20	80
5	10	90
6	6	94

bin	ones	zeroes
7	4	96
8	0	100
9	0	100
10	0	100

Next we'll convert these raw counts to percentages of 1s and 0s contained in each bin.

bin	ones	zeroes
1	0.33	0.00
2	0.30	0.01
3	0.23	0.04
4	0.07	0.11
5	0.03	0.13
6	0.02	0.13
7	0.01	0.14
8	0.00	0.14
9	0.00	0.14
10	0.00	0.14

Now think about it this way, if you decide to classify your bin 1 as all 1, you'll end up making no error. When classify entire bin 2 as 1, you'll end up miss-classifying 1% of zeroes. but thats good trade off because you are still capturing higher percentages of 1s in that bin [30%] . But when you reach to bin 4 , it doesn't make sense to classify it as 1s because now you'll be accumulating higher percentages of 0s being classified as 1s in comparison to capturing true ones. Same will be true for subsequent bins.

How does that help us in deciding cutoff. Well, if you remember that these bins were made on a data sorted by our scores. Say that i want to classify bin 1 to 3 as ones and others as zeroes, essentially my cutoff will be score at the boundary of bin 3. lets say that score at the boundry is c.

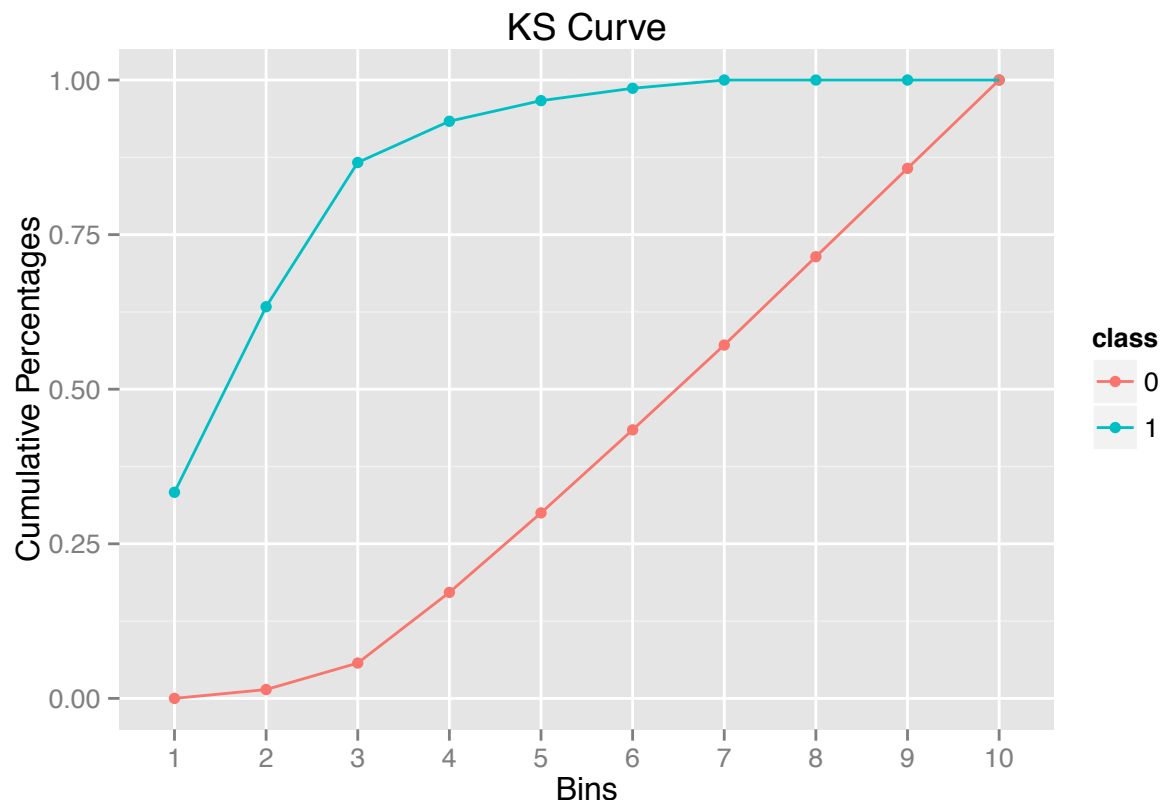
Your decision will be that if an observation has score greater than c then prediciton is 0 otherwise 1. **Remember that these 1s and 0s are just labels that you assign to both of your classes. Depending on how you have sorted the data, labels might flip. Idea here is to be able to differentiate well between two, based on this model**

If you look at cumulative percentages across bins , you'll find that cutoff will be at the boundry of the bin where difference between cumulative percentages of ones and zeros is maximum. Lets check that out.

bin	ones	zeroes	cum_one	cum_zero	Difference
1	0.333	0.000	0.333	0.000	0.333
2	0.300	0.014	0.633	0.014	0.619
3	0.233	0.043	0.867	0.057	0.810
4	0.067	0.114	0.933	0.171	0.762
5	0.033	0.129	0.967	0.300	0.667
6	0.020	0.134	0.987	0.434	0.552
7	0.013	0.137	1.000	0.571	0.429
8	0.000	0.143	1.000	0.714	0.286
9	0.000	0.143	1.000	0.857	0.143
10	0.000	0.143	1.000	1.000	0.000

If we plot these Cumulative percentages of ones and zeroes the resulting Plots is called KS curve. You'll see

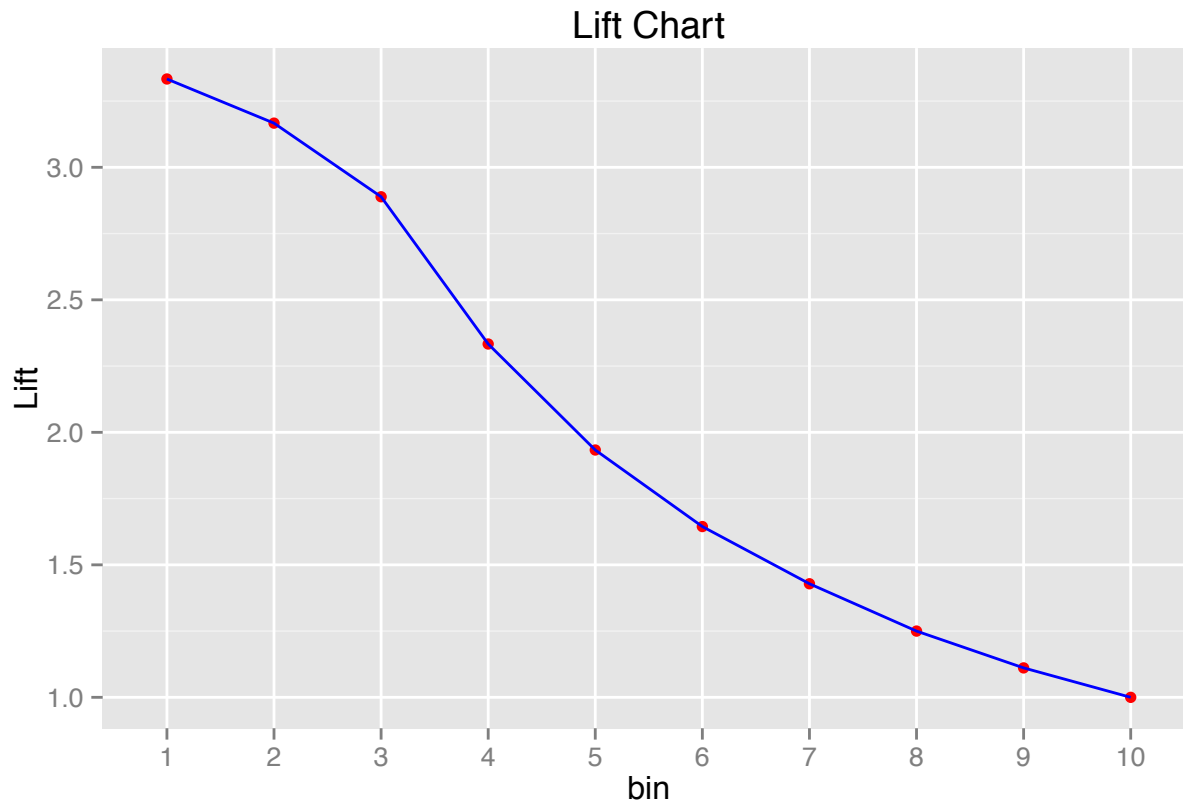
that again maximum distance between two curves will correspond to the bin which we decided to choose cutoff from.



**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 our previous bin sumamry data. Say 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 33% 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 63% 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.

bin	ones	zeroes	Gains	Lift
1	0.33	0.00	0.33	3.33
2	0.30	0.01	0.63	3.17
3	0.23	0.04	0.87	2.89
4	0.07	0.11	0.93	2.33
5	0.03	0.13	0.97	1.93
6	0.02	0.13	0.99	1.64
7	0.01	0.14	1.00	1.43
8	0.00	0.14	1.00	1.25
9	0.00	0.14	1.00	1.11
10	0.00	0.14	1.00	1.00



Here in the context of the campaign 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 facilitation 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 service offering profitable, they charge a percentage based commission 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 money 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 revenue grid category 1]. Let's 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, ...
## $ 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 losing some information by not utilising the fact that each range contains some numbers and there is certainly an order to them. Another possibility is what we discussed earlier, converting these ranges to continuous numeric values by taking average of the range given. Be aware however that we are inflating information 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 across categories and see which one is closest to “unknown” and supply that value instead. Let's 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 interesting 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)
```

Variable `year_last_moved` contains 44 obs where value is 0, which is very different from the context of the other values. It really doesn't correspond to "year" 0. We need to either find out what it can be imputed with or dropped. We are going to drop those observations. [Hint For Imputation: You can see with which year the distribution of Y matches for values 0 and impute the value 0 with that year]

```
rg=rg %>%
  filter(!(year_last_moved==0))
glimpse(rg)
```

```
## Observations: 10,051
## Variables: 40
## $ REF_NO (int) 1, 2, 3, 5, 6, 7, 8, 9, 10, 11...
## $ children (dbl) 0, 0, 0, 0, 0, 0, 0, 1, 0, 2, ...
## $ year_last_moved (int) 1972, 1998, 1996, 1997, 1995, ...
## $ 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, ...
## $ 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, ...
## $ age (dbl) 53.0, 57.5, 28.0, 19.5, 47.5, ...
## $ status_div (dbl) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ status_partner (dbl) 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, ...
## $ status_single (dbl) 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, ...
## $ occ_BM_prof (dbl) 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, ...
## $ occ_Retired (dbl) 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ occ_HW (dbl) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ op_1 (dbl) 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, ...
## $ op_2 (dbl) 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ hs_livein (dbl) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ hs_own (dbl) 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ hs_rent_private (dbl) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ hs_rent_council (dbl) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ fi_1 (dbl) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ fi_2 (dbl) 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ fi_3 (dbl) 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, ...
## $ fi_4 (dbl) 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, ...
## $ self_emp_yes (dbl) 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, ...
## $ self_emp_part_yes (dbl) 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ...
## $ gender_f (dbl) 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, ...
## $ gender_m (dbl) 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, ...
```

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

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

```
rg_test=rg[-s,]

s1=sample(1:nrow(rg_trainval),0.7*nrow(rg_trainval))
rg_train=rg_trainval[s1,]
rg_val=rg_trainval[-s1,]
```

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
##          1.432568e+00          1.306544e+00
## Average.Credit.Card.Transaction      Balance.Transfer
##          1.473872e+07          3.999694e+07
##              Term.Deposit              Life.Insurance
##          1.602129e+07          9.058653e+07
##          Medical.Insurance      Average.A.C.Balance
##          9.787567e+06          1.445710e+07
##          Personal.Loan      Investment.in.Mutual.Fund
##          6.808160e+07          3.246579e+07
##      Investment.Tax.Saving.Bond              Home.Loan
##          1.106377e+06          4.439436e+05
##      Online.Purchase.Amount      Investment.in.Community
##          2.601183e+07          2.727441e+08
##      Investment.in.Equity      Investment.in.Derivative
##          1.683442e+08          2.511723e+08
##      Portfolio.Balance              age
##          1.532566e+01          2.609425e+00
##          status_div      status_partner
##          2.128897e+00          3.738969e+00
##          status_single      occ_BM_prof
##          3.110002e+00          1.438057e+00
##          occ_Retired              occ_HW
##          2.196178e+00          1.252041e+00
##          op_1              op_2
##          1.693787e+00          1.136888e+00
##          hs_livein              hs_own
##          1.039866e+01          5.678062e+01
##          hs_rent_private      hs_rent_council
##          2.353433e+01          2.708035e+01
##          fi_1              fi_2
##          1.024928e+01          3.014130e+01
##          fi_3              fi_4
##          3.703120e+01          2.595086e+01
##          self_emp_yes      self_emp_part_yes
##          1.127022e+00          1.169886e+00
##          gender_f              gender_m
##          6.570269e+01          6.571015e+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
##              1.432402e+00              1.306519e+00
## Average.Credit.Card.Transaction      Balance.Transfer
##              1.328085e+00              2.139019e+00
##              Term.Deposit              Life.Insurance
##              1.645180e+00              3.644647e+07
##              Medical.Insurance      Average.A.C.Balance
##              3.937928e+06              1.445382e+07
##              Personal.Loan      Investment.in.Mutual.Fund
##              6.806614e+07              3.245841e+07
## Investment.Tax.Saving.Bond              Home.Loan
##              1.106125e+06              4.438174e+05
## Online.Purchase.Amount      Investment.in.Equity
##              2.600442e+07              1.682962e+08
## Investment.in.Derivative      Portfolio.Balance
##              2.511700e+08              1.531916e+01
##              age              status_div
##              2.609119e+00              2.128800e+00
##              status_partner      status_single
##              3.738850e+00              3.109976e+00
##              occ_BM_prof              occ_Retired
##              1.438038e+00              2.196169e+00
##              occ_HW              op_1
##              1.251977e+00              1.693778e+00
##              op_2              hs_livein
##              1.136721e+00              1.039866e+01
##              hs_own      hs_rent_private
##              5.677977e+01              2.353429e+01
## hs_rent_council              fi_1
##              2.708023e+01              1.024079e+01
##              fi_2              fi_3
##              3.012926e+01              3.700824e+01
##              fi_4              self_emp_yes
##              2.594191e+01              1.126907e+00
## self_emp_part_yes              gender_f
##              1.169572e+00              6.569408e+01
##              gender_m
##              6.570579e+01
```

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

```
##              children              year_last_moved
##              1.432142e+00              1.306506e+00
## Average.Credit.Card.Transaction      Balance.Transfer
##              1.327785e+00              2.138971e+00
##              Term.Deposit              Life.Insurance
##              1.644965e+00              3.691402e+00
```



##	Medical.Insurance	Average.A.C.Balance
##	1.740875e+00	8.261041e+06
##	Personal.Loan	Investment.in.Mutual.Fund
##	3.890234e+07	1.855109e+07
##	Investment.Tax.Saving.Bond	Home.Loan
##	6.322020e+05	4.382378e+05
##	Online.Purchase.Amount	Investment.in.Equity
##	2.567739e+07	1.661798e+08
##	Portfolio.Balance	age
##	1.531915e+01	2.608509e+00
##	status_div	status_partner
##	2.128766e+00	3.738848e+00
##	status_single	occ_BM_prof
##	3.108502e+00	1.436037e+00
##	occ_Retired	occ_HW
##	2.195376e+00	1.251848e+00
##	op_1	op_2
##	1.693621e+00	1.136707e+00
##	hs_livein	hs_own
##	1.039567e+01	5.677473e+01
##	hs_rent_private	hs_rent_council
##	2.353183e+01	2.707421e+01
##	fi_1	fi_2
##	1.024068e+01	3.012544e+01
##	fi_3	fi_4
##	3.699864e+01	2.593468e+01
##	self_emp_yes	self_emp_part_yes
##	1.126058e+00	1.169409e+00
##	gender_f	gender_m
##	6.568558e+01	6.570136e+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
##	1.431690	1.305286
##	Average.Credit.Card.Transaction	Balance.Transfer
##	1.327447	2.137053
##	Term.Deposit	Life.Insurance
##	1.643659	3.690627
##	Medical.Insurance	Average.A.C.Balance
##	1.740833	1.831656
##	Personal.Loan	Investment.in.Mutual.Fund
##	2.814180	2.511791
##	Investment.Tax.Saving.Bond	Home.Loan
##	1.248886	1.231317
##	Online.Purchase.Amount	Portfolio.Balance
##	1.307591	15.304304
##	age	status_div
##	2.608506	2.128457
##	status_partner	status_single
##	3.738655	3.108082
##	occ_BM_prof	occ_Retired

```
##          1.436028          2.195365
##          occ_HW          op_1
##          1.251464          1.693619
##          op_2          hs_livein
##          1.136675          10.395643
##          hs_own          hs_rent_private
##          56.769887          23.531212
##          hs_rent_council          fi_1
##          27.074196          10.237074
##          fi_2          fi_3
##          30.112766          36.977702
##          fi_4          self_emp_yes
##          25.924999          1.126045
##          self_emp_part_yes          gender_f
##          1.169367          65.685552
##          gender_m
##          65.701271
```

```
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
##          1.431569          1.305106
## Average.Credit.Card.Transaction          Balance.Transfer
##          1.326576          2.135265
##          Term.Deposit          Life.Insurance
##          1.643656          3.689909
##          Medical.Insurance          Average.A.C.Balance
##          1.740038          1.831273
##          Personal.Loan          Investment.in.Mutual.Fund
##          2.810706          2.508009
##          Investment.Tax.Saving.Bond          Home.Loan
##          1.248862          1.231207
##          Online.Purchase.Amount          Portfolio.Balance
##          1.307300          15.270095
##          age          status_div
##          2.608456          2.128441
##          status_partner          status_single
##          3.738556          3.107993
##          occ_BM_prof          occ_Retired
##          1.436023          2.194254
##          occ_HW          op_1
##          1.251462          1.693603
##          op_2          hs_livein
##          1.136300          10.395601
##          hs_own          hs_rent_private
##          56.769816          23.530786
##          hs_rent_council          fi_1
##          27.073603          10.237050
##          fi_2          fi_3
##          30.112539          36.977537
##          fi_4          self_emp_yes
##          25.924836          1.125940
```

##	self_emp_part_yes	gender_f
##	1.167724	1.197094

```
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
##	1.431114	1.305104
##	Average.Credit.Card.Transaction	Balance.Transfer
##	1.326502	2.133806
##	Term.Deposit	Life.Insurance
##	1.642748	3.689420
##	Medical.Insurance	Average.A.C.Balance
##	1.738938	1.830770
##	Personal.Loan	Investment.in.Mutual.Fund
##	2.810684	2.507702
##	Investment.Tax.Saving.Bond	Home.Loan
##	1.248812	1.231070
##	Online.Purchase.Amount	Portfolio.Balance
##	1.307156	15.267269
##	age	status_div
##	2.604134	2.128269
##	status_partner	status_single
##	3.738555	3.107929
##	occ_BM_prof	occ_Retired
##	1.436009	2.194189
##	occ_HW	op_1
##	1.251428	1.693311
##	op_2	hs_livein
##	1.136164	1.095025
##	hs_rent_private	hs_rent_council
##	1.022161	1.053395
##	fi_1	fi_2
##	10.186487	29.930476
##	fi_3	fi_4
##	36.767477	25.788048
##	self_emp_yes	self_emp_part_yes
##	1.125870	1.167642
##	gender_f	
##	1.196947	

```
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
##	1.431098	1.302295
##	Average.Credit.Card.Transaction	Balance.Transfer
##	1.325942	2.133770
##	Term.Deposit	Life.Insurance
##	1.641639	3.689374
##	Medical.Insurance	Average.A.C.Balance

##	1.737680	1.830546
##	Personal.Loan	Investment.in.Mutual.Fund
##	2.810567	2.506692
##	Investment.Tax.Saving.Bond	Home.Loan
##	1.248811	1.231059
##	Online.Purchase.Amount	Portfolio.Balance
##	1.307155	15.266940
##	age	status_div
##	2.604017	2.127948
##	status_partner	status_single
##	3.738351	3.107662
##	occ_BM_prof	occ_Retired
##	1.435751	2.194104
##	occ_HW	op_1
##	1.250854	1.691627
##	op_2	hs_livein
##	1.136122	1.086214
##	hs_rent_private	hs_rent_council
##	1.022156	1.053335
##	fi_1	fi_2
##	1.308707	1.203479
##	fi_4	self_emp_yes
##	1.200903	1.125850
##	self_emp_part_yes	gender_f
##	1.167622	1.196687

```
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
##	1.430504	1.302050
##	Average.Credit.Card.Transaction	Balance.Transfer
##	1.229205	1.812190
##	Term.Deposit	Life.Insurance
##	1.514724	2.232143
##	Medical.Insurance	Average.A.C.Balance
##	1.550344	1.593145
##	Personal.Loan	Investment.in.Mutual.Fund
##	1.527333	1.899247
##	Investment.Tax.Saving.Bond	Home.Loan
##	1.223423	1.228716
##	Online.Purchase.Amount	age
##	1.120096	2.603819
##	status_div	status_partner
##	2.126995	3.737882
##	status_single	occ_BM_prof
##	3.106242	1.435654
##	occ_Retired	occ_HW
##	2.191056	1.250843
##	op_1	op_2
##	1.690729	1.136053
##	hs_livein	hs_rent_private
##	1.086214	1.022101

```
##                hs_rent_council                fi_1
##                1.052893                1.307419
##                fi_2                fi_4
##                1.203455                1.200628
##                self_emp_yes                self_emp_part_yes
##                1.125840                1.167321
##                gender_f
##                1.196612
```

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_trainval=rg[s,]
rg_test=rg[-s,]

s1=sample(1:nrow(rg_trainval),0.7*nrow(rg_trainval))
rg_train=rg_trainval[s1,]
rg_val=rg_trainval[-s1,]

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=1469.39
## 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
## - children      1  1405.4 1467.4
## - occ_Retired    1  1405.4 1467.4
## - fi_1           1  1405.5 1467.5
## - self_emp_yes   1  1405.5 1467.5
## - op_2           1  1405.6 1467.6
## - year_last_moved 1  1405.6 1467.6
## - hs_rent_private 1  1405.7 1467.7
## - age            1  1405.7 1467.7
## - op_1           1  1405.8 1467.8
## - occ_BM_prof     1  1405.8 1467.8
## - self_emp_part_yes 1  1406.1 1468.1
## - status_div      1  1406.1 1468.1
## - occ_HW          1  1406.6 1468.6
## - hs_rent_council 1  1407.1 1469.1
## - fi_2           1  1407.2 1469.2
## <none>           1405.4 1469.4
## - fi_4           1  1407.5 1469.5
## - status_single   1  1407.7 1469.7
## - gender_f        1  1408.1 1470.1
## - status_partner   1  1408.1 1470.1
## - hs_livein       1  1408.8 1470.8
## - Balance.Transfer 1  1415.2 1477.2
## - Investment.in.Mutual.Fund 1  1415.4 1477.4
## - Medical.Insurance 1  1435.7 1497.7
## - Home.Loan       1  1455.7 1517.7
## - Average.A.C.Balance 1  1465.0 1527.0
## - Term.Deposit     1  1484.6 1546.6
## - Life.Insurance   1  1666.0 1728.0
## - Personal.Loan    1  1674.2 1736.2
## - Investment.Tax.Saving.Bond 1  1685.1 1747.1
## - Average.Credit.Card.Transaction 1  1739.4 1801.4
## - Online.Purchase.Amount 1  2491.1 2553.1
##
## Step:  AIC=1467.39
## 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 +
##      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
## - occ_Retired      1  1405.5 1465.5
## - fi_1             1  1405.5 1465.5
## - self_emp_yes     1  1405.5 1465.5
## - op_2             1  1405.6 1465.6
## - year_last_moved  1  1405.6 1465.6
## - hs_rent_private   1  1405.7 1465.7
## - op_1             1  1405.8 1465.8
## - occ_BM_prof       1  1405.8 1465.8

```

```

## - age 1 1405.8 1465.8
## - self_emp_part_yes 1 1406.1 1466.1
## - status_div 1 1406.1 1466.1
## - occ_HW 1 1406.6 1466.6
## - hs_rent_council 1 1407.1 1467.1
## - fi_2 1 1407.2 1467.2
## <none> 1405.4 1467.4
## - fi_4 1 1407.5 1467.5
## - status_single 1 1407.9 1467.9
## - gender_f 1 1408.1 1468.1
## - status_partner 1 1408.1 1468.1
## - hs_livein 1 1408.8 1468.8
## - Balance.Transfer 1 1415.2 1475.2
## - Investment.in.Mutual.Fund 1 1415.4 1475.4
## - Medical.Insurance 1 1435.8 1495.8
## - Home.Loan 1 1455.7 1515.7
## - Average.A.C.Balance 1 1465.0 1525.0
## - Term.Deposit 1 1484.6 1544.6
## - Life.Insurance 1 1666.4 1726.4
## - Personal.Loan 1 1675.2 1735.2
## - Investment.Tax.Saving.Bond 1 1685.2 1745.2
## - Average.Credit.Card.Transaction 1 1739.6 1799.6
## - Online.Purchase.Amount 1 2491.4 2551.4
##
## Step: AIC=1465.45
## 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 +
## age + status_div + status_partner + status_single + occ_BM_prof +
## 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
## - fi_1 1 1405.6 1463.6
## - self_emp_yes 1 1405.6 1463.6
## - year_last_moved 1 1405.7 1463.7
## - op_2 1 1405.7 1463.7
## - hs_rent_private 1 1405.7 1463.7
## - op_1 1 1405.8 1463.8
## - occ_BM_prof 1 1405.8 1463.8
## - age 1 1406.1 1464.1
## - self_emp_part_yes 1 1406.2 1464.2
## - status_div 1 1406.2 1464.2
## - occ_HW 1 1406.6 1464.6
## - hs_rent_council 1 1407.2 1465.2
## - fi_2 1 1407.3 1465.3
## <none> 1405.5 1465.5
## - fi_4 1 1407.5 1465.5
## - status_single 1 1407.9 1465.9
## - status_partner 1 1408.2 1466.2
## - gender_f 1 1408.2 1466.2
## - hs_livein 1 1408.8 1466.8
## - Balance.Transfer 1 1415.3 1473.3

```

```

## - Investment.in.Mutual.Fund      1  1415.5 1473.5
## - Medical.Insurance              1  1435.8 1493.8
## - Home.Loan                      1  1455.7 1513.7
## - Average.A.C.Balance            1  1465.1 1523.1
## - Term.Deposit                   1  1484.8 1542.8
## - Life.Insurance                 1  1666.5 1724.5
## - Personal.Loan                  1  1675.4 1733.4
## - Investment.Tax.Saving.Bond     1  1685.7 1743.7
## - Average.Credit.Card.Transaction 1  1739.6 1797.6
## - Online.Purchase.Amount         1  2491.6 2549.6
##
## Step: AIC=1463.56
## 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 +
##     age + status_div + status_partner + status_single + occ_BM_prof +
##     occ_HW + op_1 + op_2 + hs_livein + hs_rent_private + hs_rent_council +
##     fi_2 + fi_4 + self_emp_yes + self_emp_part_yes + gender_f
##
##                                     Df Deviance    AIC
## - self_emp_yes                     1  1405.7 1461.7
## - op_2                             1  1405.8 1461.8
## - year_last_moved                  1  1405.8 1461.8
## - hs_rent_private                  1  1405.8 1461.8
## - occ_BM_prof                      1  1405.9 1461.9
## - op_1                             1  1405.9 1461.9
## - status_div                       1  1406.2 1462.2
## - self_emp_part_yes                1  1406.2 1462.2
## - age                             1  1406.3 1462.3
## - occ_HW                           1  1406.7 1462.7
## - hs_rent_council                  1  1407.2 1463.2
## - fi_2                             1  1407.3 1463.3
## - fi_4                             1  1407.5 1463.5
## <none>                             1405.6 1463.6
## - status_single                    1  1407.9 1463.9
## - status_partner                   1  1408.2 1464.2
## - gender_f                         1  1408.3 1464.3
## - hs_livein                        1  1409.0 1465.0
## - Balance.Transfer                 1  1415.3 1471.3
## - Investment.in.Mutual.Fund        1  1415.7 1471.7
## - Medical.Insurance                1  1436.0 1492.0
## - Home.Loan                       1  1455.9 1511.9
## - Average.A.C.Balance              1  1465.1 1521.1
## - Term.Deposit                     1  1484.8 1540.8
## - Life.Insurance                   1  1666.6 1722.6
## - Personal.Loan                    1  1675.4 1731.4
## - Investment.Tax.Saving.Bond       1  1685.8 1741.8
## - Average.Credit.Card.Transaction 1  1739.9 1795.9
## - Online.Purchase.Amount           1  2492.4 2548.4
##
## Step: AIC=1461.69
## 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 +
##      age + status_div + status_partner + status_single + occ_BM_prof +
##      occ_HW + op_1 + op_2 + hs_livein + hs_rent_private + hs_rent_council +
##      fi_2 + fi_4 + self_emp_part_yes + gender_f
##
##
##              Df Deviance    AIC
## - op_2              1   1405.9 1459.9
## - year_last_moved    1   1405.9 1459.9
## - hs_rent_private    1   1406.0 1460.0
## - op_1              1   1406.0 1460.0
## - occ_BM_prof        1   1406.0 1460.0
## - status_div         1   1406.4 1460.4
## - age               1   1406.4 1460.4
## - self_emp_part_yes  1   1406.6 1460.6
## - occ_HW            1   1406.8 1460.8
## - hs_rent_council    1   1407.3 1461.3
## - fi_2              1   1407.4 1461.4
## - fi_4              1   1407.6 1461.6
## <none>              1405.7 1461.7
## - status_single      1   1408.1 1462.1
## - status_partner     1   1408.3 1462.3
## - gender_f           1   1408.6 1462.6
## - hs_livein          1   1409.2 1463.2
## - Balance.Transfer   1   1415.4 1469.4
## - Investment.in.Mutual.Fund 1   1416.0 1470.0
## - Medical.Insurance  1   1436.2 1490.2
## - Home.Loan          1   1455.9 1509.9
## - Average.A.C.Balance 1   1465.5 1519.5
## - Term.Deposit       1   1485.1 1539.1
## - Life.Insurance     1   1667.0 1721.0
## - Personal.Loan      1   1675.9 1729.9
## - Investment.Tax.Saving.Bond 1   1685.9 1739.9
## - Average.Credit.Card.Transaction 1   1740.4 1794.4
## - Online.Purchase.Amount 1   2493.9 2547.9
##
## Step:  AIC=1459.88
## 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 +
##      age + status_div + status_partner + status_single + occ_BM_prof +
##      occ_HW + op_1 + hs_livein + hs_rent_private + hs_rent_council +
##      fi_2 + fi_4 + self_emp_part_yes + gender_f
##
##
##              Df Deviance    AIC
## - op_1              1   1406.1 1458.1
## - year_last_moved    1   1406.1 1458.1
## - hs_rent_private    1   1406.2 1458.2
## - occ_BM_prof        1   1406.2 1458.2
## - status_div         1   1406.6 1458.6
## - age               1   1406.6 1458.6
## - self_emp_part_yes  1   1406.9 1458.9
## - occ_HW            1   1407.0 1459.0

```

```

## - fi_2 1 1407.5 1459.5
## - hs_rent_council 1 1407.5 1459.5
## - fi_4 1 1407.7 1459.7
## <none> 1405.9 1459.9
## - status_single 1 1408.2 1460.2
## - status_partner 1 1408.6 1460.6
## - gender_f 1 1408.6 1460.6
## - hs_livein 1 1409.3 1461.3
## - Balance.Transfer 1 1415.7 1467.7
## - Investment.in.Mutual.Fund 1 1416.0 1468.0
## - Medical.Insurance 1 1436.3 1488.3
## - Home.Loan 1 1455.9 1507.9
## - Average.A.C.Balance 1 1465.8 1517.8
## - Term.Deposit 1 1485.2 1537.2
## - Life.Insurance 1 1667.3 1719.3
## - Personal.Loan 1 1676.4 1728.4
## - Investment.Tax.Saving.Bond 1 1686.6 1738.6
## - Average.Credit.Card.Transaction 1 1740.5 1792.5
## - Online.Purchase.Amount 1 2494.0 2546.0
##
## Step: AIC=1458.09
## 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 +
## age + status_div + status_partner + status_single + occ_BM_prof +
## occ_HW + hs_livein + hs_rent_private + hs_rent_council +
## fi_2 + fi_4 + self_emp_part_yes + gender_f
##
## Df Deviance AIC
## - year_last_moved 1 1406.3 1456.3
## - hs_rent_private 1 1406.4 1456.4
## - occ_BM_prof 1 1406.5 1456.5
## - age 1 1406.6 1456.6
## - status_div 1 1406.7 1456.7
## - self_emp_part_yes 1 1407.1 1457.1
## - occ_HW 1 1407.3 1457.3
## - fi_2 1 1407.7 1457.7
## - hs_rent_council 1 1407.7 1457.7
## - fi_4 1 1408.0 1458.0
## <none> 1406.1 1458.1
## - status_single 1 1408.4 1458.4
## - gender_f 1 1408.9 1458.9
## - status_partner 1 1409.1 1459.1
## - hs_livein 1 1409.6 1459.6
## - Balance.Transfer 1 1415.9 1465.9
## - Investment.in.Mutual.Fund 1 1416.2 1466.2
## - Medical.Insurance 1 1436.6 1486.6
## - Home.Loan 1 1456.0 1506.0
## - Average.A.C.Balance 1 1465.9 1515.9
## - Term.Deposit 1 1485.4 1535.4
## - Life.Insurance 1 1667.4 1717.4
## - Personal.Loan 1 1676.5 1726.5
## - Investment.Tax.Saving.Bond 1 1686.6 1736.6

```

```

## - Average.Credit.Card.Transaction 1 1741.0 1791.0
## - Online.Purchase.Amount 1 2496.4 2546.4
##
## Step: AIC=1456.34
## 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 + age + status_div + status_partner +
## status_single + occ_BM_prof + occ_HW + hs_livein + hs_rent_private +
## hs_rent_council + fi_2 + fi_4 + self_emp_part_yes + gender_f
##
##
## Df Deviance AIC
## - hs_rent_private 1 1406.6 1454.6
## - occ_BM_prof 1 1406.7 1454.7
## - status_div 1 1407.0 1455.0
## - self_emp_part_yes 1 1407.3 1455.3
## - age 1 1407.5 1455.5
## - occ_HW 1 1407.6 1455.6
## - fi_2 1 1407.9 1455.9
## - hs_rent_council 1 1408.0 1456.0
## - fi_4 1 1408.2 1456.2
## <none> 1406.3 1456.3
## - status_single 1 1408.6 1456.6
## - gender_f 1 1409.1 1457.1
## - status_partner 1 1409.4 1457.4
## - hs_livein 1 1409.8 1457.8
## - Balance.Transfer 1 1416.1 1464.1
## - Investment.in.Mutual.Fund 1 1416.4 1464.4
## - Medical.Insurance 1 1436.8 1484.8
## - Home.Loan 1 1456.4 1504.4
## - Average.A.C.Balance 1 1466.3 1514.3
## - Term.Deposit 1 1485.9 1533.9
## - Life.Insurance 1 1667.6 1715.6
## - Personal.Loan 1 1676.9 1724.9
## - Investment.Tax.Saving.Bond 1 1687.1 1735.1
## - Average.Credit.Card.Transaction 1 1741.6 1789.6
## - Online.Purchase.Amount 1 2496.7 2544.7
##
## Step: AIC=1454.6
## 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 + age + status_div + status_partner +
## status_single + occ_BM_prof + occ_HW + hs_livein + hs_rent_council +
## fi_2 + fi_4 + self_emp_part_yes + gender_f
##
##
## Df Deviance AIC
## - occ_BM_prof 1 1407.0 1453.0
## - status_div 1 1407.2 1453.2
## - self_emp_part_yes 1 1407.6 1453.6
## - age 1 1407.7 1453.7
## - occ_HW 1 1407.8 1453.8
## - fi_2 1 1408.2 1454.2
## - hs_rent_council 1 1408.3 1454.3

```

```

## - fi_4 1 1408.5 1454.5
## <none> 1406.6 1454.6
## - status_single 1 1409.0 1455.0
## - gender_f 1 1409.4 1455.4
## - status_partner 1 1409.7 1455.7
## - hs_livein 1 1410.1 1456.1
## - Balance.Transfer 1 1416.4 1462.4
## - Investment.in.Mutual.Fund 1 1416.6 1462.6
## - Medical.Insurance 1 1437.2 1483.2
## - Home.Loan 1 1456.5 1502.5
## - Average.A.C.Balance 1 1466.7 1512.7
## - Term.Deposit 1 1486.2 1532.2
## - Life.Insurance 1 1667.6 1713.6
## - Personal.Loan 1 1677.8 1723.8
## - Investment.Tax.Saving.Bond 1 1687.1 1733.1
## - Average.Credit.Card.Transaction 1 1741.7 1787.7
## - Online.Purchase.Amount 1 2497.3 2543.3
##
## Step: AIC=1452.97
## 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 + age + status_div + status_partner +
## status_single + occ_HW + hs_livein + hs_rent_council + fi_2 +
## fi_4 + self_emp_part_yes + gender_f
##
## Df Deviance AIC
## - status_div 1 1407.7 1451.7
## - age 1 1407.8 1451.8
## - occ_HW 1 1408.0 1452.0
## - self_emp_part_yes 1 1408.0 1452.0
## - fi_2 1 1408.5 1452.5
## - hs_rent_council 1 1408.8 1452.8
## - fi_4 1 1408.8 1452.8
## <none> 1407.0 1453.0
## - status_single 1 1409.5 1453.5
## - status_partner 1 1410.1 1454.1
## - gender_f 1 1410.4 1454.4
## - hs_livein 1 1410.7 1454.7
## - Balance.Transfer 1 1416.7 1460.7
## - Investment.in.Mutual.Fund 1 1417.0 1461.0
## - Medical.Insurance 1 1437.5 1481.5
## - Home.Loan 1 1457.1 1501.1
## - Average.A.C.Balance 1 1466.9 1510.9
## - Term.Deposit 1 1486.6 1530.6
## - Life.Insurance 1 1667.7 1711.7
## - Personal.Loan 1 1677.8 1721.8
## - Investment.Tax.Saving.Bond 1 1687.8 1731.8
## - Average.Credit.Card.Transaction 1 1741.8 1785.8
## - Online.Purchase.Amount 1 2500.4 2544.4
##
## Step: AIC=1451.67
## 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 + age + status_partner +
##      status_single + occ_HW + hs_livein + hs_rent_council + fi_2 +
##      fi_4 + self_emp_part_yes + gender_f
##
##
##      Df Deviance      AIC
## - age                1  1408.3 1450.3
## - occ_HW              1  1408.6 1450.6
## - self_emp_part_yes   1  1408.7 1450.7
## - fi_2                1  1409.2 1451.2
## - hs_rent_council     1  1409.5 1451.5
## - status_single       1  1409.5 1451.5
## - fi_4                1  1409.6 1451.6
## <none>                1407.7 1451.7
## - status_partner      1  1410.5 1452.5
## - gender_f            1  1411.2 1453.2
## - hs_livein           1  1411.3 1453.3
## - Balance.Transfer    1  1417.3 1459.3
## - Investment.in.Mutual.Fund 1  1417.6 1459.6
## - Medical.Insurance   1  1438.2 1480.2
## - Home.Loan           1  1458.0 1500.0
## - Average.A.C.Balance 1  1468.2 1510.2
## - Term.Deposit        1  1487.3 1529.3
## - Life.Insurance      1  1669.1 1711.1
## - Personal.Loan       1  1678.8 1720.8
## - Investment.Tax.Saving.Bond 1  1688.7 1730.7
## - Average.Credit.Card.Transaction 1  1742.8 1784.8
## - Online.Purchase.Amount 1  2503.1 2545.1
##
## Step:  AIC=1450.26
## 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 + status_partner + status_single +
##      occ_HW + hs_livein + hs_rent_council + fi_2 + fi_4 + self_emp_part_yes +
##      gender_f
##
##
##      Df Deviance      AIC
## - occ_HW                1  1409.2 1449.2
## - self_emp_part_yes     1  1409.2 1449.2
## - fi_2                  1  1409.7 1449.7
## - status_single         1  1409.7 1449.7
## - hs_rent_council       1  1410.0 1450.0
## - fi_4                  1  1410.0 1450.0
## <none>                  1408.3 1450.3
## - status_partner        1  1410.9 1450.9
## - gender_f              1  1412.0 1452.0
## - hs_livein             1  1412.0 1452.0
## - Balance.Transfer       1  1418.0 1458.0
## - Investment.in.Mutual.Fund 1  1418.2 1458.2
## - Medical.Insurance     1  1438.7 1478.7
## - Home.Loan             1  1458.5 1498.5
## - Average.A.C.Balance   1  1468.8 1508.8
## - Term.Deposit          1  1488.4 1528.4

```

```

## - Life.Insurance          1  1669.6 1709.6
## - Personal.Loan          1  1685.6 1725.6
## - Investment.Tax.Saving.Bond 1  1689.3 1729.3
## - Average.Credit.Card.Transaction 1  1743.4 1783.4
## - Online.Purchase.Amount  1  2503.2 2543.2
##
## Step: AIC=1449.15
## 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 + status_partner + status_single +
##   hs_livein + hs_rent_council + fi_2 + fi_4 + self_emp_part_yes +
##   gender_f
##
##               Df Deviance    AIC
## - self_emp_part_yes      1  1410.2 1448.2
## - status_single          1  1410.5 1448.5
## - fi_2                   1  1410.6 1448.6
## - hs_rent_council        1  1410.8 1448.8
## - fi_4                   1  1410.9 1448.9
## <none>                   1409.2 1449.2
## - status_partner         1  1412.0 1450.0
## - gender_f               1  1412.3 1450.3
## - hs_livein              1  1412.8 1450.8
## - Balance.Transfer        1  1418.7 1456.7
## - Investment.in.Mutual.Fund 1  1419.2 1457.2
## - Medical.Insurance       1  1439.4 1477.4
## - Home.Loan               1  1459.7 1497.7
## - Average.A.C.Balance     1  1469.0 1507.0
## - Term.Deposit            1  1489.0 1527.0
## - Life.Insurance          1  1669.7 1707.7
## - Personal.Loan           1  1685.8 1723.8
## - Investment.Tax.Saving.Bond 1  1689.7 1727.7
## - Average.Credit.Card.Transaction 1  1743.6 1781.6
## - Online.Purchase.Amount  1  2503.3 2541.3
##
## Step: AIC=1448.19
## 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 + status_partner + status_single +
##   hs_livein + hs_rent_council + fi_2 + fi_4 + gender_f
##
##               Df Deviance    AIC
## - status_single          1  1411.6 1447.6
## - fi_2                   1  1411.7 1447.7
## - fi_4                   1  1411.8 1447.8
## - hs_rent_council        1  1411.9 1447.9
## <none>                   1410.2 1448.2
## - gender_f               1  1412.8 1448.8
## - status_partner         1  1413.7 1449.7
## - hs_livein              1  1413.8 1449.8
## - Balance.Transfer        1  1419.8 1455.8
## - Investment.in.Mutual.Fund 1  1420.5 1456.5

```

```

## - Medical.Insurance          1  1440.3 1476.3
## - Home.Loan                  1  1460.6 1496.6
## - Average.A.C.Balance        1  1470.1 1506.1
## - Term.Deposit               1  1489.8 1525.8
## - Life.Insurance             1  1673.2 1709.2
## - Personal.Loan              1  1686.0 1722.0
## - Investment.Tax.Saving.Bond  1  1689.8 1725.8
## - Average.Credit.Card.Transaction 1  1744.4 1780.4
## - Online.Purchase.Amount      1  2503.3 2539.3
##
## Step:  AIC=1447.63
## 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 + status_partner + hs_livein +
##      hs_rent_council + fi_2 + fi_4 + gender_f
##
##                                     Df Deviance    AIC
## - hs_rent_council                 1  1413.3 1447.3
## - fi_2                            1  1413.3 1447.3
## - fi_4                            1  1413.4 1447.4
## <none>                             1411.6 1447.6
## - status_partner                  1  1413.7 1447.7
## - gender_f                        1  1414.5 1448.5
## - hs_livein                      1  1414.9 1448.9
## - Balance.Transfer                1  1421.2 1455.2
## - Investment.in.Mutual.Fund       1  1421.8 1455.8
## - Medical.Insurance               1  1441.6 1475.6
## - Home.Loan                      1  1462.2 1496.2
## - Average.A.C.Balance             1  1471.2 1505.2
## - Term.Deposit                   1  1491.2 1525.2
## - Life.Insurance                 1  1673.6 1707.6
## - Personal.Loan                  1  1687.4 1721.4
## - Investment.Tax.Saving.Bond      1  1690.6 1724.6
## - Average.Credit.Card.Transaction 1  1745.6 1779.6
## - Online.Purchase.Amount          1  2503.5 2537.5
##
## Step:  AIC=1447.29
## 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 + status_partner + hs_livein +
##      fi_2 + fi_4 + gender_f
##
##                                     Df Deviance    AIC
## - fi_2                            1  1415.1 1447.1
## - fi_4                            1  1415.2 1447.2
## <none>                             1413.3 1447.3
## - status_partner                  1  1415.5 1447.5
## - gender_f                        1  1416.4 1448.4
## - hs_livein                      1  1416.5 1448.5
## - Investment.in.Mutual.Fund       1  1423.2 1455.2
## - Balance.Transfer                1  1423.2 1455.2
## - Medical.Insurance               1  1443.6 1475.6

```

```

## - Home.Loan          1  1464.8 1496.8
## - Average.A.C.Balance 1  1472.6 1504.6
## - Term.Deposit       1  1492.9 1524.9
## - Life.Insurance     1  1675.8 1707.8
## - Personal.Loan      1  1688.5 1720.5
## - Investment.Tax.Saving.Bond 1  1693.2 1725.2
## - Average.Credit.Card.Transaction 1  1748.7 1780.7
## - Online.Purchase.Amount 1  2505.9 2537.9
##
## Step: AIC=1447.13
## 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 + status_partner + hs_livein +
##   fi_4 + gender_f
##
##               Df Deviance    AIC
## - fi_4          1  1416.0 1446.0
## <none>           1415.1 1447.1
## - status_partner 1  1417.5 1447.5
## - gender_f       1  1418.0 1448.0
## - hs_livein      1  1418.5 1448.5
## - Investment.in.Mutual.Fund 1  1424.9 1454.9
## - Balance.Transfer 1  1425.0 1455.0
## - Medical.Insurance 1  1445.4 1475.4
## - Home.Loan      1  1465.9 1495.9
## - Average.A.C.Balance 1  1474.7 1504.7
## - Term.Deposit   1  1494.2 1524.2
## - Life.Insurance 1  1676.2 1706.2
## - Personal.Loan  1  1689.2 1719.2
## - Investment.Tax.Saving.Bond 1  1695.5 1725.5
## - Average.Credit.Card.Transaction 1  1749.0 1779.0
## - Online.Purchase.Amount 1  2506.0 2536.0
##
## Step: AIC=1446.04
## 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 + status_partner + hs_livein +
##   gender_f
##
##               Df Deviance    AIC
## <none>           1416.0 1446.0
## - status_partner 1  1418.7 1446.7
## - gender_f       1  1419.0 1447.0
## - hs_livein      1  1419.3 1447.3
## - Investment.in.Mutual.Fund 1  1425.5 1453.5
## - Balance.Transfer 1  1425.8 1453.8
## - Medical.Insurance 1  1446.1 1474.1
## - Home.Loan      1  1466.7 1494.7
## - Average.A.C.Balance 1  1476.1 1504.1
## - Term.Deposit   1  1495.4 1523.4
## - Life.Insurance 1  1677.8 1705.8
## - Personal.Loan  1  1690.9 1718.9

```



```
## - Investment.Tax.Saving.Bond      1   1695.7 1723.7
## - Average.Credit.Card.Transaction 1   1749.6 1777.6
## - Online.Purchase.Amount          1   2507.0 2535.0
```

If you look at `summary(fit1)`, you'll find there are still some variable with high p-values. This is because dropping variables based on AIC scores has different inbuilt threshold, 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 ~ 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 + hs_livein +
##   gender_f
```

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

```
fit1=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 + status_single + fi_2 +
  fi_4 + self_emp_yes + gender_f , data=rg_train, family = "binomial")
summary(fit1)
```

```
##
## 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 +
##   status_single + fi_2 + fi_4 + self_emp_yes + gender_f, family = "binomial",
##   data = rg_train)
##
## Deviance Residuals:
##   Min       1Q   Median       3Q      Max
## -3.6579  -0.2361  -0.1521  -0.0753   4.1351
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.9219172   0.1982488 -19.783 < 2e-16 ***
## Average.Credit.Card.Transaction  0.0219430   0.0012550  17.485 < 2e-16 ***
## Balance.Transfer    -0.0037240   0.0012304  -3.027  0.00247 **
## Term.Deposit       -0.0164441   0.0021498  -7.649 2.02e-14 ***
## Life.Insurance      0.0148983   0.0009988  14.916 < 2e-16 ***
## Medical.Insurance   -0.0149173   0.0028626  -5.211 1.88e-07 ***
## Average.A.C.Balance -0.0157106   0.0022430  -7.004 2.48e-12 ***
## Personal.Loan       -0.0261163   0.0024248 -10.771 < 2e-16 ***
## Investment.in.Mutual.Fund -0.0044039   0.0014359  -3.067  0.00216 **
## Investment.Tax.Saving.Bond  0.0842711   0.0052988  15.904 < 2e-16 ***
## Home.Loan         -0.0675148   0.0105952  -6.372 1.86e-10 ***
```

```
## Online.Purchase.Amount      0.0507796  0.0024788  20.485 < 2e-16 ***
## status_single               -0.0418179  0.2373672  -0.176  0.86016
## fi_2                        0.2584439  0.1688212   1.531  0.12580
## fi_4                        0.2753048  0.1818719   1.514  0.13009
## self_emp_yes                0.1866080  0.2577350   0.724  0.46905
## gender_f                    -0.2746746  0.1634270  -1.681  0.09282 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3316.7 on 4923 degrees of freedom
## Residual deviance: 1418.6 on 4907 degrees of freedom
## AIC: 1452.6
##
## Number of Fisher Scoring iterations: 8
```

```
fit1=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_yes +gender_f , data=rg_train, family = "binomial")
summary(fit1)
```

```
##
## 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_yes + gender_f, family = "binomial",
##      data = rg_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.6574  -0.2359  -0.1522  -0.0753   4.1367
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.927644   0.195679 -20.072 < 2e-16 ***
## Average.Credit.Card.Transaction  0.021949   0.001255  17.494 < 2e-16 ***
## Balance.Transfer  -0.003724   0.001230  -3.028  0.00246 **
## Term.Deposit     -0.016445   0.002150  -7.650 2.01e-14 ***
## Life.Insurance    0.014906   0.000998  14.936 < 2e-16 ***
## Medical.Insurance -0.014927   0.002862  -5.215 1.83e-07 ***
## Average.A.C.Balance -0.015708   0.002243  -7.004 2.49e-12 ***
## Personal.Loan     -0.026103   0.002423 -10.771 < 2e-16 ***
## Investment.in.Mutual.Fund -0.004411   0.001435  -3.075  0.00210 **
## Investment.Tax.Saving.Bond  0.084254   0.005296  15.908 < 2e-16 ***
## Home.Loan        -0.067498   0.010596  -6.370 1.89e-10 ***
## Online.Purchase.Amount  0.050787   0.002479  20.486 < 2e-16 ***
## fi_2              0.257888   0.168799   1.528  0.12657
## fi_4              0.275640   0.181832   1.516  0.12954
## self_emp_yes      0.187719   0.257616   0.729  0.46620
```

```

## gender_f                -0.273755    0.163329  -1.676  0.09372 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3316.7  on 4923  degrees of freedom
## Residual deviance: 1418.6  on 4908  degrees of freedom
## AIC: 1450.6
##
## Number of Fisher Scoring iterations: 8

fit1=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 +gender_f , data=rg_train, family = "binomial")
summary(fit1)

##
## 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 + gender_f, family = "binomial", data = rg_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.6586  -0.2364  -0.1525  -0.0754   4.1396
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.9048255    0.1928202 -20.251 < 2e-16 ***
## Average.Credit.Card.Transaction  0.0219566    0.0012543  17.506 < 2e-16 ***
## Balance.Transfer    -0.0037029    0.0012295   -3.012  0.00260 **
## Term.Deposit       -0.0164637    0.0021474   -7.667 1.76e-14 ***
## Life.Insurance      0.0149439    0.0009972  14.985 < 2e-16 ***
## Medical.Insurance  -0.0149162    0.0028576   -5.220 1.79e-07 ***
## Average.A.C.Balance -0.0157583    0.0022435   -7.024 2.16e-12 ***
## Personal.Loan      -0.0260659    0.0024172  -10.783 < 2e-16 ***
## Investment.in.Mutual.Fund -0.0044567    0.0014317   -3.113  0.00185 **
## Investment.Tax.Saving.Bond  0.0842198    0.0052942  15.908 < 2e-16 ***
## Home.Loan         -0.0673455    0.0105892   -6.360 2.02e-10 ***
## Online.Purchase.Amount  0.0508638    0.0024802  20.508 < 2e-16 ***
## fi_2              0.2526508    0.1686192    1.498  0.13404
## fi_4              0.2716811    0.1816813    1.495  0.13482
## gender_f          -0.2848737    0.1626101   -1.752  0.07979 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3316.7  on 4923  degrees of freedom

```

```
## Residual deviance: 1419.2 on 4909 degrees of freedom
## AIC: 1449.2
##
## Number of Fisher Scoring iterations: 8
```

```
fit1=glm(Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +
  Term.Deposit + Life.Insurance + Medical.Insurance + Average.A.C.Balance +
  Personal.Loan + Investment.Tax.Saving.Bond +
  Home.Loan + Online.Purchase.Amount +
  gender_f , data=rg_train, family = "binomial")
summary(fit1)
```

```
##
## Call:
## glm(formula = Revenue.Grid ~ Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##     Home.Loan + Online.Purchase.Amount + gender_f, family = "binomial",
##     data = rg_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.5327  -0.2397  -0.1566  -0.0764   3.9062
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.765935   0.1737934 -21.667 < 2e-16 ***
## Average.Credit.Card.Transaction  0.0216549   0.0012486  17.343 < 2e-16 ***
## Balance.Transfer    -0.0045577   0.0012146  -3.753 0.000175 ***
## Term.Deposit       -0.0164977   0.0021435  -7.697 1.40e-14 ***
## Life.Insurance      0.0144039   0.0009815  14.676 < 2e-16 ***
## Medical.Insurance   -0.0149638   0.0028568  -5.238 1.62e-07 ***
## Average.A.C.Balance -0.0168700   0.0022408  -7.529 5.13e-14 ***
## Personal.Loan       -0.0261618   0.0023919 -10.938 < 2e-16 ***
## Investment.Tax.Saving.Bond    0.0818530   0.0052189  15.684 < 2e-16 ***
## Home.Loan          -0.0710940   0.0106376  -6.683 2.34e-11 ***
## Online.Purchase.Amount    0.0500121   0.0024363  20.528 < 2e-16 ***
## gender_f           -0.2986161   0.1620054  -1.843 0.065293 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 3316.7 on 4923 degrees of freedom
## Residual deviance: 1432.7 on 4912 degrees of freedom
## AIC: 1456.7
##
## Number of Fisher Scoring iterations: 8
```

Next we'll build the model on val data to find out variables which are consistently entering our model. We are skipping the step to find highly correlated variables for avoiding redundancy in the material [ You should include that step in your process]. We'll be dropping same set of variables from val data as we did from train for high VIF values.

```

rg_fit_val=rg_val %>%
  select(-REF_NO,-Investment.in.Commodity,-Investment.in.Derivative,
         -Investment.in.Equity,-gender_m,-hs_own,-fi_3,-Portfolio.Balance)
fit=glm(Revenue.Grid~.,family = "binomial",data=rg_fit_val)
fit=step(fit)

## Start:  AIC=683.46
## 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 +
##      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
## - children      1  619.46 681.46
## - op_1           1  619.47 681.47
## - Investment.in.Mutual.Fund  1  619.47 681.47
## - fi_2           1  619.48 681.48
## - hs_livein      1  619.49 681.49
## - fi_4           1  619.49 681.49
## - hs_rent_private  1  619.51 681.51
## - occ_HW         1  619.61 681.61
## - age           1  619.82 681.82
## - hs_rent_council  1  619.93 681.93
## - status_div     1  620.04 682.04
## - year_last_moved  1  620.11 682.11
## - fi_1           1  620.15 682.15
## - status_partner  1  620.34 682.34
## - Average.A.C.Balance  1  620.45 682.45
## - occ_Retired    1  621.27 683.27
## - self_emp_part_yes  1  621.35 683.35
## - status_single  1  621.41 683.41
## <none>           1  619.46 683.46
## - occ_BM_prof    1  621.82 683.82
## - op_2           1  621.90 683.90
## - self_emp_yes    1  621.98 683.98
## - gender_f       1  622.70 684.70
## - Medical.Insurance  1  626.71 688.71
## - Home.Loan      1  627.44 689.44
## - Balance.Transfer  1  635.14 697.14
## - Term.Deposit   1  663.81 725.81
## - Life.Insurance  1  678.19 740.19
## - Investment.Tax.Saving.Bond  1  749.75 811.75
## - Personal.Loan   1  752.35 814.35
## - Average.Credit.Card.Transaction  1  776.58 838.58
## - Online.Purchase.Amount  1  991.36 1053.36
##
## Step:  AIC=681.46
## 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 +
##      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
## - op_1              1   619.47  679.47
## - Investment.in.Mutual.Fund      1   619.47  679.47
## - fi_2              1   619.48  679.48
## - hs_livein          1   619.49  679.49
## - fi_4              1   619.49  679.49
## - hs_rent_private      1   619.51  679.51
## - occ_HW              1   619.61  679.61
## - age                1   619.85  679.85
## - hs_rent_council      1   619.94  679.94
## - status_div          1   620.04  680.04
## - year_last_moved      1   620.11  680.11
## - fi_1              1   620.15  680.15
## - status_partner       1   620.34  680.34
## - Average.A.C.Balance      1   620.45  680.45
## - occ_Retired          1   621.27  681.27
## - self_emp_part_yes      1   621.35  681.35
## - status_single        1   621.45  681.45
## <none>              619.46  681.46
## - occ_BM_prof          1   621.82  681.82
## - op_2              1   621.94  681.94
## - self_emp_yes          1   621.98  681.98
## - gender_f            1   622.71  682.71
## - Medical.Insurance      1   626.72  686.72
## - Home.Loan            1   627.44  687.44
## - Balance.Transfer       1   635.17  695.17
## - Term.Deposit          1   663.85  723.85
## - Life.Insurance        1   678.24  738.24
## - Investment.Tax.Saving.Bond      1   749.76  809.76
## - Personal.Loan         1   753.35  813.35
## - Average.Credit.Card.Transaction 1   776.60  836.60
## - Online.Purchase.Amount      1   991.37 1051.37
##
## Step:  AIC=679.47
## 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 +
##      age + status_div + status_partner + status_single + occ_BM_prof +
##      occ_Retired + occ_HW + 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
## - Investment.in.Mutual.Fund      1   619.48  677.48
## - fi_2              1   619.48  677.48
## - hs_livein          1   619.50  677.50

```

```

## - fi_4 1 619.50 677.50
## - hs_rent_private 1 619.52 677.52
## - occ_HW 1 619.62 677.62
## - age 1 619.89 677.89
## - hs_rent_council 1 619.94 677.94
## - status_div 1 620.06 678.06
## - year_last_moved 1 620.12 678.12
## - fi_1 1 620.18 678.18
## - status_partner 1 620.34 678.34
## - Average.A.C.Balance 1 620.46 678.46
## - self_emp_part_yes 1 621.36 679.36
## - occ_Retired 1 621.44 679.44
## - status_single 1 621.46 679.46
## <none> 619.47 679.47
## - occ_BM_prof 1 621.83 679.83
## - self_emp_yes 1 621.98 679.98
## - op_2 1 622.07 680.07
## - gender_f 1 622.71 680.71
## - Medical.Insurance 1 626.74 684.74
## - Home.Loan 1 627.46 685.46
## - Balance.Transfer 1 635.17 693.17
## - Term.Deposit 1 663.95 721.95
## - Life.Insurance 1 678.33 736.33
## - Investment.Tax.Saving.Bond 1 749.87 807.87
## - Personal.Loan 1 753.37 811.37
## - Average.Credit.Card.Transaction 1 777.45 835.45
## - Online.Purchase.Amount 1 991.37 1049.37
##
## Step: AIC=677.48
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
## Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
## Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
## Home.Loan + Online.Purchase.Amount + age + status_div + status_partner +
## status_single + occ_BM_prof + occ_Retired + occ_HW + 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
## - fi_2 1 619.49 675.49
## - hs_livein 1 619.50 675.50
## - fi_4 1 619.51 675.51
## - hs_rent_private 1 619.52 675.52
## - occ_HW 1 619.63 675.63
## - age 1 619.89 675.89
## - hs_rent_council 1 619.95 675.95
## - status_div 1 620.06 676.06
## - year_last_moved 1 620.13 676.13
## - fi_1 1 620.20 676.20
## - status_partner 1 620.35 676.35
## - Average.A.C.Balance 1 620.56 676.56
## - self_emp_part_yes 1 621.37 677.37
## - occ_Retired 1 621.45 677.45
## - status_single 1 621.46 677.46
## <none> 619.48 677.48

```

```

## - occ_BM_prof          1  621.85  677.85
## - self_emp_yes         1  621.99  677.99
## - op_2                 1  622.09  678.09
## - gender_f             1  622.72  678.72
## - Medical.Insurance    1  626.76  682.76
## - Home.Loan            1  627.57  683.57
## - Balance.Transfer     1  636.28  692.28
## - Term.Deposit         1  663.95  719.95
## - Life.Insurance       1  681.02  737.02
## - Investment.Tax.Saving.Bond 1  751.00  807.00
## - Personal.Loan        1  753.51  809.51
## - Average.Credit.Card.Transaction 1  778.92  834.92
## - Online.Purchase.Amount 1  992.25 1048.25
##
## Step:  AIC=675.49
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##      Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##      Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##      Home.Loan + Online.Purchase.Amount + age + status_div + status_partner +
##      status_single + occ_BM_prof + occ_Retired + occ_HW + op_2 +
##      hs_livein + hs_rent_private + hs_rent_council + fi_1 + fi_4 +
##      self_emp_yes + self_emp_part_yes + gender_f
##
##                                     Df Deviance    AIC
## - fi_4                            1  619.51  673.51
## - hs_livein                       1  619.52  673.52
## - hs_rent_private                  1  619.54  673.54
## - occ_HW                          1  619.64  673.64
## - age                             1  619.91  673.91
## - hs_rent_council                  1  619.96  673.96
## - status_div                      1  620.07  674.07
## - year_last_moved                  1  620.15  674.15
## - fi_1                            1  620.21  674.21
## - status_partner                   1  620.35  674.35
## - Average.A.C.Balance              1  620.59  674.59
## - self_emp_part_yes                1  621.40  675.40
## - occ_Retired                     1  621.46  675.46
## - status_single                    1  621.46  675.46
## <none>                             619.49  675.49
## - occ_BM_prof                     1  621.85  675.85
## - self_emp_yes                     1  621.99  675.99
## - op_2                           1  622.09  676.09
## - gender_f                        1  622.75  676.75
## - Medical.Insurance               1  626.79  680.79
## - Home.Loan                       1  627.58  681.58
## - Balance.Transfer                 1  636.28  690.28
## - Term.Deposit                    1  664.08  718.08
## - Life.Insurance                  1  681.07  735.07
## - Investment.Tax.Saving.Bond      1  751.04  805.04
## - Personal.Loan                   1  754.37  808.37
## - Average.Credit.Card.Transaction 1  778.93  832.93
## - Online.Purchase.Amount           1  992.26 1046.26
##
## Step:  AIC=673.51

```



```

## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##   Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##   Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##   Home.Loan + Online.Purchase.Amount + age + status_div + status_partner +
##   status_single + occ_BM_prof + occ_Retired + occ_HW + op_2 +
##   hs_livein + hs_rent_private + hs_rent_council + fi_1 + self_emp_yes +
##   self_emp_part_yes + gender_f
##
##
##           Df Deviance      AIC
## - hs_livein          1   619.54  671.54
## - hs_rent_private     1   619.56  671.56
## - occ_HW              1   619.67  671.67
## - age                 1   619.93  671.93
## - hs_rent_council     1   619.99  671.99
## - status_div          1   620.10  672.10
## - year_last_moved     1   620.18  672.18
## - fi_1                1   620.21  672.21
## - status_partner      1   620.37  672.37
## - Average.A.C.Balance 1   620.62  672.62
## - self_emp_part_yes   1   621.43  673.43
## - occ_Retired         1   621.47  673.47
## - status_single       1   621.47  673.47
## <none>                1   619.51  673.51
## - occ_BM_prof         1   621.85  673.85
## - self_emp_yes        1   622.02  674.02
## - op_2                1   622.10  674.10
## - gender_f            1   622.76  674.76
## - Medical.Insurance   1   626.81  678.81
## - Home.Loan           1   627.60  679.60
## - Balance.Transfer     1   636.28  688.28
## - Term.Deposit        1   664.09  716.09
## - Life.Insurance      1   681.21  733.21
## - Investment.Tax.Saving.Bond 1   751.41  803.41
## - Personal.Loan       1   754.44  806.44
## - Average.Credit.Card.Transaction 1   778.95  830.95
## - Online.Purchase.Amount 1   992.30 1044.30
##
## Step:  AIC=671.54
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##   Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##   Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##   Home.Loan + Online.Purchase.Amount + age + status_div + status_partner +
##   status_single + occ_BM_prof + occ_Retired + occ_HW + op_2 +
##   hs_rent_private + hs_rent_council + fi_1 + self_emp_yes +
##   self_emp_part_yes + gender_f
##
##
##           Df Deviance      AIC
## - hs_rent_private     1   619.59  669.59
## - occ_HW              1   619.70  669.70
## - age                 1   620.00  670.00
## - hs_rent_council     1   620.01  670.01
## - status_div          1   620.12  670.12
## - year_last_moved     1   620.19  670.19
## - fi_1                1   620.23  670.23

```

```

## - status_partner          1  620.41  670.41
## - Average.A.C.Balance    1  620.65  670.65
## - self_emp_part_yes      1  621.44  671.44
## - status_single          1  621.48  671.48
## - occ_Retired            1  621.50  671.50
## <none>                   619.54  671.54
## - occ_BM_prof            1  621.91  671.91
## - self_emp_yes           1  622.03  672.03
## - op_2                   1  622.11  672.11
## - gender_f               1  622.80  672.80
## - Medical.Insurance      1  626.81  676.81
## - Home.Loan              1  627.61  677.61
## - Balance.Transfer       1  636.44  686.44
## - Term.Deposit           1  664.31  714.31
## - Life.Insurance         1  681.31  731.31
## - Investment.Tax.Saving.Bond 1  751.80  801.80
## - Personal.Loan          1  755.36  805.36
## - Average.Credit.Card.Transaction 1  779.19  829.19
## - Online.Purchase.Amount  1  994.03 1044.03
##
## Step: AIC=669.59
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##   Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##   Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##   Home.Loan + Online.Purchase.Amount + age + status_div + status_partner +
##   status_single + occ_BM_prof + occ_Retired + occ_HW + op_2 +
##   hs_rent_council + fi_1 + self_emp_yes + self_emp_part_yes +
##   gender_f
##
##                                Df Deviance    AIC
## - occ_HW                      1  619.75  667.75
## - hs_rent_council             1  620.03  668.03
## - age                         1  620.05  668.05
## - status_div                  1  620.15  668.15
## - fi_1                       1  620.24  668.24
## - year_last_moved             1  620.25  668.25
## - status_partner              1  620.43  668.43
## - Average.A.C.Balance         1  620.70  668.70
## - self_emp_part_yes           1  621.47  669.47
## - status_single               1  621.51  669.51
## - occ_Retired                 1  621.56  669.56
## <none>                       619.59  669.59
## - occ_BM_prof                1  621.95  669.95
## - self_emp_yes                1  622.09  670.09
## - op_2                       1  622.13  670.13
## - gender_f                   1  622.83  670.83
## - Medical.Insurance           1  626.95  674.95
## - Home.Loan                   1  627.67  675.67
## - Balance.Transfer             1  636.49  684.49
## - Term.Deposit                1  664.50  712.50
## - Life.Insurance              1  681.55  729.55
## - Investment.Tax.Saving.Bond  1  752.38  800.38
## - Personal.Loan               1  755.37  803.37
## - Average.Credit.Card.Transaction 1  780.77  828.77

```

```

## - Online.Purchase.Amount          1    994.52 1042.52
##
## Step: AIC=667.75
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##     Home.Loan + Online.Purchase.Amount + age + status_div + status_partner +
##     status_single + occ_BM_prof + occ_Retired + op_2 + hs_rent_council +
##     fi_1 + self_emp_yes + self_emp_part_yes + gender_f
##
##                                     Df Deviance    AIC
## - age                             1    620.24  666.24
## - hs_rent_council                 1    620.24  666.24
## - status_div                      1    620.30  666.30
## - year_last_moved                 1    620.46  666.46
## - fi_1                           1    620.47  666.47
## - status_partner                  1    620.63  666.63
## - Average.A.C.Balance             1    620.80  666.80
## - occ_Retired                     1    621.56  667.56
## - self_emp_part_yes               1    621.57  667.57
## - status_single                   1    621.67  667.67
## <none>                            1    619.75  667.75
## - occ_BM_prof                     1    621.96  667.96
## - self_emp_yes                    1    622.21  668.21
## - op_2                           1    622.35  668.35
## - gender_f                       1    623.27  669.27
## - Medical.Insurance               1    627.05  673.05
## - Home.Loan                      1    627.91  673.91
## - Balance.Transfer                 1    636.55  682.55
## - Term.Deposit                    1    664.73  710.73
## - Life.Insurance                  1    681.55  727.55
## - Investment.Tax.Saving.Bond      1    752.40  798.40
## - Personal.Loan                   1    755.66  801.66
## - Average.Credit.Card.Transaction 1    780.79  826.79
## - Online.Purchase.Amount          1    994.76 1040.76
##
## Step: AIC=666.24
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##     Home.Loan + Online.Purchase.Amount + status_div + status_partner +
##     status_single + occ_BM_prof + occ_Retired + op_2 + hs_rent_council +
##     fi_1 + self_emp_yes + self_emp_part_yes + gender_f
##
##                                     Df Deviance    AIC
## - hs_rent_council                 1    620.67  664.67
## - status_div                      1    620.69  664.69
## - status_partner                  1    620.95  664.95
## - fi_1                           1    621.09  665.09
## - Average.A.C.Balance             1    621.25  665.25
## - year_last_moved                 1    621.77  665.77
## - status_single                   1    621.78  665.78
## - self_emp_part_yes               1    622.23  666.23
## <none>                            1    620.24  666.24

```

```

## - occ_BM_prof          1  622.40  666.40
## - self_emp_yes         1  622.80  666.80
## - op_2                 1  623.08  667.08
## - gender_f             1  623.45  667.45
## - occ_Retired          1  624.06  668.06
## - Medical.Insurance    1  627.44  671.44
## - Home.Loan            1  628.29  672.29
## - Balance.Transfer     1  637.08  681.08
## - Term.Deposit         1  665.70  709.70
## - Life.Insurance       1  681.91  725.91
## - Investment.Tax.Saving.Bond 1  753.04  797.04
## - Personal.Loan        1  755.71  799.71
## - Average.Credit.Card.Transaction 1  781.50  825.50
## - Online.Purchase.Amount 1  995.27 1039.27
##
## Step: AIC=664.67
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##     Home.Loan + Online.Purchase.Amount + status_div + status_partner +
##     status_single + occ_BM_prof + occ_Retired + op_2 + fi_1 +
##     self_emp_yes + self_emp_part_yes + gender_f
##
##                                     Df Deviance    AIC
## - status_div                     1  621.07  663.07
## - fi_1                           1  621.30  663.30
## - status_partner                  1  621.39  663.39
## - Average.A.C.Balance             1  621.66  663.66
## - status_single                   1  622.11  664.11
## - year_last_moved                 1  622.30  664.30
## - self_emp_part_yes               1  622.63  664.63
## <none>                           620.67  664.67
## - occ_BM_prof                    1  622.99  664.99
## - self_emp_yes                   1  623.28  665.28
## - op_2                           1  623.62  665.62
## - gender_f                       1  623.81  665.81
## - occ_Retired                    1  624.42  666.42
## - Medical.Insurance              1  627.81  669.81
## - Home.Loan                      1  628.87  670.87
## - Balance.Transfer                1  637.08  679.08
## - Term.Deposit                   1  666.69  708.69
## - Life.Insurance                 1  682.17  724.17
## - Investment.Tax.Saving.Bond      1  753.92  795.92
## - Personal.Loan                  1  756.77  798.77
## - Average.Credit.Card.Transaction 1  782.26  824.26
## - Online.Purchase.Amount          1  995.61 1037.61
##
## Step: AIC=663.07
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##     Home.Loan + Online.Purchase.Amount + status_partner + status_single +
##     occ_BM_prof + occ_Retired + op_2 + fi_1 + self_emp_yes +
##     self_emp_part_yes + gender_f

```

```

##
##
##      Df Deviance      AIC
## - status_partner      1  621.39  661.39
## - fi_1                  1  621.62  661.62
## - Average.A.C.Balance  1  622.02  662.02
## - status_single        1  622.15  662.15
## - year_last_moved      1  622.66  662.66
## - self_emp_part_yes    1  622.99  662.99
## <none>                  621.07  663.07
## - occ_BM_prof          1  623.37  663.37
## - self_emp_yes         1  623.57  663.57
## - gender_f             1  624.05  664.05
## - op_2                 1  624.05  664.05
## - occ_Retired          1  624.59  664.59
## - Medical.Insurance    1  628.29  668.29
## - Home.Loan            1  629.17  669.17
## - Balance.Transfer      1  637.83  677.83
## - Term.Deposit         1  667.60  707.60
## - Life.Insurance       1  682.55  722.55
## - Investment.Tax.Saving.Bond 1  754.36  794.36
## - Personal.Loan        1  757.19  797.19
## - Average.Credit.Card.Transaction 1  783.49  823.49
## - Online.Purchase.Amount 1  996.04 1036.04
##
## Step:  AIC=661.39
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##      Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##      Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##      Home.Loan + Online.Purchase.Amount + status_single + occ_BM_prof +
##      occ_Retired + op_2 + fi_1 + self_emp_yes + self_emp_part_yes +
##      gender_f
##
##      Df Deviance      AIC
## - fi_1                  1  621.78  659.78
## - status_single        1  622.19  660.19
## - Average.A.C.Balance  1  622.35  660.35
## - year_last_moved      1  623.08  661.08
## - self_emp_part_yes    1  623.21  661.21
## <none>                  621.39  661.39
## - occ_BM_prof          1  623.69  661.69
## - self_emp_yes         1  623.89  661.89
## - op_2                 1  624.25  662.25
## - gender_f             1  624.35  662.35
## - occ_Retired          1  624.83  662.83
## - Medical.Insurance    1  628.58  666.58
## - Home.Loan            1  629.30  667.30
## - Balance.Transfer      1  638.10  676.10
## - Term.Deposit         1  667.96  705.96
## - Life.Insurance       1  682.74  720.74
## - Investment.Tax.Saving.Bond 1  754.36  792.36
## - Personal.Loan        1  757.28  795.28
## - Average.Credit.Card.Transaction 1  783.50  821.50
## - Online.Purchase.Amount 1  996.04 1034.04
##

```

```

## Step: AIC=659.78
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##     Home.Loan + Online.Purchase.Amount + status_single + occ_BM_prof +
##     occ_Retired + op_2 + self_emp_yes + self_emp_part_yes + gender_f
##
##
##           Df Deviance    AIC
## - status_single      1  622.62  658.62
## - Average.A.C.Balance 1  622.72  658.72
## - year_last_moved     1  623.59  659.59
## - self_emp_part_yes   1  623.64  659.64
## <none>                621.78  659.78
## - occ_BM_prof        1  623.95  659.95
## - self_emp_yes       1  624.24  660.24
## - op_2               1  624.71  660.71
## - gender_f           1  624.85  660.85
## - occ_Retired        1  625.76  661.76
## - Medical.Insurance  1  629.07  665.07
## - Home.Loan          1  629.54  665.54
## - Balance.Transfer    1  638.49  674.49
## - Term.Deposit        1  668.53  704.53
## - Life.Insurance      1  682.89  718.89
## - Investment.Tax.Saving.Bond 1  754.96  790.96
## - Personal.Loan       1  757.28  793.28
## - Average.Credit.Card.Transaction 1  784.94  820.94
## - Online.Purchase.Amount 1  996.71 1032.71
##
## Step: AIC=658.62
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Average.A.C.Balance + Personal.Loan + Investment.Tax.Saving.Bond +
##     Home.Loan + Online.Purchase.Amount + occ_BM_prof + occ_Retired +
##     op_2 + self_emp_yes + self_emp_part_yes + gender_f
##
##
##           Df Deviance    AIC
## - Average.A.C.Balance 1  623.56  657.56
## - year_last_moved     1  624.11  658.11
## - self_emp_part_yes   1  624.61  658.61
## <none>                622.62  658.62
## - occ_BM_prof        1  625.05  659.05
## - self_emp_yes       1  625.05  659.05
## - gender_f           1  625.60  659.60
## - op_2               1  625.64  659.64
## - occ_Retired        1  626.47  660.47
## - Medical.Insurance  1  629.84  663.84
## - Home.Loan          1  630.61  664.61
## - Balance.Transfer    1  639.31  673.31
## - Term.Deposit        1  668.93  702.93
## - Life.Insurance      1  684.23  718.23
## - Investment.Tax.Saving.Bond 1  755.56  789.56
## - Personal.Loan       1  757.41  791.41
## - Average.Credit.Card.Transaction 1  785.19  819.19
## - Online.Purchase.Amount 1  998.07 1032.07

```

```

##
## Step: AIC=657.56
## Revenue.Grid ~ year_last_moved + Average.Credit.Card.Transaction +
##     Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##     Personal.Loan + Investment.Tax.Saving.Bond + Home.Loan +
##     Online.Purchase.Amount + occ_BM_prof + occ_Retired + op_2 +
##     self_emp_yes + self_emp_part_yes + gender_f
##
##
##           Df Deviance    AIC
## - year_last_moved      1  625.05  657.05
## - self_emp_part_yes      1  625.48  657.48
## <none>                  623.56  657.56
## - occ_BM_prof          1  625.88  657.88
## - self_emp_yes         1  626.02  658.02
## - op_2                 1  626.46  658.46
## - gender_f             1  626.90  658.90
## - occ_Retired          1  627.53  659.53
## - Medical.Insurance    1  631.30  663.30
## - Home.Loan            1  633.51  665.51
## - Balance.Transfer     1  641.73  673.73
## - Term.Deposit         1  671.90  703.90
## - Life.Insurance       1  687.59  719.59
## - Investment.Tax.Saving.Bond 1  755.57  787.57
## - Personal.Loan        1  757.41  789.41
## - Average.Credit.Card.Transaction 1  787.00  819.00
## - Online.Purchase.Amount 1  998.37 1030.37
##
## Step: AIC=657.05
## Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +
##     Term.Deposit + Life.Insurance + Medical.Insurance + Personal.Loan +
##     Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##     occ_BM_prof + occ_Retired + op_2 + self_emp_yes + self_emp_part_yes +
##     gender_f
##
##
##           Df Deviance    AIC
## - self_emp_part_yes      1  627.02  657.02
## <none>                  625.05  657.05
## - occ_BM_prof          1  627.09  657.09
## - self_emp_yes         1  627.61  657.61
## - op_2                 1  628.07  658.07
## - gender_f             1  628.50  658.50
## - occ_Retired          1  630.32  660.32
## - Medical.Insurance    1  632.26  662.26
## - Home.Loan            1  634.80  664.80
## - Balance.Transfer     1  643.09  673.09
## - Term.Deposit         1  674.02  704.02
## - Life.Insurance       1  688.66  718.66
## - Investment.Tax.Saving.Bond 1  757.05  787.05
## - Personal.Loan        1  758.22  788.22
## - Average.Credit.Card.Transaction 1  787.65  817.65
## - Online.Purchase.Amount 1 1001.07 1031.07
##
## Step: AIC=657.02
## Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +

```

```
##      Term.Deposit + Life.Insurance + Medical.Insurance + Personal.Loan +
##      Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
##      occ_BM_prof + occ_Retired + op_2 + self_emp_yes + gender_f
##
##                                Df Deviance      AIC
## <none>                        627.02  657.02
## - occ_BM_prof                  1   629.30  657.30
## - self_emp_yes                 1   629.35  657.35
## - op_2                        1   629.66  657.66
## - gender_f                    1   630.13  658.13
## - occ_Retired                 1   633.29  661.29
## - Medical.Insurance           1   634.72  662.72
## - Home.Loan                   1   636.67  664.67
## - Balance.Transfer            1   645.29  673.29
## - Term.Deposit                1   675.15  703.15
## - Life.Insurance              1   690.93  718.93
## - Investment.Tax.Saving.Bond  1   757.09  785.09
## - Personal.Loan               1   759.09  787.09
## - Average.Credit.Card.Transaction 1   790.74  818.74
## - Online.Purchase.Amount      1  1001.56 1029.56
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = Revenue.Grid ~ Average.Credit.Card.Transaction +
##      Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##      Personal.Loan + Investment.Tax.Saving.Bond + Home.Loan +
##      Online.Purchase.Amount + occ_BM_prof + occ_Retired + op_2 +
##      self_emp_yes + gender_f, family = "binomial", data = rg_fit_val)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.9295  -0.2424  -0.1538  -0.0815   3.8435
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.527562   0.312227  -11.298 < 2e-16 ***
## Average.Credit.Card.Transaction  0.022237   0.001866   11.914 < 2e-16 ***
## Balance.Transfer  -0.007371   0.001868   -3.946 7.96e-05 ***
## Term.Deposit     -0.020718   0.003667   -5.650 1.61e-08 ***
## Life.Insurance    0.010431   0.001320    7.900 2.79e-15 ***
## Medical.Insurance -0.011181   0.004240   -2.637 0.00836 **
## Personal.Loan     -0.030801   0.004135   -7.449 9.40e-14 ***
## Investment.Tax.Saving.Bond  0.084911   0.007891   10.760 < 2e-16 ***
## Home.Loan        -0.042243   0.014945   -2.827 0.00470 **
## Online.Purchase.Amount  0.051350   0.004038   12.715 < 2e-16 ***
## occ_BM_prof      -0.383069   0.256070   -1.496 0.13467
## occ_Retired      -0.753534   0.310337   -2.428 0.01518 *
## op_2              0.692203   0.409556    1.690 0.09100 .
## self_emp_yes     -0.645753   0.436161   -1.481 0.13873
## gender_f         -0.452514   0.254217   -1.780 0.07507 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1402.57 on 2110 degrees of freedom
## Residual deviance: 627.02 on 2096 degrees of freedom
## AIC: 657.02
##
## Number of Fisher Scoring iterations: 8

formula(fit)

## Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +
## Term.Deposit + Life.Insurance + Medical.Insurance + Personal.Loan +
## Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
## occ_BM_prof + occ_Retired + op_2 + self_emp_yes + gender_f

fit2=glm(Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +
Term.Deposit + Life.Insurance + Medical.Insurance + Personal.Loan +
Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
occ_Retired + op_2 + self_emp_yes + gender_f,family="binomial",data=rg_fit_val)
summary(fit2)

##
## Call:
## glm(formula = Revenue.Grid ~ Average.Credit.Card.Transaction +
## Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
## Personal.Loan + Investment.Tax.Saving.Bond + Home.Loan +
## Online.Purchase.Amount + occ_Retired + op_2 + self_emp_yes +
## gender_f, family = "binomial", data = rg_fit_val)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -3.9900 -0.2392 -0.1568 -0.0808 3.8258
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.729535 0.285711 -13.054 < 2e-16 ***
## Average.Credit.Card.Transaction 0.022302 0.001860 11.989 < 2e-16 ***
## Balance.Transfer -0.007237 0.001865 -3.880 0.000104 ***
## Term.Deposit -0.020527 0.003634 -5.649 1.61e-08 ***
## Life.Insurance 0.010385 0.001327 7.828 4.94e-15 ***
## Medical.Insurance -0.011328 0.004206 -2.694 0.007070 **
## Personal.Loan -0.030883 0.004169 -7.408 1.28e-13 ***
## Investment.Tax.Saving.Bond 0.084166 0.007851 10.720 < 2e-16 ***
## Home.Loan -0.042941 0.015063 -2.851 0.004362 **
## Online.Purchase.Amount 0.051225 0.004033 12.702 < 2e-16 ***
## occ_Retired -0.590180 0.290518 -2.031 0.042207 *
## op_2 0.700953 0.407720 1.719 0.085577 .
## self_emp_yes -0.626106 0.439702 -1.424 0.154466
## gender_f -0.366329 0.247165 -1.482 0.138307
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1402.6 on 2110 degrees of freedom
## Residual deviance: 629.3 on 2097 degrees of freedom
## AIC: 657.3
##
## Number of Fisher Scoring iterations: 8

fit2=glm(Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +
  Term.Deposit + Life.Insurance + Medical.Insurance + Personal.Loan +
  Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
  occ_Retired + op_2 + gender_f,family="binomial",data=rg_fit_val)
summary(fit2)

##
## Call:
## glm(formula = Revenue.Grid ~ Average.Credit.Card.Transaction +
## Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
## Personal.Loan + Investment.Tax.Saving.Bond + Home.Loan +
## Online.Purchase.Amount + occ_Retired + op_2 + gender_f, family = "binomial",
## data = rg_fit_val)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -3.9432 -0.2375 -0.1588 -0.0836 3.8107
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.799028 0.282571 -13.445 < 2e-16 ***
## Average.Credit.Card.Transaction 0.022130 0.001851 11.953 < 2e-16 ***
## Balance.Transfer -0.007237 0.001880 -3.849 0.000118 ***
## Term.Deposit -0.019961 0.003575 -5.583 2.37e-08 ***
## Life.Insurance 0.010370 0.001323 7.837 4.61e-15 ***
## Medical.Insurance -0.011331 0.004210 -2.691 0.007116 **
## Personal.Loan -0.030489 0.004120 -7.401 1.36e-13 ***
## Investment.Tax.Saving.Bond 0.083339 0.007791 10.697 < 2e-16 ***
## Home.Loan -0.045670 0.015180 -3.009 0.002624 **
## Online.Purchase.Amount 0.050739 0.003994 12.704 < 2e-16 ***
## occ_Retired -0.528656 0.285564 -1.851 0.064131 .
## op_2 0.710620 0.407586 1.743 0.081249 .
## gender_f -0.316976 0.244613 -1.296 0.195035
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1402.57 on 2110 degrees of freedom
## Residual deviance: 631.45 on 2098 degrees of freedom
## AIC: 657.45
##
## Number of Fisher Scoring iterations: 8
```

```
fit2=glm(Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +
  Term.Deposit + Life.Insurance + Medical.Insurance + Personal.Loan +
  Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +
  occ_Retired + op_2,family="binomial",data=rg_fit_val)
summary(fit2)
```

```
##
## Call:
## glm(formula = Revenue.Grid ~ Average.Credit.Card.Transaction +
##      Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +
##      Personal.Loan + Investment.Tax.Saving.Bond + Home.Loan +
##      Online.Purchase.Amount + occ_Retired + op_2, family = "binomial",
##      data = rg_fit_val)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.9654  -0.2407  -0.1590  -0.0832   3.7774
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.049134   0.213586 -18.958 < 2e-16 ***
## Average.Credit.Card.Transaction  0.022115   0.001846  11.982 < 2e-16 ***
## Balance.Transfer    -0.007290   0.001882  -3.873 0.000108 ***
## Term.Deposit       -0.019924   0.003576  -5.572 2.52e-08 ***
## Life.Insurance      0.010419   0.001319   7.899 2.81e-15 ***
## Medical.Insurance   -0.011233   0.004206  -2.671 0.007569 **
## Personal.Loan       -0.030808   0.004130  -7.459 8.70e-14 ***
## Investment.Tax.Saving.Bond  0.083057   0.007792  10.659 < 2e-16 ***
## Home.Loan          -0.046111   0.015282  -3.017 0.002550 **
## Online.Purchase.Amount  0.051084   0.004008  12.745 < 2e-16 ***
## occ_Retired        -0.471132   0.279867  -1.683 0.092295 .
## op_2               0.808005   0.400726   2.016 0.043763 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1402.6  on 2110  degrees of freedom
## Residual deviance:  633.1  on 2099  degrees of freedom
## AIC: 657.1
##
## Number of Fisher Scoring iterations: 8
```

We'll select variables which are common to models built for train and validation i.e. `fit1` and `fit2`. and use those variables finally to build a model on `train` again.

```
formula(fit1)
```

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

```
formula(fit2)
```

```
## Revenue.Grid ~ Average.Credit.Card.Transaction + Balance.Transfer +  
##      Term.Deposit + Life.Insurance + Medical.Insurance + Personal.Loan +  
##      Investment.Tax.Saving.Bond + Home.Loan + Online.Purchase.Amount +  
##      occ_Retired + op_2
```

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

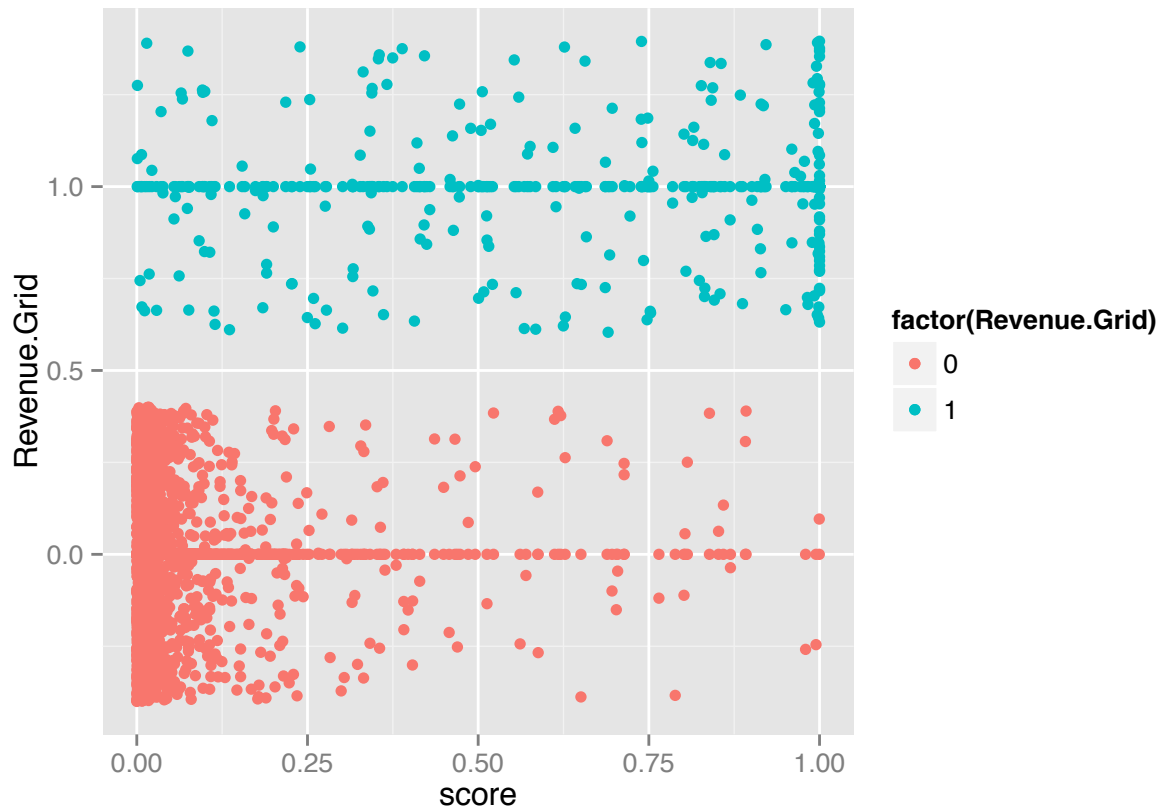
```
##  
## Call:  
## glm(formula = Revenue.Grid ~ Average.Credit.Card.Transaction +  
##      Balance.Transfer + Term.Deposit + Life.Insurance + Medical.Insurance +  
##      Personal.Loan + Investment.Tax.Saving.Bond + Home.Loan +  
##      Online.Purchase.Amount, family = "binomial", data = rg_train)  
##  
## Deviance Residuals:  
##      Min       1Q   Median       3Q      Max   
## -4.2071  -0.2496  -0.1730  -0.0856   3.5619   
##  
## Coefficients:  
##              Estimate Std. Error z value Pr(>|z|)      
## (Intercept)    -3.9736758   0.1274277  -31.184 < 2e-16 ***  
## Average.Credit.Card.Transaction  0.0202933   0.0012007   16.901 < 2e-16 ***  
## Balance.Transfer    -0.0056296   0.0012093   -4.655 3.23e-06 ***  
## Term.Deposit      -0.0170637   0.0020850   -8.184 2.74e-16 ***  
## Life.Insurance     0.0114624   0.0008671   13.219 < 2e-16 ***  
## Medical.Insurance  -0.0149815   0.0027633   -5.422 5.91e-08 ***  
## Personal.Loan     -0.0246517   0.0021925  -11.244 < 2e-16 ***  
## Investment.Tax.Saving.Bond    0.0749716   0.0050373   14.883 < 2e-16 ***  
## Home.Loan        -0.0886641   0.0104108   -8.517 < 2e-16 ***  
## Online.Purchase.Amount    0.0466041   0.0023093   20.181 < 2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## (Dispersion parameter for binomial family taken to be 1)  
##  
##      Null deviance: 3316.7  on 4923  degrees of freedom  
## Residual deviance: 1504.4  on 4914  degrees of freedom  
## AIC: 1524.4  
##  
## 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. we'll get those probability scores for validation data using predict function.

```
rg_val$score=predict(fit2,newdata=rg_val,type = "response")
```

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

```
library(ggplot2)
ggplot(rg_val,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 way 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_val$score>cutoff)
TP=sum(predicted==1 & rg_val$Revenue.Grid==1)
FP=sum(predicted==1 & rg_val$Revenue.Grid==0)
FN=sum(predicted==0 & rg_val$Revenue.Grid==1)
TN=sum(predicted==0 & rg_val$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_val$score>cutoff)

  TP=sum(predicted==1 & rg_val$Revenue.Grid==1)
  FP=sum(predicted==1 & rg_val$Revenue.Grid==0)
  FN=sum(predicted==0 & rg_val$Revenue.Grid==1)
  TN=sum(predicted==0 & rg_val$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 correctly 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 positives 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 percentage 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

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

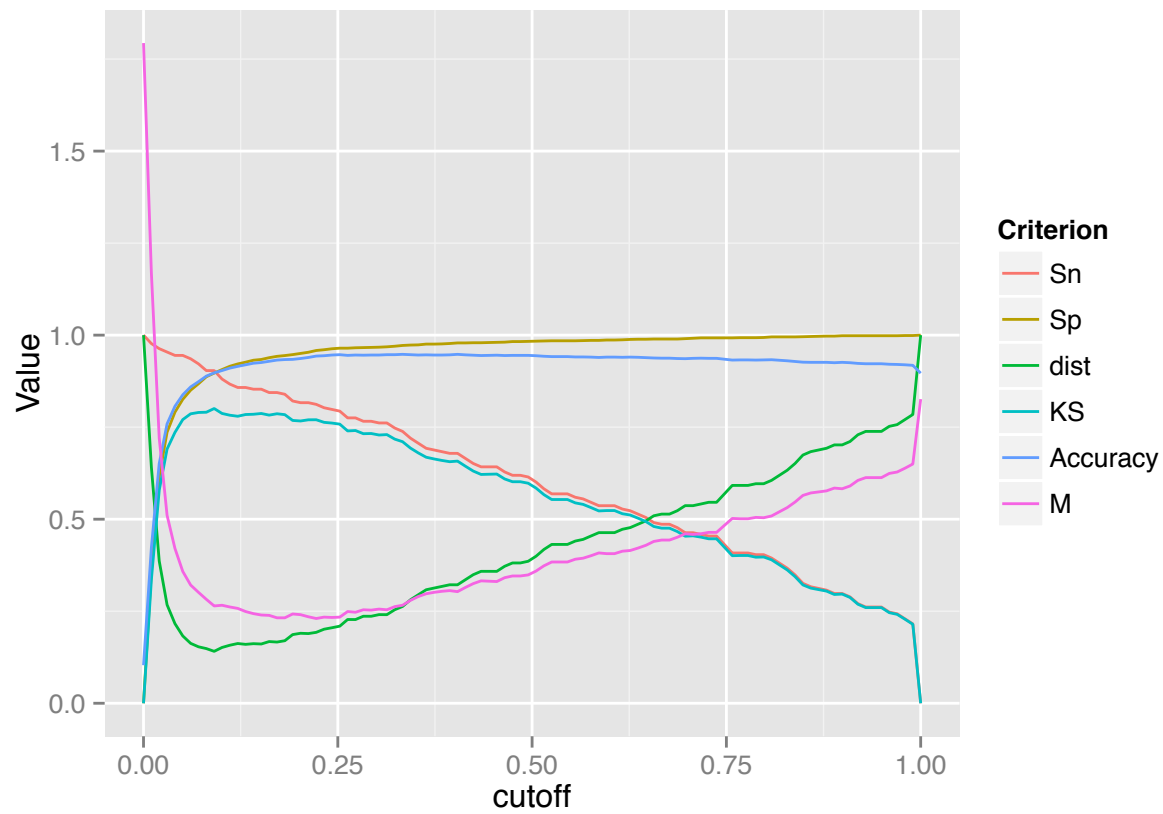
We will calculate all those performance metrics and get cutoff according to them. Lets get to work.

```
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)

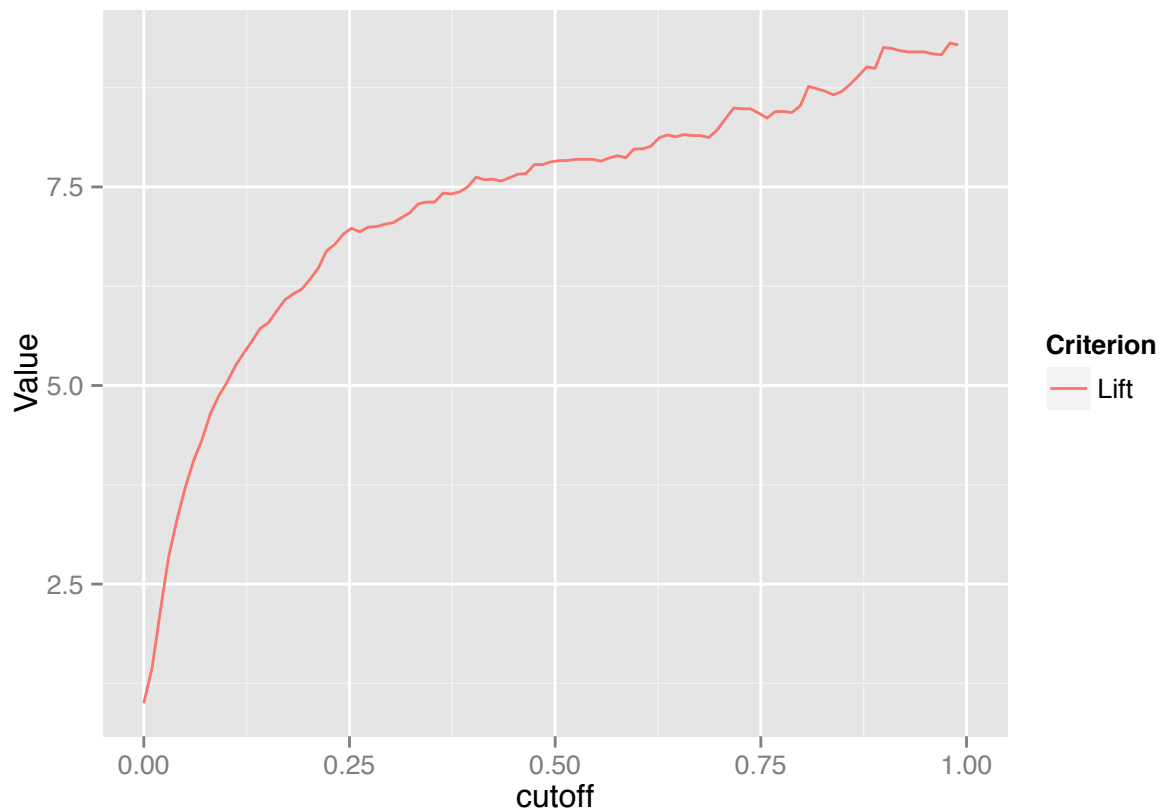
ggplot(filter(cutoff_viz,Criterion!="Lift"),aes(x=cutoff,y=Value,color=Criterion))+
  geom_line()
```



*#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] 0.09090909
```

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

```
##
##      0      1
## 0 2384  288
## 1   31  313
```

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

```
## [1] 0.09090909
```

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

```
##  
##      0      1  
## 0 2384  288  
## 1   31  313
```

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] 0.3333333
```

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

```
##  
##      0      1  
## 0 2605   67  
## 1  101  243
```

```
# 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] 0.2222222
```

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

```
##  
##      0      1  
## 0 2561  111  
## 1   74  270
```

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.