IDS 572: Assignment 2

Target Marketing - Fundraising

Archan Patel - 661105271 - apate381@uic.edu

Jasbir Singh - 651837003 - jasingh3@uic.edu

Jeetyog Rangnekar - 656052696 - jrangn2@uic.edu

Assignment Question 1

- The dataset has total of 479 input variables and 1 output variable TARGET_B. All of the input variables cannot be used for the model. *(We are not considering TARGET_D as output variable since we are not concerned with amount of donation for this problem statement)*.

Step 1

- So the first task performed is to understand the variables using the data dictionary.
    - After going through the variables, many of them can be dropped because of different reasons. Some of them are listed below along with their reason.
    - Some of the variables can be dropped because of their irrelevance to the problem statement. Some of the examples: *(It is not a full list of the dropped variables, rather - a small subset of the all dropped variables for this criteria. This applies to all given sample examples in the remaining report.)*
        - RECSWEEP
        - COLLECT1
        - BIBLE
        - CDPLAY
        - STEREO
        - FISHER
        - CARDS
        - PLATES
        - RAMNT
        - MINRAMNT
        - MAXRAMNT
    - Some of the variables can be dropped because of redundant data. Some of the examples:
        - DOB - since we already have AGE variable.
        - RFA_2R - Since we already have RFA_2 variable containing same value
        - RFA_2F - Since we already have RFA_2 variable containing same value
        - RFA_2A - Since we already have RFA_2 variable containing same value
        - AC1, AC2,.. - Since we already have AGEC variables defining ages
        - RP1 - Since the count is already included in RP4
        - HC1, HC2,.. - Since the newer houses will have higher values/Rent
        - ETHC1, ETHC2,.. - Since Ethnicity information is already captured in ETH1, ETH2,...
    - Some of the variables can be dropped because of too many missing data (when missing data do not indicate anything). Some of the examples:

        - PVASTATE - Because the Data only contains ‘P’ & we have 15876 missing values, out of each we are not sure which one is ‘E’ and which one is really missing
        - CHILD03
        - CHILD07
        - LASTDATE
        - FISTDATE
- After removing the variables based on the above mentioned criteria, the total no. of input variables are reduced to 223 from 479.

Step 2

- The next step is to replace the missing values. Some of the blank values in the dataset are not actually missing data. Instead, they carry information. For example,
    - In MAILCODE - the blank value means the Address is Ok.
    - In MAJOR - the blank value means the donor is not a major donor.
    - So, this type of missing values need to replaced with some value - they can not be dropped. Otherwise, we may lose important information from dataset.
- First of all, the numeric attributes missing values are being replaced by the average values of the variable. Some of the examples:
    - AGE
    - INCOME
    - MBCRAFT, MBGARDEN,...
    - PUBOPP, PUBGARDN,...
- Since, missing values for nominal variables can’t be handled with average, a specific value has been assigned to missing data based on the variable definition. Some of the examples:
    - DATASRC - replaced by ‘3’ (maximum occurrence)
    - SOLP3, SOLIH - replaced by a code ‘10’
    - MAJOR - replaced by code ‘NX’ for not a major donor
    - GEOCODE - replaced by ‘0’ for no code instead of blank
    - MAILCODE - replaced by ‘A’ for good address
    - RFA - replaced by ‘XXXX’ code

Step 3

- After replacing the missing values as mentioned above, there are still some variables with very few missing data. Rather than replacing them with some value, we can remove the records with the missing value from the dataset.
    - For example, records with missing values in OSOURCE, GEOCODE etc. can be removed from the dataset since the missing values are around 2% of the total records for these variables.
- After removing the records with missing data for few of the variables (CLUSTER, DOMAIN, GEOCODE etc.), now the only variables remain with missing values are ADATE_3,...RDATE_3,...RFA_3,...

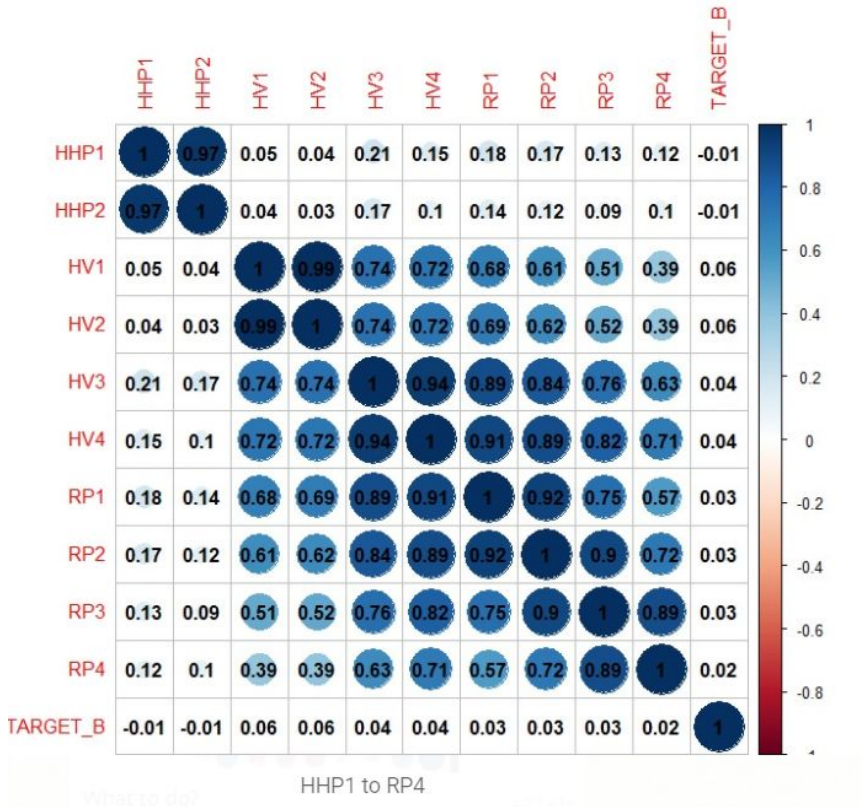
Step 4

- Next step would be to create new meaningful attributes based on the values from some of the existing attributes and then drop them.
- Since, we are not concerned with the actual date when the promotion was mailed, using the ADATE_3, ADATE_4,... variables, we can create a new variable which can tell us the total number of times the promotion was mailed across all promotion types mentioned in the dataset.
    - So, using all ADATE series variables, a new variable 'Mail_Sent_Count' is created
- In the similar fashion, a new variable 'Resp_Date' is created using all RDATE series variables which gives us the total number of times the donor has responded to the promotion across all promotion types.
- RFA variables indicate the rating of the donor for different promotion type. Since, they are nominal variables, the same amount of new variables were created which can define a score for the donor. The higher the score, the higher R,F,A rating for the donor.
    - For example, is a donor's RFA_3 code is 'S4G', then the score for the RFA_2 promotion for that donor would be 6+4+7=18.
    - In the similar fashion, new rating variables for all RFA codes are created.
- Since, we are not concerned with the individual promotion type, a new variable is created which can give us the average score across all promotion type - from the above newly mentioned rating variables.
    - So this new variable will total the rating of each promotion type for a donor and then give the average score of RFA rating of the donor.
- Once we have these new variables, we can drop the existing variables which were used to create them.
    - We have 3 new variables now which gives us meaningful information - Mail_Sent_Count, Resp_Count, RFA_Rate_Average
    - So, we can now drop ADATE,.., RDATE..,RFA,.. & newly created RFA_Rating variables as well.
- So, we now have the dataset with zero missing value for any of the variable and much less number of variables from the initial count.

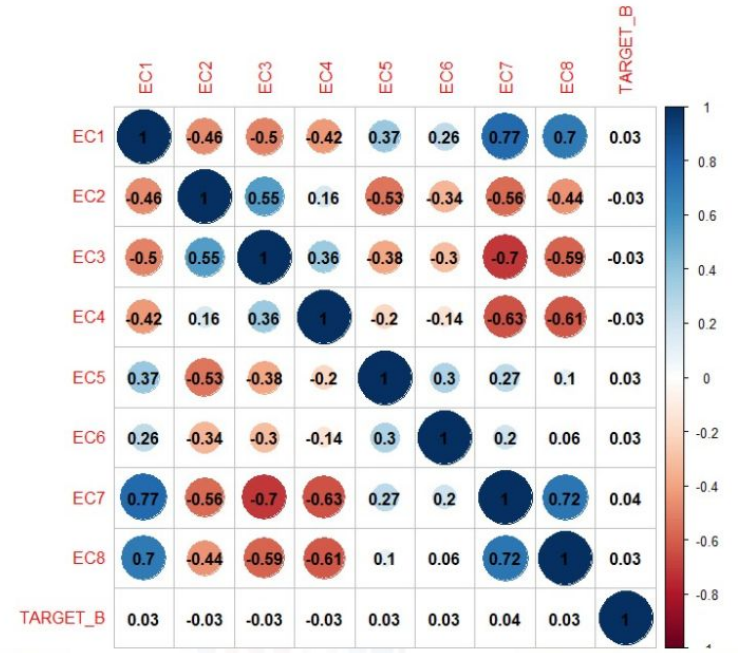
Step 5

- In order to reduce the data further, the outcome dataset till now can be run on different models like Decision Tree - weight by Information Gain, Random Forest etc. This is to determine the important variables which are considered as main variables in the decision tree splits. Some examples:
    - CLUSTER
    - TCODE
    - INCOME

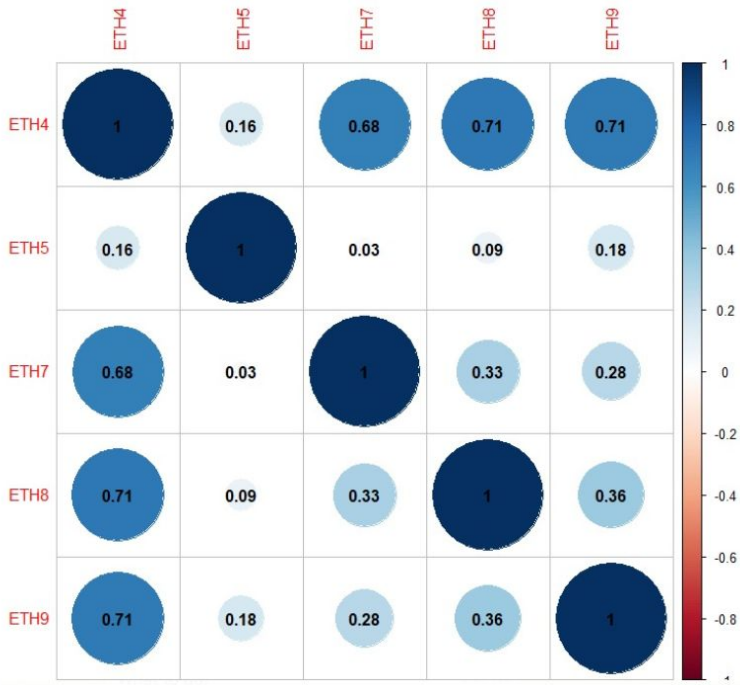
    - Resp_Count
    - STATE
- Apart from that, for some of the variables the correlation among the variables and with the target variable can also be considered.
    - The variables who have higher correlation among them (value > 0.9) - we can keep only one and remove the remaining variables since they all provide similar kind of information.
        - In the case, where median value and average value provide the same information, we can keep the variable giving the median value since, median is immune to outliers (if there are any)
    - We can keep the variables who have significant correlation (approx. between 0.5 - 0.9) among them as well as with the target variable as they can be considered significant.
    - The variables who have lower correlation among them as well as with the target variable can be dropped as we can consider them not significant ones.
    - R commands[1] *(listed at the end of the report)* were used to find the correlation among variables and below are some of the correlation plot for the variables.

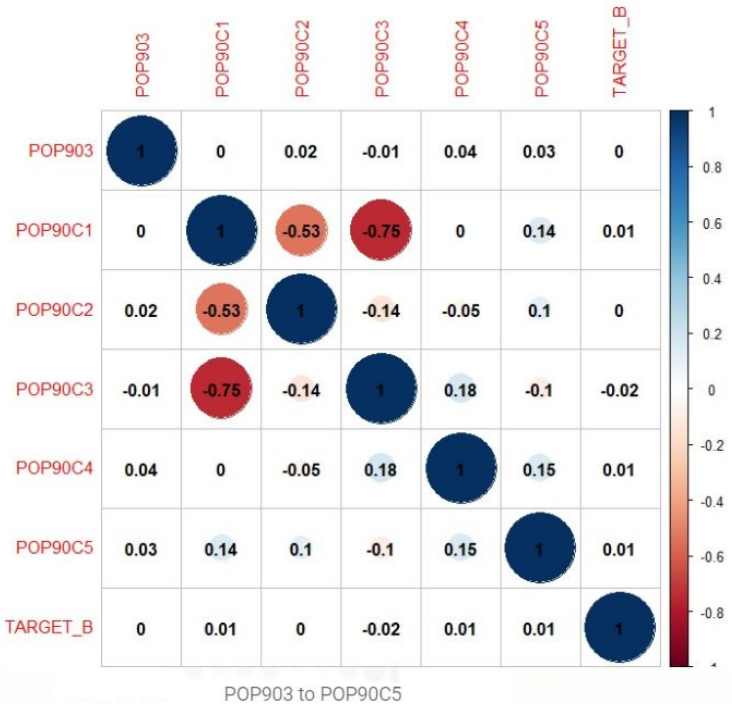


HHP1 to RP4


EC1
EC2
EC3
EC4
EC5
EC6
EC7
EC8
TARGET_B
EC1 1 -0.46 -0.5 -0.42 0.37 0.26 0.77 0.7 0.03
EC2 -0.46 1 0.55 0.16 -0.53 -0.34 -0.56 -0.44 -0.03
EC3 -0.5 0.55 1 0.36 -0.38 -0.3 -0.7 -0.59 -0.03
EC4 -0.42 0.16 0.36 1 -0.2 -0.14 -0.63 -0.61 -0.03
EC5 0.37 -0.53 -0.38 -0.2 1 0.3 0.27 0.1 0.03
EC6 0.26 -0.34 -0.3 -0.14 0.3 1 0.2 0.06 0.03
EC7 0.77 -0.56 -0.7 -0.63 0.27 0.2 1 0.72 0.04
EC8 0.7 -0.44 -0.59 -0.61 0.1 0.06 0.72 1 0.03
TARGET_B 0.03 -0.03 -0.03 -0.03 0.03 0.03 0.04 0.03 1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

EC1 to EC8

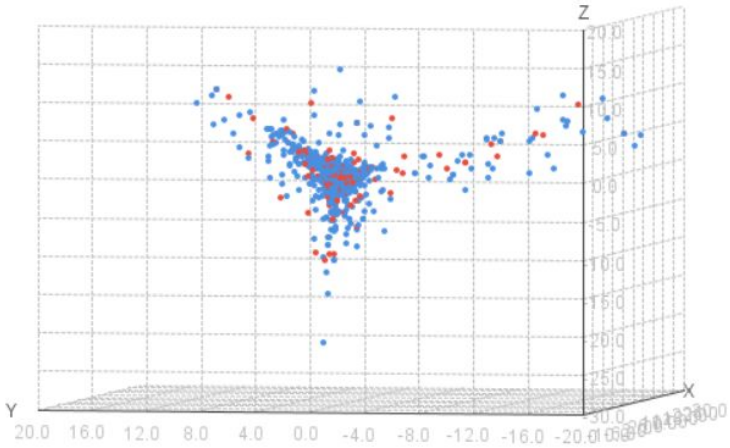

ETH4
ETH5
ETH7
ETH8
ETH9
ETH4 1 0.16 0.68 0.71 0.71
ETH5 0.16 1 0.03 0.09 0.18
ETH7 0.68 0.03 1 0.33 0.28
ETH8 0.71 0.09 0.33 1 0.36
ETH9 0.71 0.18 0.28 0.36 1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8



- Considering the different classifiers and correlation, the variables can be reduced further. Some of the examples:
    - HHP2
    - EIC1, EIC2,..
    - OCC1, OCC2,..
    - POP901, POP902,..
- After dropping these variables, we now have 64 input variables.

Step 6

- Next, we would further reduce the dimension of the variables using PCA. Rather than applying PCA for all similar group of variables, we can run it on the highly correlated variables. The correlation can be found from correlation plot explained above. Also, the values need to be normalized.
    - So, based on the correlation findings above, we can see that VC1, VC3, VIETVETS & WWIIVETS are highly correlated. So we perform PCA on them and find reduced PCA vectors PCA_Veterans1 & PCA_Veterans2 (after renaming them).
    - Similarly, we can run the PCA for AGE901, HHAGE2, HHAGE3 and find reduced PCA vectors PCA_AGE1 & PCA_AGE2.
    - Run PCA for EC1, EC7 & EC8 → PCA_EC1 & PCA_EC2
    - Run PCA for ETH4, ETH7, ETH8, ETH9 → PCA_ETH1, PCA_ETH2, PCA_ETH3
    - Here is the 3-D chart of the PCA_ETH vectors for example.



- Since we now have PCA vectors, we can remove the existing attributes which were used for PCA. Some of the examples:
    - AGE901
    - VC1
    - ETH4
- After removing these attributes, we now have 52 inputs variables on which we can apply different models.

Assignment Question 2

Below are the considered classifier models and their outcome.

- **Random Forest**
    - The higher number of trees in Random Forest are giving 0% class precision for class 1. The lower number of trees are giving negligible amount of class precision for class 1.

accuracy: 78.71%

	true 0	true 1	class precision
pred. 0	4719	1260	78.93%
pred. 1	18	5	21.74%
class recall	99.62%	0.40%	

- **Random Forest with Bagging & Cross-Validation**
    - If we consider Random Forest with Bagging and Cross-Validation, then also the model is giving 0% class precision

accuracy: 78.92%

	true 0	true 1	class precision
pred. 0	10658	2847	78.92%
pred. 1	0	0	0.00%
class recall	100.00%	0.00%	

- Hence, this model can also not be considered as a good model.

- **Logistic Regression**
  - Here are the different outcomes for the Ridge & Lasso Logistic Regression using different lambda values.

Lambda = 1.2E-0.6		Lambda = 1.5E-0.4		Lambda = 2.2E-0.8	
Ridge	Lasso	Ridge	Lasso	Ridge	Lasso
CTr = 39.46 CTe = 14.07	CTr = 5.69 CTe = 3.00	CTr = 16.54 CTe = 6.96	CTr = 5.16 CTe = 3.08	CTr = 40.04 CTe = 14.70	CTr = 10.01 CTe = 4.66

CTr = Class Precision of class 1 for Train Dataset
CTe = Class Precision of class 1 for Test Dataset

accuracy: 74.19%

	true 0	true 1	class precision
pred. 0	4267	1079	79.82%
pred. 1	470	186	28.35%
class recall	90.08%	14.70%	

accuracy: 78.46%

	true 0	true 1	class precision
pred. 0	4650	1206	79.41%
pred. 1	87	59	40.41%
class recall	98.16%	4.66%	

- **Bagging**
  - The better model for Bagging can be obtained from Bagging with Cross-Validation with number of folds = 10, Decision Tree split criteria = Information Gain with maximal depth=16 & iterations of Bagging = 10.

- Training Performance

accuracy: 87.63% +/- 0.23% (mikro: 87.63%)

	true 0	true 1	class precision
pred. 0	99246	9369	91.37%
pred. 1	7341	19098	72.23%
class recall	93.11%	67.09%	

- Testing Performance

accuracy: 69.11% +/- 0.66% (mikro: 69.11%)

	true 0	true 1	class precision
pred. 0	9789	2581	79.14%
pred. 1	2054	582	22.08%
class recall	82.66%	18.40%	

- **Boosting**
  - Boosting is giving ovetfit models with the trainign set - it is giving 100% accuracy on both the classes for any parameters.
  - And the class precision for class 1 on test dataset is around 20% for all.

accuracy: 68.76% +/- 0.65% (mikro: 68.76%)

	true 0	true 1	class precision
pred. 0	9671	2516	79.36%
pred. 1	2172	647	22.95%
class recall	81.66%	20.46%	

  - Because of overfitting, Boosting cannot be considered a good model.

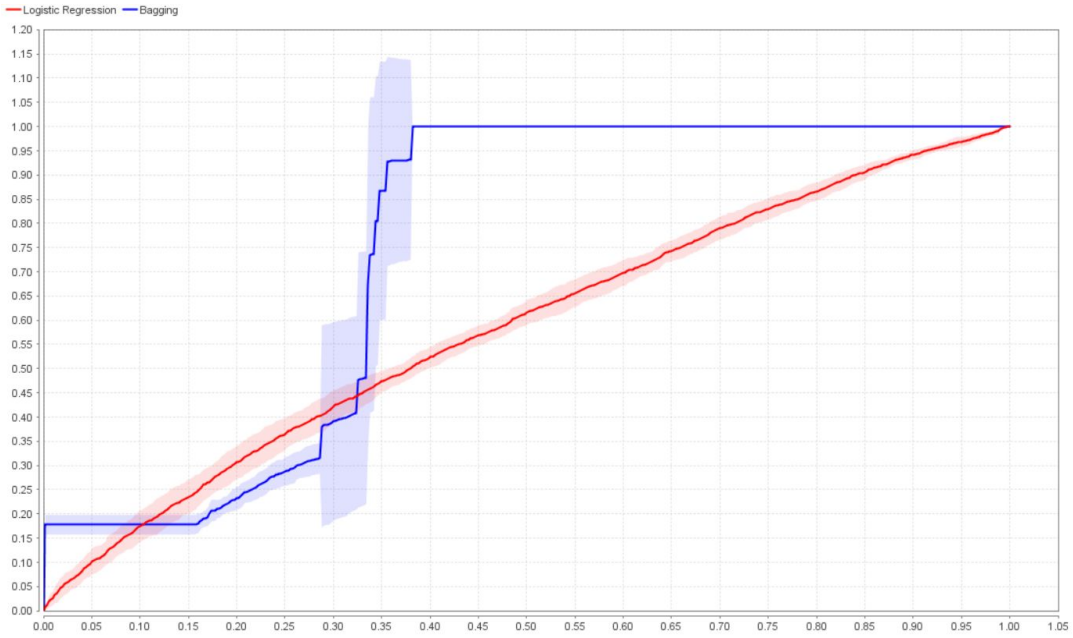
- **Decision Tree**
  - As explained in Boosting, the Decision tree with different split criteria is also providing 100% accuracy on both classes for training dataset.
  - Hence, it can also not be considered.

- **J48 Decision Tree**
  - Regardless of the parameters, all J48 decision tree are giving the 0% class precision for class 1. Which means the model is predicting class 0 for each output - which cannot be considered a good model at all.

- **CART Decision Tree**
  - As explained in J48, regardless of the parameters, all CART decision tree are giving 0% class precision for class 1. It cannot be considered a good model as

well.

- So, from examining all different models & considering the average donation $13 & cost $0.68, below is the calculation of cost considering comparatively two better model - Bagging & Logistic Regression.
    - Bagging:
        - Cost: (582+2054+2581)*0.68 = $3547.56
        - Donation: 582 * 13 = $7566
        - Profit: $7566 - $3547.56 = $4018.44
    - Logistic Regression
        - Cost: (186+1079+470) * 0.68 = $1179.8
        - Donation: 186 * 13 = $2418
        - Profit: $2418 - $1179.8 = $1238.2
- So, we can consider Bagging to be a good model based on the cost.
- Below is the ROC curve comparison for these 2 models.



- From the ROC curve also, we can say that the Bagging is a good model than Logistic Regression.

Assignment Question 3

- Original data consist of 13 millions observation and 480 variables , In such a large data set , we generally face issue of having a skewed target response variable . A sample is

usually a miniature of a population it came of and it should be representative w.r.t all variables measured. However, due to skewed distribution of target variable , we face the problem of non-response, where in a large number of non-responder dominate as compared to smaller of responder, so when we are partitioning the data we may find that the likely of non-responders will be higher in the training set , thereby inducing a bias & hence no reliable patterns can be drawn from the data for the lack of representative.

- In weighting adjustment the particular weight is assigned to each responder, which helps us in using the weighted the values rather than the actual values thereby reducing the skewness.
  On the original data set the actual response rate was 5.1% as against the response rate of 21% obtained by weighted sampling, so this shows, using the simple random sampling we may end up building a model which has only non-responder or very few responder.
  Also weighted sampling helps in dealing with a large number of variables where in it is important to capture information for each variable for successful response donation.

- Now, we can depend only on classification accuracy as a good performance metric for maximising net profit as it make use of a weighted sample.
  Apart from accuracy we also need to consider precision, recall & F-score, the precision gives us the fraction of correctly predicted positive labels of all the example & on the other hand recall gives us the fraction of all the positive examples that were picked up by the classifiers. As our goal is to build a model that capture as many real donors as possible, in order to maximize the net profit we will focus on the class recall & precision for the donor. A higher recall & a lower miss-rate is an indicator of a better model. Therefore, with all the information available, while selecting our best model we could make ROC curve to make a decision on predictive model.

R-commands

[1] Some of the R-commands to find the correlation plots.

- exp <- read.csv(file.choose(), header = T)
- corrplot(cor(exp[,c(43:50)]), method = "circle", addCoef.col = T)
- corrplot(cor(exp[,c(41:43,49,50)]), method = "circle", addCoef.col = T)
- corrplot(cor(exp[,c(28:32)]), method = "circle", addCoef.col = T)