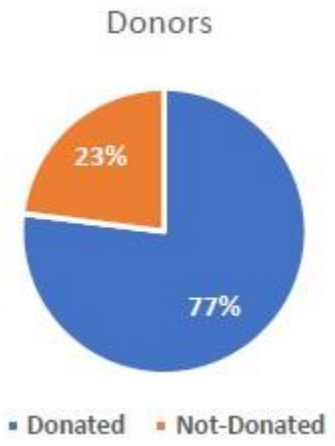
Target Marketing
PVA Fundraising

OBJECTIVE: To clean the data, conduct an exploratory analysis on which variables may be useful to predict donors, and then build and compare predictive models for effectively identifying people who will respond to a marketing mail campaign.

The pvaDataForModeling_Spr2019.csv dataset contains 480 attributes. On studying the Target Variable, TARGET−B Binary Indicator for Response 1 = Donor & 0 = Non-donor, we see the distribution below. 77% of the people in the dataset are Donors, while the remaining 23% are Non – Donors.



QUESTION 1 :

Data cleaning, exploration The dataset has many variables – some (many?) of them may not be useful for our purpose. Your first task is to explore the data, determine missing values and how you might handle these, which variables you think need not be considered, which should be transformed, creating new variables, etc. This is a major task – and can take significant time, much more than the modeling step that comes next. (You will find below two tables with subsets of variables that were found useful in earlier analysis). (a) Which attributes will you omit from the analyses and why? (b) How do you clean the data, handle missing values? (c) What new attributes/values do you derive? (d) Which variables will you consider for modeling (and why)? How do your findings relate with the variable subsets given in Tables 1 and 2. Explain how you approach data reduction, variable selection? What methods do you try and find useful? Summarize your findings.

Data cleaning and exploration:

a) Missing Values

Making a classification model with 480 attributes is a difficult process. Hence the initial step is to perform data investigation – deciding missing qualities, changing certain attributes and lastly perform Principal Component Analysis on certain factors. By running PCA, we can diminish the quantity of factors. Instinctively, we have also eliminated the factors that we felt don't contribute towards our target variable.

The variables have been transformed to the best of my understanding, the reasons for their transformation have been mentioned in the table below.

- **Generate Attributes:** This block generated new attributes that modified the values of some of our existing attributes

Some of the missing values carry information. Instead of removing them completely from the dataset and loose the chance for it to be passed to a decision tree or PCA I have provided these missing data with a certain value. This will help distinguish the missing data from the original data whilst keeping the essence of the data.

Original Attribute Name	**New Name**	**Attribute Transformation**	**Reasons**
MAJOR	Major_1	if(MAJOR == "X",1,0)	All blanks replaced with 0 to identify people who are not majors.
SOLIH	SOLIH_1	if(missing(SOLIH) , 24, SOLIH)	Uniquely identify people who can be contacted at any time.
RECINHSE	rechinse	if(RECINHSE == "X", "1","0")	

RECP3
 	
RECSWEEP
	Recp3
	
	Recsweep
		if(RECP3 == "X", "1", "0")
		
		if(RECSWEEP == "X", "1", "0")
			All blanks replaced with 0 to identify a No.

PEPSTRFL	Pepstrfl	if(PEPSTRFL=="X", "1", "0")	
HOMEOWNR	Homeowner	if(HOMEOWNR=="H", "1", "0")	
ADATE_3	96NK	if(ADATE_3 != 0, 1, 0)	Replacing blanks with 0 to identify if any impact on promotions.
ADATE_14	95NK	if(ADATE_14 != 0, 1, 0)	
ADATE_24	94NK	if(ADATE_24 != 0, 1, 0)	
MAGFAML		'?' TO 0	Missing values changed to 0 to identify mails where contact was not made
MAGFEM		'?' TO 0	
MAGMALE		'?' TO 0	
MBBOOKS		'?' TO 0	
MBCOLECT		'?' TO 0	
MBCRAFT		'?' TO 0	
MBGARDEN		'?' TO 0	
PUBCULIN		'?' TO 0	Missing values changed to 0 to identify
PUBDOITY		'?' TO 0	
PUBGARDEN		'?' TO 0	

PUBHEALTH		'?' TO 0	mails where contact was not made
PUBNEWFN		'?' TO 0	
PUBOPP		'?' TO 0	
PUBPHOTO		'?' TO 0	
AGE		'?' TO 0	Replacing missing values with 0.
INCOME		'?' TO 0	
TIMELAG		'?' TO 0	

WEALTH1		'?' TO -1	Replacing with -1 to distinguish the missing values from the original dataset
WEALTH2		'?' TO -1	
CLUSTER		'?' TO -1	Replacing with -1 to distinguish the missing values from the original dataset
CLUSTER2		'?' TO -1	
NUMCHLD 1,	Child_flag	if (NUMCHLD > 0, 0)	Replaced with 0 to identify people with no childen
DOMAIN	domainSES	'?' TO Z	Replacing with Z to distinguish the missing values from the original dataset
DOMAIN	Urbanicity	'?' TO Z	

b) Eliminating Variables based on redundant data

Post our initial attribute transformation we notice that many of the attributes cover the same type of data. Removing these redundant attributes will make our data set mush smaller. We even dropped attributes that would not add any value to our target variable.

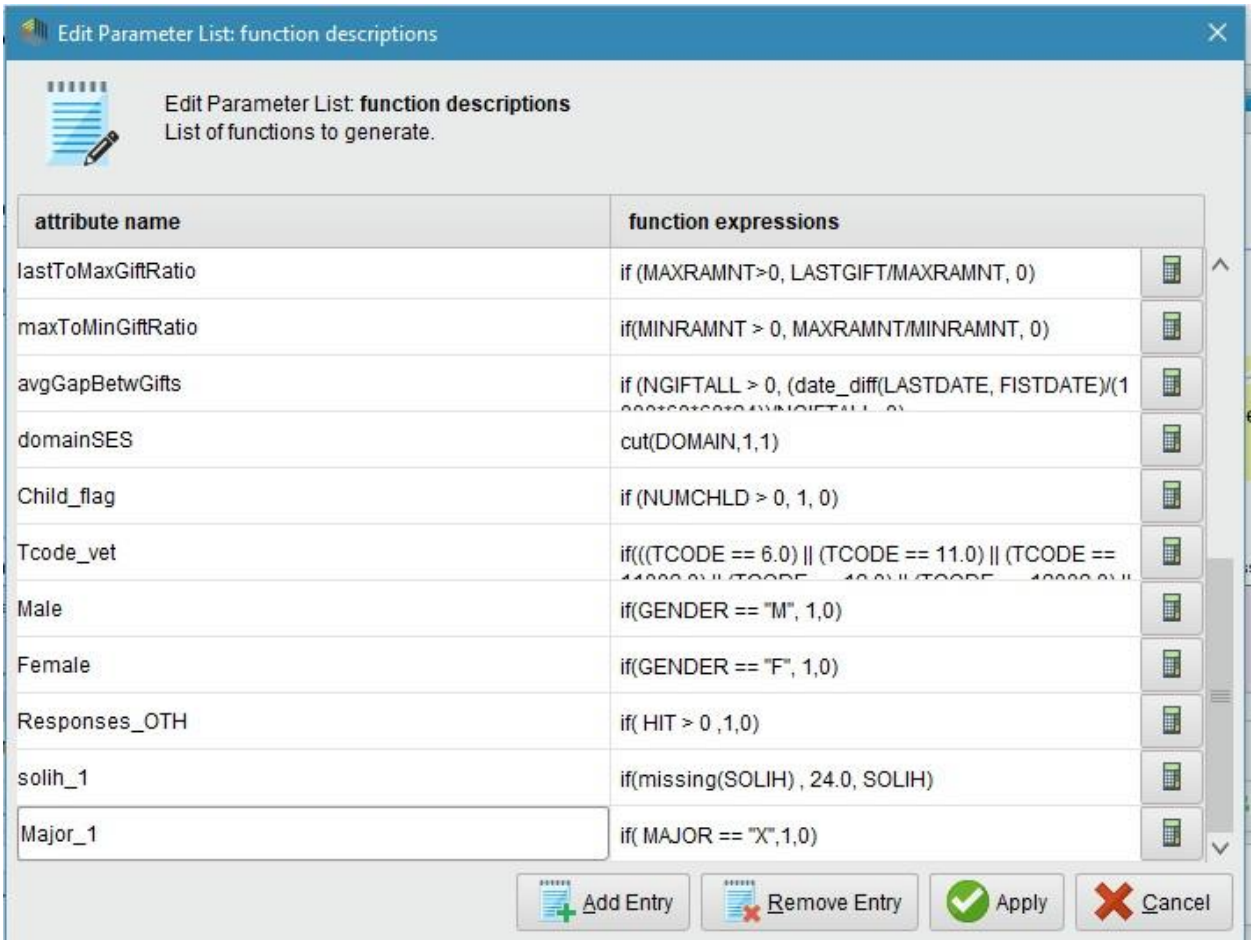
List of attributes dropped:

ADATE_4 to ADATE_23, AGEFLAG, CHILD03, CHILD 07, CHILD 12, CHILD18, DATASRC, DOB, GEOCODE2, HPONE_D, LIFESRC, MAILCODE, MDMAUD, MDMAUD_A, MDMAUD_F, MDMAUD_R, MINRDATE, NEXTDATE, NOEXCH, ODATEDW, OSOURCE, RAMNT_3 TO RAMNT_24, RDATE_3 TO RDATE_24, RFA_2,RFA_2R, RFA_3 TO RFA_24, SOLP3, STATE and ZIP

c) Concatenation of attributes

New attributes were created combining old attributed for easier classification.

For example:



Old attribute: GENDER was transformed to Is_Male with condition if(GENDER == "M", 1,0) to see how gender affects the response

- **Principal Components Analysis (PCA)**
  PCA was performed on highly correlated variables. PCA helps in removing multicollinearity between the variables, if any by converting a set of possible correlated variables into a set of linearly uncorrelated variables. These unrelated variables are called principal components.

Name	**Reason**	**Attributes selected for PCA**	**# PCs**
Response to mail marketing	This attribute measures the response to other mail order offers.	MAGFAML, MAGFEM, MAGMALE, MBBOOKS, MBCOLECT, MBCRAFT, MBGARDEN, PUBCULIN, PUBDOITY, PUBGARDN, PUBHLTH, PUBNEWFN, PUBOPP, PUBPHOTO	5
Interest in Donors	This attribute represents the donor's interest	BIBLE, BOATS, CARDS, CATLG, CDPLAY, CRAFTS, FISHER, GARDENIN, HOMEE, KIDSTUFF, PCOWNERS, PETS, PHOTO, PLATES, STEREO, VETERANS, WALKER	5
Census attributes	To reduce the number of attributes	Variables with importance (weight) greater than 0.3 were picked	18

• Decision Trees • Logistic Regression, using Ridge and Lasso. • Random forest • Boosted trees

I. DECISION TREE (W-J48) PERFORMANCE :

Model 1:

We designed a decision tree model 1 taking into consideration the following factors listed in the table:

Criterion	Information gain
Confidence	0.5
Maximal depth	10
Pre-pruning?	Yes

The performance of the training and test samples of Model 1 were as follows:

Training Data:

accuracy: 76.51%

	true false	true true	class precision
pred. false	6430	1945	76.78%
pred. true	32	11	25.58%
class recall	99.50%	0.56%	

Testing Data:

accuracy: 77.67%

	true false	true true	class precision
pred. false	9737	2816	77.57%
pred. true	4	71	94.67%
class recall	99.96%	2.46%	

We can observe that the true class prediction is very low. In order to increase the number of true class predictions, I have tried other splitting criteria like Gini Index, Gini Ratio and Least Square criterion. It was also observed that the change in Pre-pruning parameters does not increase the performance of the model.

Out of the various splitting criteria mentioned above, it was observed that the performance of the model increased with Gini Index as the splitting criteria. So model 2 was developed with Gini Index as the splitting criterion.

Model 2 :

Designed a decision tree model 2 taking into consideration the following factors listed in the table:

Criterion	Gini Index
Confidence	0.3
Maximal depth	12
Pre-pruning?	No

The performance of the training and test samples of Model 2 were as follows:

Training Data:

accuracy: 86.41%

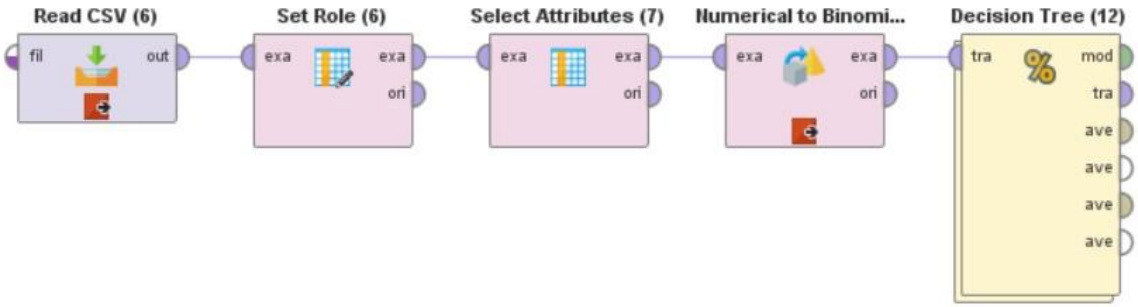
	true false	true true	class precision
pred. false	9581	1556	86.03%
pred. true	160	1331	89.27%
class recall	98.36%	46.10%	

Testing Data:

accuracy: 71.89%

	true false	true true	class precision
pred. false	5772	1676	77.50%
pred. true	690	280	28.87%
class recall	89.32%	14.31%	

We can observe that with Gini index as the splitting criteria, we get a good true positive rate of 46% for the training data and a true positive rate of 14% for the testing data. Therefore, with a confidence interval of 0.3, the model is overfitting the training data.



Snapshot of the RapidMiner code for Decision Trees

II. LOGISTIC REGRESSION PERFORMANCE (RIDGE):

In order to prevent overfitting we use Logistic Regression (Ridge). In this case, we keep the Alpha value as 0 and we change the lambda value and see the performance change of the model.

Model 1 :

λ = 1E-5

Training Data:

accuracy: 56.03%

	true false	true true	class precision
pred. false	5155	966	84.22%
pred. true	4586	1921	29.52%
class recall	52.92%	66.54%	

Testing Data:

accuracy: 54.73%

	true false	true true	class precision
pred. false	3345	694	82.82%
pred. true	3117	1262	28.82%
class recall	51.76%	64.52%	

Model 2 :

$\lambda = 1$

Training Data:

accuracy: 54.58%

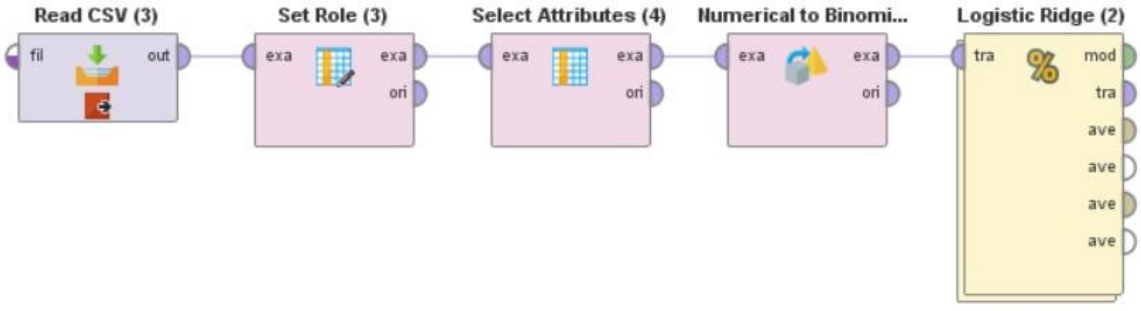
	true false	true true	class precision
pred. false	4907	902	84.47%
pred. true	4834	1985	29.11%
class recall	50.37%	68.76%	

Training Data:

accuracy: 52.99%

	true false	true true	class precision
pred. false	3161	656	82.81%
pred. true	3301	1300	28.25%
class recall	48.92%	66.46%	

As we can see, with the increase in λ value the overall accuracy decreases, recall for true prediction increases and the recall for true negative decreases. So depending on the requirement, we have a trade off between overall accuracy and recall as seen in Model 2.



Snapshot of the RapidMiner code for Logistic Regression (Ridge)

III. LOGISTIC REGRESSION PERFORMANCE (LASSO):

We use Logistic Regression (LASSO) for Interpretability. In this case, I kept the Alpha value as 1 and I change the lambda value and see the performance change of the model.

Model 1:

λ = 1E-8

Training Data:

accuracy: 56.03%

	true false	true true	class precision
pred. false	5155	966	84.22%
pred. true	4586	1921	29.52%
class recall	52.92%	66.54%	

Testing Data:

accuracy: 54.73%

	true false	true true	class precision
pred. false	3345	694	82.82%
pred. true	3117	1262	28.82%
class recall	51.76%	64.52%	

With a lower lambda value, we get good true positive rate for the training and the testing data but we have to compromise with the overall accuracy value.

Model 2:

λ = 0.01

Training Data:

accuracy: 45.01%

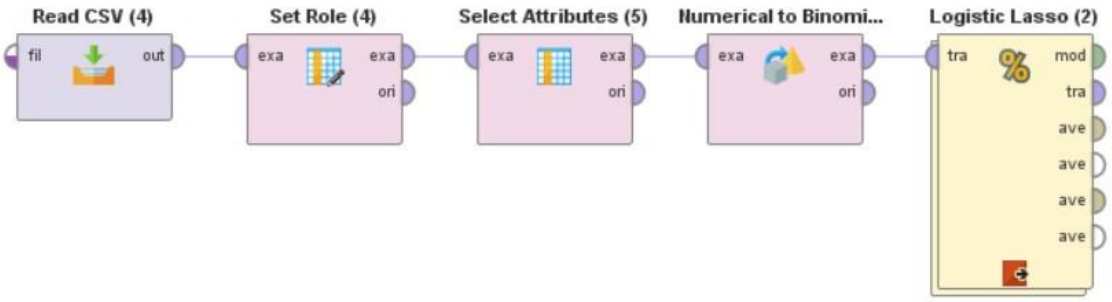
	true false	true true	class precision
pred. false	3436	639	84.32%
pred. true	6305	2248	26.28%
class recall	35.27%	77.87%	

Testing Data:

accuracy: 44.70%

	true false	true true	class precision
pred. false	2246	439	83.65%
pred. true	4216	1517	26.46%
class recall	34.76%	77.56%	

By increasing the lambda value to 0.01, we get a better true positive rate but the accuracy should be compromised like the previous model.



Snapshot of the RapidMiner code for Logistic Regression (Lasso)

IV. RANDOM FOREST PERFORMANCE

Model 1 :

Developed a Random Forest Model 1 , taking into consideration the following factors listed in the table:

Criterion	Information Gain
No. of trees	100
Maximal depth	10
Pre-pruning and Post Pruning?	No

Training Data:

accuracy: 78.13%

	true false	true true	class precision
pred. false	9741	2762	77.91%
pred. true	0	125	100.00%
class recall	100.00%	4.33%	

Testing Data:

accuracy: 76.76%

	true false	true true	class precision
pred. false	6462	1956	76.76%
pred. true	0	0	0.00%
class recall	100.00%	0.00%	

As we can from Model 1, we took the splitting criteria as Information Gain, and for 100 trees we observed that the true class prediction is very low. To improve the performance of the model we must increase the number of trees.

Model 2 :

Developed a Random Forest Model 2 , taking into consideration the following factors listed in the table:

Criterion	Information Gain
No. of trees	500
Maximal depth	10
Pre-pruning and Post Pruning?	No

Training Data:

accuracy: 78.27%

	true false	true true	class precision
pred. false	9741	2744	78.02%
pred. true	0	143	100.00%
class recall	100.00%	4.95%	

Testing Data:

accuracy: 76.76%

	true false	true true	class precision
pred. false	6462	1956	76.76%
pred. true	0	0	0.00%
class recall	100.00%	0.00%	

In Model 2 , we observe that the training error decreases with the increase in the number of trees from 100 to 500 but the true class prediction is still very low. The model prediction might tend to increase with the increase in the maximum depth as in Model 3.

Model 3 :

Developed a Random Forest Model 3 , taking into consideration the following factors listed in the table:

Criterion	Information Gain
No. of trees	300
Maximal depth	20
Pre-pruning and Post Pruning?	No

Training Data:

accuracy: 98.55%

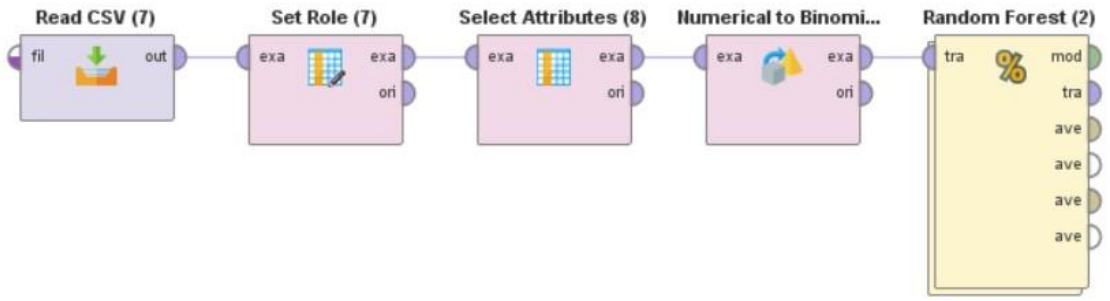
	true false	true true	class precision
pred. false	9741	183	98.16%
pred. true	0	2704	100.00%
class recall	100.00%	93.66%	

Testing Data:

accuracy: 76.74%

	true false	true true	class precision
pred. false	6455	1951	76.79%
pred. true	7	5	41.67%
class recall	99.89%	0.26%	

It can be observed that the training error has decreased drastically but the true class prediction is still very low. With further increase in the depth , the model will overfit and result in increase in the test error.



Snapshot of the RapidMiner code for Random Forests

V. BOOSTED TREES PERFORMANCE

Model 1 :

Developed a Boosted Trees Model 1, taking into consideration the following factors listed in the table:

No. of trees	50
Maximal depth	5
Learning rate	0.1
Minimal rows	10

Training Data:

accuracy: 75.95%

	true false	true true	class precision
pred. false	7830	1126	87.43%
pred. true	1911	1761	47.96%
class recall	80.38%	61.00%	

Testing Data:

accuracy: 66.51%

	true false	true true	class precision
pred. false	4777	1134	80.82%
pred. true	1685	822	32.79%
class recall	73.92%	42.02%	

Model 2 :

Developed a Boosted Trees Model 2 , taking into consideration the following factors listed in the table:

No. of trees	100
Maximal depth	5
Learning rate	0.1
Minimal rows	10

Training Data:

accuracy: 82.51%

	true false	true true	class precision
pred. false	8509	977	89.70%
pred. true	1232	1910	60.79%
class recall	87.35%	66.16%	

Testing Data:

accuracy: 67.96%

	true false	true true	class precision
pred. false	5036	1271	79.85%
pred. true	1426	685	32.45%
class recall	77.93%	35.02%	

As we can see, with increasing the number of boosted trees, the overall accuracy has increased, but the true recall has been decreased. In order to improve the recall, we have to reduce the learning rate value.

Model 3 :

Developed a Boosted Trees Model 3 , taking into consideration the following factors listed in the table:

No. of trees	100
Maximal depth	5
Learning rate	0.05
Minimal rows	10

Training Data:

accuracy: 70.83%

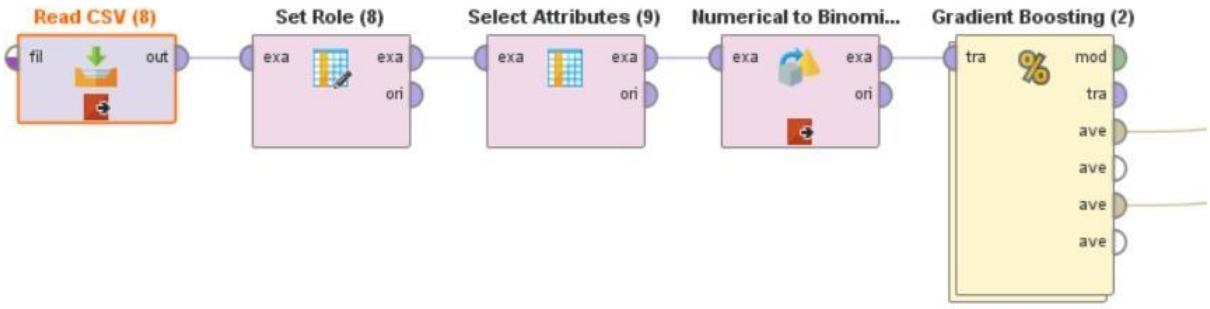
	true false	true true	class precision
pred. false	7236	1179	85.99%
pred. true	2505	1708	40.54%
class recall	74.28%	59.16%	

Testing Data:

accuracy: 64.41%

	true false	true true	class precision
pred. false	4511	1045	81.19%
pred. true	1951	911	31.83%
class recall	69.81%	46.57%	

The true recall rate has been increase from 35.02 to 46.57 by increasing the number of trees and by decreasing the learning rate. By further reducing the learning rate and the number of trees, the model may tend to overfit the training data and decrease the performance.



Snapshot of the RapidMiner code for Boosted Trees

INFERENCE:

Logistic Regression (Lasso) is our best model after analyzing and comparing the performance metric of various techniques.

Followed the following approach for **Variable Selection** :

- We observed the attributes that interests the donors and developed a PCA.
- We then observed the attributes concerning donation campaign responses.
- Census Information : In order to check the variable importance, we developed a Random Forest model and calculated the weights of the trees based on their importance and created a PCA process for the same.
- Based on different techniques for variable selection like Information Gain, Gini Index, Chi-squared available in decision trees, important variables will be selected and used where the base model is a decision tree.

Classification under asymmetric response and cost: What is the reasoning behind using weighted sampling to produce a training set with equal numbers of donors and non-donors?

SOLUTION:

In the given dataset where the quantity of individuals who donated is just 5.1% of the aggregate. This implies that the dataset has a great deal of information relating to the individuals who did not donate. No model will perform well on this dataset if the point is to foresee the donors since there isn't much data identified with the donors in contrast with the non-donors.

To stay away from such a predisposition towards the non-donors, we utilize the weighted example. The weighted example contains 9999 points with 3499 donors and 6500 non-donors. On the off chance that the 5.1% of donor's are taken, at that point there would be 510 donors and around 9489 non-donors. So, the example would have a proportion of roughly 5:95 and the model would tilt towards the non-donors. Hence weighed sampling has been used to reduce this bias. For us, the importance of this situation is to identify and predict the donors accurately to expand benefits. So instead of focusing on accuracy, we need to see how many donors were actually identified as donors. This is the class recall. Therefore, model may exist with low accuracy and high class recall.

In the analysis, Logistic Regression (LASSO) has the highest class recall of 77.5%. This is the best model to the best of our understanding.

accuracy: 44.70%

	true false	true true	class precision
pred. false	2246	439	83.65%
pred. true	4216	1517	26.46%
class recall	34.76%	77.56%	