October 14, 2015

Aditi vishwasrao

Mounica sirineni

nishanth reddy konkala

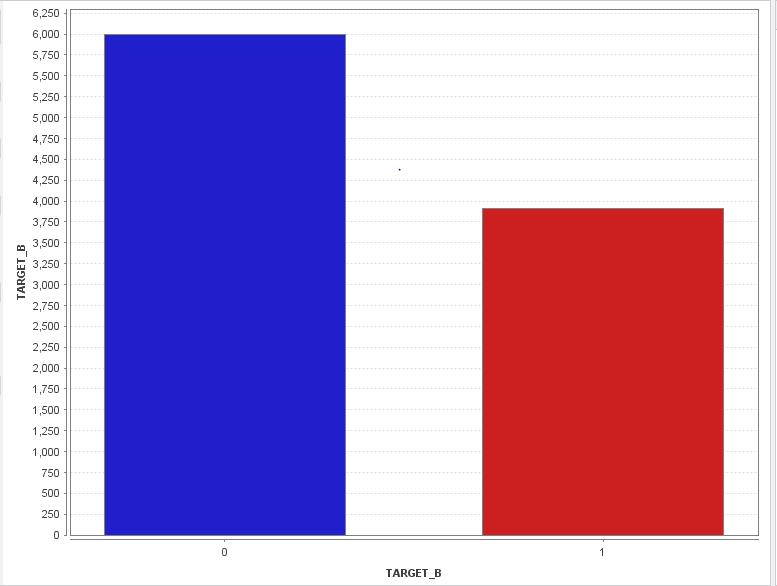
University of Illinois Medical Center

IDS 572 Assignment 2

**1.1 DATA EXPLORATION:**

The data set consists of 9912 data points and 480 attributes. Some of the significant variables in the data set are:

**TARGET\_B:**This is a binomial variable indicating the response of mailing. The distribution shows that data consists of 40% donors (3912) and 60% non-donors (6000).



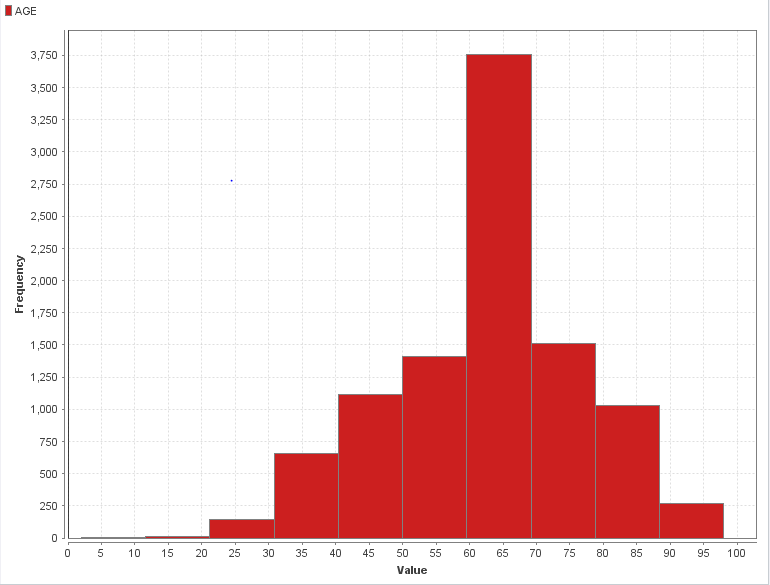
**TARGET\_D:** This variable indicates the donation amount in dollars associated with a donor. Since our aim is to build a classification model for donors and non-donors with maximum profit, TARGET\_D is not considered in the data set.

**Table 1.1 Distributions of some significant variables**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **Min** | **Max** | **Mean** | **Standard Deviation** |
| AGE | 2 | 98 | 61.868 | 14.204 |
| INCOME | 1 | 7 | 3.947 | 1.851 |
| LASTGIFT | 0 | 500 | 16.625 | 13.571 |
| AVGGIFT | 1.5 | 948.5 | 12.878 | 13.116 |
| Percent Home Value >= $200,000 | 0 | 99 | 14.865 | 27.987 |
| Percent Home Value >= $150,000 | 0 | 99 | 22.481 | 33.008 |
| $ 1 or 2 Room Housing Units | 0 | 99 | 4.828 | 7.4 |
| Percent >= 6 Room Housing Units | 0 | 99 | 46.081 | 21.422 |

* Percent home value >= $200,000 or $150,000 indicates the percentage of population in the data set with home value greater than $200,000 or $150,000. So the company can target this group for donation.
* Similarly, the company can target the percentage of donors in data set with housing units >= 6.
* LASTGIFT denotes the dollar amount given in the recent gift. AVGGIFT is the average dollar amount from a donor. The company can target individuals based on their average or last gift amount.

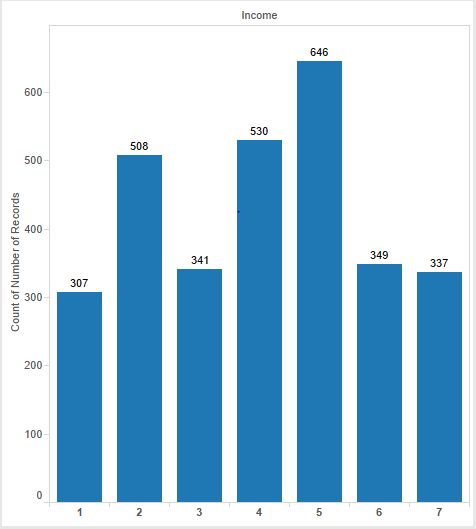
**AGE:**



**Figure 1.1 Distribution of Age**

In the data set, minimum age is 2 and maximum age is 98. But histogram shows us that the proportion of people in the age group 50 to 80 is highest in the data.

**INCOME:**



**Figure 1.2 Distribution of Income**

The graph presents the number of records who donated with their income status. So we can see that maximum number of donors are from the income level of 5. Company can target income levels 2,4,5.

**1.2 DATA CLEANING:**

**1.2.1 Variable transformations:**

When the data set was imported into RapidMiner, we identified some variables have missing values. So we applied variable transformations using the operator “**Generate Attributes**” and generated new variables to replace missing values with their corresponding values. The table below provides a summary of this process. Click here for the table.

**Table 1.2.1 Table of Variable Transformations**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Original Variable** | **Original Value** | **Missing value replacement** | **Variable transformation** | **New variable** |
| PVASTATE | P or E | 0 | if(PVASTATE=="E" ||PVASTATE=="P", "1", "0") | EP\_PvaState |
| RECINHSE | X | 0 | if( RECINHSE == "X", "1","0") | rechinse |
| RECP3 | X | 0 | if( RECP3 == "X", "1", "0") | recp3 |
| RECPGVG | X | 0 | if( RECPGVG == "X", "1", "0") | recpgvg |
| RECSWEEP | X | 0 | if( RECSWEEP == "X", "1", "0") | recsweep |
| DOMAIN | C,R,S,T,U | - | cut(DOMAIN,0,1) | DOMAIN\_Type |
| MAJOR | X | 0 | if(MAJOR=="X", "1", "0") | major |
| PEPSTRFL | X | 0 | if(PEPSTRFL=="X", "1", "0") | pepstrfl |
| CHILD03, CHILD07, CHILD08, CHILD12 | M, F, B | 0 | if(Variable=="M" || Variable =="F"|| Variable =="B", "1", "0") | child3, child7, child12, child18 |
| HOMEOWNR | H | 0 | if(HOMEOWNR=="H", "1", "0") | homeownr |
| - | - | - | date\_diff(LASTDATE, ADATE\_2)/(1000\*60\*60\*24) | totDays |
| DOMAIN | - | - | DOMAIN | DOMAIN\_SES |

This process gives us a data set of 495 variables. To generate new attribute DOMAIN\_SES to store the Social Economic Status of a donor, we have used “**Cut**” operator to cut the second byte of variable DOMAIN and store it in the variable DOMAIN\_SES.

**1.2.2 DATA REDUCTION:**

Since we have generated new attributes in previous step, the next step is to remove the old variables which were used to generate new attributes. This is completed using the operator “**Remove old attributes**”. [Click here](#removeold) to view table 8.1 of the variables removed from the data set.

The next step is to “**Remove useless attributes**”. Since all the 480 variables are not required for the analysis. We have removed variables which do not provide significant information about response of a donor. [Click here](#removeuseless) to view table 8.2 the removed attributes.

**1.2.3 REPLACE MISSING VALUES:**

To replace missing values we have used the operator “**Map**”. This operator maps specified values of selected attributes to new values. In these variables, “?” is replaced by “N”.[Click here](#map) to view table 8.3 the variables mapped in this process.

In the next step, operator “**Replace Missing Values**” is used replace missing values in selected attributes by a specified replacement. The [table 8.4](#replacemiss)provides the summary of this process.

At the end of this step, we have 364 attributes. There are subsets of variables which gives almost same information. So we implement Principal Component Analysis to select dominant variables from each subset to eliminate redundancy.

**1.2.4 PRINCIPAL COMPONENT ANALYSIS:**

**Need to perform PCA:** In the data set, there are almost 100 variables which reflect characteristics of the donor’s neighborhood. To eliminate redundancy in the data set and improve efficiency of model using minimum variables, these variables have been given to PCA. First the select variables are normalized for the scale 0.0 to 1.0 and used a cutoff threshold as 0.9. We tried with different threshold values. But we got better results with 0.9 as threshold.

**1.2.4.1 Variables given to Neighbor PCA:**

**Table 1.2.2 Table of Neighbor PCA variables**

|  |  |
| --- | --- |
| **Variables** | **Meaning** |
| AGE 901 to AGE 907 | Age of population |
| ETH1 to ETH16 | Ethnicity |
| ETHC1 to ETHC6 | Race |
| POP901 to POP903 | Number of people |
| POP90C1 to POP90C5 | Population in different areas and gender |

At the end of Neighbor PCA step, we have 328 attributes.

**1.2.4.2 Variables given to Interests PCA are:**

First nominal variables are converted to numeric type and PCA is applied. These variables represent the interests of a donor.

COLLECT1 VETERANS BIBLE CATLG HOMEE PETS CDPLAY

STEREO PCOWNERS PHOTO CRAFTS FISHER GARDENIN BOATS

WALKER KIDSTUFF CARDS PLATES

At the end of Interests PCA, we have 319 attributes.

**1.2.4.3 Variables given to OtherTypeMail PCA are:**

These variables indicate the number of known times the donor has responded to other types of mail order offers.

MBCCRAFT MBGARDEN MBBOOKS MBCOLLECT

MAGFAML MAGFEM MAGMALE PUBGARDN

PUBCULIN PUBHLTH PUBDOITY PUBNEWFN

PUBPHOTO PUBOPP

At the end of OtherTypeMail PCA, we have 309 attributes.

**1.2.5** **EXCLUDED VARIABLES:**

After conducting PCA, the following variables have been removed from data set. These variables do not provide much information to determine donor/non-donor.[Click here](#excludedvar) to view the table 8.5 of excluded variables.

Their information is be provided by other significant variables. So to remove redundant information, they have been removed from dataset. A subset of nominal variables are converted to numeric type.

**1.3 VARIABLES CONSIDERED FOR MODEL ARE:**

At the end of data exploration, we have 62 variables in the data set. Some of the significant variables in the final data set are:

AGE AVGGIFT DOMAIN\_SES DOMAIN\_TYPE GENDER

INCOME LASTGIFT TARGET\_B WEALTH

**2. MODELING**

The data set is partitioned into 60% for training and 40% for validation. Initial random seed is set to 12345. In this step we run 6 types of models on training and validation data. We change the subset of variables in the data set by considering each PCA subset at a time. This is done by adding/removing each PCA block in the rapid miner process. The final data set consists of 3 PCA subsets and other attributes. Therefore, we will have 5 sets of variables.

* All PCA values considered: PCA values of three subsets are considered along with other attributes for modeling.
* Neighborhood PCA values removed: PCA values of Neighborhood subset are not considered for modeling.
* Interests PCA values removed: PCA values of Interests subset are not considered for modeling.
* Other type mail PCA values removed: PCA values of Other type mail subset are not considered for modeling.
* Remove PCA values of all three subsets: PCA values of all three subsets are not considered for modeling.

**2.1 NAIVE BAYES MODEL:**

Using Naive Bayes modelling technique, models can be built by either including Laplace correction or not. In both cases, the performance of models by including the above subsets are found to be same.

The only parameter for Naïve Bayes model is Laplace correction. We have tried using with and without Laplace correction. But the performance did not change. So we built Naïve Bayes model using Laplace correction. We performed the modeling for different subsets of variables explained in the 1st section.

**Table 2.1.1 Performance of Naïve Bayes model**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenarios** | **Training Accuracy** | **Validation Accuracy** | **Validation Recall (true 1)** |
| All PCA values considered | 45.65% | 43.53% | 87.15% |
| Neighborhood PCA values removed | 40.12% | 39.17% | 99.29% |
| Interest PCA values removed | 46.66% | 44.31% | 85.47% |
| Other type mail PCA values removed | 45.65% | 43.53% | 87.15% |
| Remove PCA values of all three categories | 40.07% | 39.17% | 99.35% |

**Conclusion:**The model highlighted above by removing the Interest PCA values is the best model, as it is the most accurate and has a decent test recall rate. [Click here](#naive) to view the screenshots of training and validation accuracy for all scenarios.

**2.2 K-NEAREST NEIGHBOR MODEL:**

The K-Nearest Neighbour algorithm is based on learning by analogy, that is, by comparing a given test example with training examples that are similar to it. The training examples are described by n attribute. Our parameters are as follows: ‘Mixed Euclidean Distance’, within the mixed measure type, to find K- nearest neighbours.

Considering the accuracy and recall of KNN models with different K values, we have reached the conclusion that the accuracy as well as recall is highest at K =33. We have shown the computations of model performance at k= 1, 19, 33 and 51. As we can see from the tables below, the values of accuracy are a curve peaking at k = 33.

**Table 2.2.1 K-Nearest Neighbour model performance with K = 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenarios** | **Training Accuracy** | **Testing Accuracy** | **Testing Recall (true 1)** | **K** |
| All PCA values considered | 100.00% | 53.54% | 40.41% | 1 |
| Neighborhood PCA values removed | 99.75% | 53.57% | 40.48% | 1 |
| Interest PCA values removed | 100.00% | 53.37% | 40.54% | 1 |
| Other type mail PCA values removed | 100.00% | 53.54% | 40.41% | 1 |
| Remove PCA values of all three categories | 99.63% | 53.22% | 40.15% | 1 |

**Table 2.2.2 K-Nearest Neighbour model performance with K = 19**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenarios** | **Training Accuracy** | **Testing Accuracy** | **Testing Recall (true 1)** | **K** |
| All PCA values considered | 62.33% | 56.27% | 55.46% | 19 |
| Neighborhood PCA values removed | 62.17% | 56.27% | 55.33% | 19 |
| Interest PCA values removed | 62.23% | 56.72% | 55.84% | 19 |
| Other type mail PCA values removed | 62.33% | 56.27% | 55.46% | 19 |
| Remove PCA values of all three categories | 62.13% | 55.52% | 55.58% | 19 |

**Table 2.2.3 K-Nearest Neighbour model performance with K = 33**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenarios** | **Training Accuracy** | **Testing Accuracy** | **Testing Recall (true 1)** | **K** |
| All PCA values considered | 56.84% | 51.75% | 69.98% | 33 |
| Neighborhood PCA values removed | 56.94% | 51.98% | 70.11% | 33 |
| Interest PCA values removed | 56.92% | 51.70% | 69.92% | 33 |
| Other type mail PCA values removed | 56.84% | 51.75% | 69.98% | 33 |
| Remove PCA values of all three categories | 57.04% | 51.85% | 70.05% | 33 |

**Table 2.2.4 K-Nearest Neighbour model performance with K = 51**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenarios** | **Training Accuracy** | **Testing Accuracy** | **Testing Recall (true 1)** | **K** |
| All PCA values considered | 56.25% | 52.31% | 68.04% | 51 |
| Neighborhood PCA values removed | 56.35% | 52.23% | 67.85% | 51 |
| Interest PCA values removed | 56.23% | 52.26% | 68.11% | 51 |
| Other type mail PCA values removed | 56.25% | 52.31% | 68.04% | 51 |
| Remove PCA values of all three categories | 62.13% | 56.52% | 55.58% | 51 |

**Conclusion:** As we can conclude from the above tables the best accuracy and recall can be seen when K=33 with the subset “Neighborhood values” removed. [Click here](#knn) to view the screenshots of training and validation performance of KNN model.

**2.3 DECISION TREE-J-48**

A **decision tree** is a **decision** support tool that uses a **tree**-like graph or model of **decisions** and their possible consequences, including chance event outcomes, resource costs, and utility. This model (from the WEKA extension) creates a decision tree using the C4.5 algorithm.

By changing the parameters like pruning, confidence and the minimum number at leaf, we got the following results to be the best .The parameters used are:

U=Unchecked

C=0.5

M=8

Un-pruning the tree didn’t show any significant effect (Increase) on accuracy. In fact, accuracy and positive’s recall decreased by nearly 1%.

**Table 2.3.1 Decision Tree model performance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenarios** | **Training Accuracy** | **Validation Accuracy** | **Validation Recall** |
| All PCA values considered | 72 | 55.49 | 49.06 |
| Neighborhood PCA values removed | 70.62 | 54.80 | 48.81 |
| Interest PCA values removed | 70.15 | 56.65 | 51.78 |
| Other type mail PCA values removed | 72 | 55.49 | 49.06 |
| Remove PCA values of all three categories | 65.13 | 53.54 | 57.46 |

**Conclusion:** The best model we got is for the last case, i. e; when all PCA’s are removed. In this case we observe that the difference between the training and validation accuracies is minimum and the validation recall is also high.[Click here](#decisiontree) to view the screenshots of training and validation accuracy of Decision tree model.

**2.4 W-LOGISTIC REGRESSION:**

Logistic Regression is a technique used for building and using a multinomial logistic regression model to predict a binary response, in this case, to distinguish a donor from non-donor.

The same scenarios were tested for logistic regression with Laplace correction. We used the default values for R & M. The results for the model are as under:

**Table 2.4.1 Logistic regression model performance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenarios** | **Training Accuracy** | **Validation Accuracy** | **Validation Recall** |
| All PCA values considered | 49.13 | 47.11 | 84.44 |
| Neighborhood PCA values removed | 47 | 45.62 | 89.15 |
| Interest PCA values removed | 48.78 | 46.73 | 85.09 |
| Other type mail PCA values removed | 49.13 | 47.11 | 84.44 |
| Remove PCA values of all three categories | 47.52 | 46.15 | 88.38 |

**Conclusion**: The performance of the model on different PCA Considerations didn’t change much. The best model we got is when we consider all the PCA’s in the model. Surprisingly, the performance metrics match with “Other type Mail PCA values removed”.[Click here](#logistic) to view the screenshots of training and validation accuracy of Logistic regression model.

**2.5 RANDOM FORESTS:**

Random forests grows many classification trees. Each tree gives a classification and votes for that class. The forest chooses the classification having the most votes.

To determine the best model for random forests, the first step is to set the parameters. We have used different parameter sets and results are shown in the table below.

**Table 2.5.1 Random forest performance with different “Number of Trees”**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of trees** | **Training Accuracy** | **Validation Accuracy** | **Recall (true 1)** |
| 47 | 62.94 | 60.96 | 26.47 |
| 48 | 61.11 | 58.79 | 44.74 |
| 49 | 62.91 | 61.34 | 26.4 |

The table shows that we get maximum recall for true 1 at number of trees at 48.

**Table 2.5.2Random forest performance with different “Criteria”**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Training Accuracy** | **Validation Accuracy** | **Recall (true 1)** |
| Gain ratio | 61.11 | 58.79 | 44.74 |
| Information gain | 59.31 | 57.65 | 47.13 |
| Gini index | 59.86 | 57.78 | 51.52 |

From the above table we can say that training and validation accuracy are almost same for 3 criteria. But we get maximum recall for true 1 using Gini Index as criteria. Therefore, the parameters used to build the model for Random Forests are:

Number of trees = 48

Criterion = Gini Index

Maximal depth = 20

Apply pruning = Checked

Confidence = 0.25

Apply prepruning = Checked

Minimal gain = 0.01

Minimal leaf size = 2

Minimal size for split = 2

Number of prepruning alternatives = 3

Using the above parameters, we have used 5 subsets of variables to evaluate the performance of model for Random forests. The below table provides the summary of performance for each subset.

**Table 2.5.3 Random forest performance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenarios** | **Training Accuracy** | **Validation Accuracy** | **Recall (true 1)** |
| All PCA values considered | 58.47 | 55.01 | 59.13 |
| Neighborhood PCA values removed | 62.43 | 57.48 | 47.71 |
| Interest PCA values removed | 58.37 | 54.91 | 57.84 |
| Other type mail PCA values removed | 58.16 | 55.51 | 56.1 |
| Remove PCA values of all three categories | 60.32 | 58.56 | 47 |

**Conclusion:** From the above tables, two subsets “All PCA variables included” and “Interest PCA variables removed” has high Recall (true 1). But when the minimum difference between training and validation accuracy is considered, we choose the best model with all PCA variables included in the data set for Random Forests.[Click here](#randomforest) to view the screenshots of training and validation accuracy of Random forests model.

**2.6 SUPPORT VECTOR MACHINES:**

Support vector machine uses nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane (decision boundary). With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane.

To determine the best model for support vector machines, the first step is to set the parameters. We have used different parameter sets and results are shown in the table below.

**Table 2.6.1 SVM performance with different “C”**

|  |  |  |  |
| --- | --- | --- | --- |
| **C** | **Training Accuracy** | **Validation Accuracy** | **Recall (true 1)** |
| 0 | 44.38 | 42.57 | 91.8 |
| 1 | 46.7 | 45.07 | 87.48 |

The table shows that we get maximum recall for true 1 at C = 1.

**Table 2.6.2 SVM performance with different “Kernel type”**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kernel Type** | **Training Accuracy** | **Validation Accuracy** | **Recall (true 1)** |
| Dot | 44.38 | 42.57 | 91.8 |
| Radial | 98.86 | 43.46 | 81.92 |
| Polynomial | 54.62 | 48.12 | 72.24 |

From the above table we can say that when we use Kernel type as “Dot”, the difference between training and validation accuracy is minimum and Recall for true 1 is maximum. Therefore, the parameters used to build the model for SVM are:

Kernel type = dot

Kernel cache = 200

C = 0.0

Convergence epsilon = 0.001

Max iterations = 100000

Scale = checked

L pos = 1.0

L neg = 1.0

Epsilon = 0.0

Epsilon plus = 0.0

Epsilon minus = 0.0

Using the above parameters, we have used 5 subsets of variables to evaluate the performance of model for Random forests. The below table provides the summary of performance for each subset.

**Table 2.6.3 SVM performance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenarios** | **Training accuracy** | **Validation accuracy** | **Recall (true 1)** |
| All PCA values considered | 44.38 | 42.57 | 91.8 |
| Neighborhood PCA values removed | 41.21 | 39.9 | 97.87 |
| Interest PCA values removed | 45.65 | 44.11 | 91.35 |
| Other type mail PCA values removed | 44.38 | 42.57 | 91.8 |
| Remove PCA values of all three categories | 41.69 +/- 0.65 | 40.69 +/- 0.62 | 96.8 |

**Conclusion:** From the table, we can say that the best model is the one which neighborhood PCA values removed from the data set. [Click here](#svm) to view the screenshots of training and validation accuracy for Support Vector Machines model.

**3. CLASSIFICATION UNDER ASYMMETRIC RESPONSE AND COST**

The reasons behind using weighted sampling to produce training set of equal number of donors and non-donors are as follows:

1. The response rate of the data is 5.1%, so when we use random sampling to build the tree, there is high probability of model being trained on non-donor’s data and it is highly likely that the model predicts donor as non-donor when tested on validation and testing data. In order to avoid this danger, we weight the sampling in order to produce equal number of donors and non-donors.
2. A model should always be built on rare-cases in the data in order to perform well and to avoid bias; therefore we use weighted sampling so as to include equal number of donors and non-donors.

Because the samples are weighted, the classification accuracy is not a good performance metric to maximize the net profit.

We must either weight the profit values according to the split in the data or adjust the confusion matrix based on the split and then calculate the net profit for each model and for each classification type (decision tree, random forests, KNN etc.). The model and its parameters which is giving the highest profit must be selected to test the new data in order to maximize the profit.

After testing, the cumulative profit values will give us the max profit. And the solicitations should be sent to people who have the threshold values higher than the corresponding threshold value of the max cumulative profit.

**4. CALCULATE NET PROFIT**

We have assumed weighted sampling (40% responders and 60% non-responders) in our model. Therefore we have weighted the profit and cost for Net Profit Calculation.

**Given:**

Profit= $13-$0.68=$12.32

Cost=$0.68

**Adjusted Values:**

Profit= (12.32\*0.051)/0.4= 1.571

Cost= (0.68\*0.9491)/0.6= 1.076

So, for Net Profit Calculation, if TARGET\_B=1, then profit is 1.571 else it’s -1.076.

The Net Profit is calculated as follows:

* Sort the data on confidence(1)
* Compare the actual and the predicted donors
* Profit associated with predicting a donor correctly = $1.571
* Loss associated with mailing to non-donors predicted as donors = -$1.076
* Opportunity loss of actual donors not predicted as donors = -$1.571

**Table 4.1 Best model from each of the techniques in Question 2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Accuracy/ Models** | **KNN** | **W J48** | **W Logistic** | **Naïve Bayes** | **SVM** | **Random Forests** |
| Training Accuracy | 56.94 | 65.13 | 49.13 | 40.07 | 41.21 | 58.47 |
| Validation Accuracy | 51.98 | 53.54 | 47.11 | 39.17 | 39.9 | 55.01 |

**Table 4.2 Lift of net profit for best models**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Models** | **KNN (K=33)** | **W J48** | **W Logistic** | **Naïve Bayes** | **SVM** | **Random Forests** |
| Training Dataset | 779.531 | 1361.645 | 600.164 | 492.848 |  | 603.929 |
| Validation Dataset | 219.052 | 162.883 | 262.882 | 202.234 | 261.756 | 269.219 |

**4.1 LIFT CALCULATION:**

At max profit, the difference between cumulative net profit of the model and the no model is the lift. To be more precise, it is how much better/worse the model is performing compared to no model (Which randomly predicts 50% points as ‘1’ and 50% points as ‘0’).

In our model, there are 5947 training cases and 3965 cases in validation (rounded of).

**Training:**

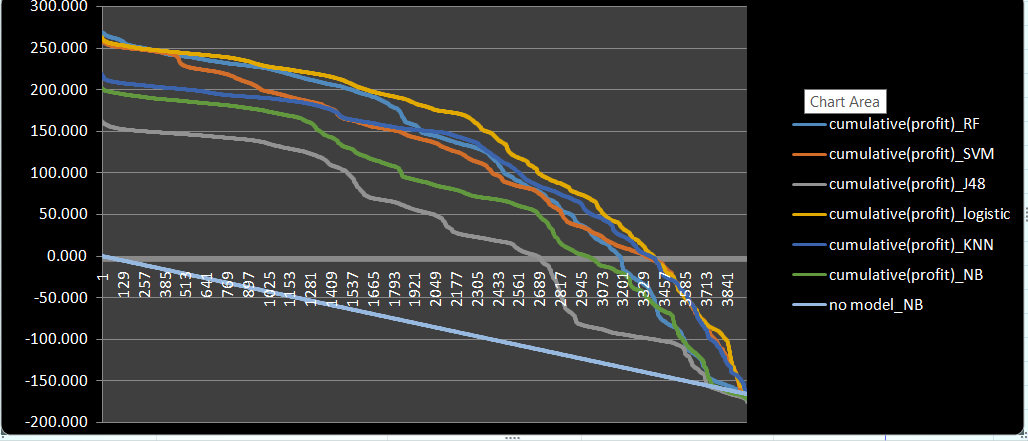
With No model, the Max. Cumulative profit will be (5947/2)\*1.571-(5947/2)\*1.076 = 1471.8825

**Validation:**

With No model, The Max. Cumulative profit will be as follows: (3965/2)\*1.571-(3965/2)\*1.076=981.3375

**5. DRAW LIFT CURVES**

The below chat shows the max. Cumulative profit of all the best models validation set. The performance of W-logistic and Random forests is almost same, but random forests exceeds the profit given by logistic by 6 units.



Conclusion:

Random forests dominate with Maximum Cumulative profit of 269.2

**6. BEST MODEL**

In defining the best model, all the parameters like training, validation’s accuracy and recall Max cumulative profit and threshold to maximize the number of donors predicted so that we can maximize the donation and minimize the solicitation cost

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **Training Accuracy** | **Validation Accuracy** | **Validation Recall** | **Validation Max Cumulative profit** | **Threshold** |
| Naïve Bayes | 40.07 | 39.17 | 99.35 | 202.2 | 0.041 |
| KNN | 56.94 | 51.98 | 70.11 | 291 | 0.5 |
| W-J48 | 65.13 | 53.54 | 57.46 | 162.840 | 0.6 |
| W-logistic | 49.13 | 47.11 | 84.44 | 262.840 | 0.4 |
| SVM | 41.21 | 39.9 | 97.87 | 261.7 | 0.3 |
| Random Forests | 58.47 | 55.01 | 59.13 | 269.2 | 0.1 |

The best model in terms of accuracy is: Random forests

The best model in terms of Recall is: Naïve Bayes

The best model in terms of Net profit is: KNN

**Conclusion:**Random Forests has high accuracy but less validation recall (true 1). That means less number of actuals 1’s are predicted as 1’s. Naïve Bayes has high validation recall but least accuracy of all models. KNN has high profit but difference between training and validation for KNN is high. So, as a tradeoff between accuracy, validation recall and net profit, we choose W-logistic as the best model.

**7. TESTING**

We applied our best model i.e W-Logistic and it has predicted the following number of donors and non-donors:

 Number of donors: **1408**

Number of non-donors: **18592**

The cutoff used to predict donors and non-donors is **0.5.**

The excel sheet containing the results has been attached below.

Also the RapidMiner process has been shown in the appendix.

**APPENDIX:**

**8.1 Table of removed old attributes** [**Back to the top**](#backold)

|  |  |  |  |
| --- | --- | --- | --- |
| CHILD03 | CHILD07 | CHILD12 | CHILD18 |
| DOMAIN | MAJOR | HOMEOWNR | PEPSTRFL |
| PVASTATE | RECINHSE | RECP3 | RECPGVG |
| RECSWEEP |  |  |  |

**8.2 Table of removed useless attributes** [**Back to the top**](#backuseless)

|  |  |
| --- | --- |
| **Removed attributes** | **Meaning** |
| ADATE2 to ADATE24 | Date the promotion was mailed |
| AGEFLAG | Age flag |
| RFA\_2 to RFA\_24, RFA\_2A, F, R | Donor’s RFA status |
| RDATE\_3 to RDATE\_24 | Date the gift was received |
| RAMNT\_3 to RAMNT\_24 | Dollar amount of the gift |
| MINRDATE, MAXRDATE, LASTDATE, FISTDATE, NEXTDATE, MAXADATE | Variables from giving history file |
| CLUSTER2 | Classic cluster code |
| DATASRCE | Source of Overlay Data. Indicates which third-party data source the donor matched against |
| GEOCODE, GEOCODE2 | Geography |
| TCODE, ODATEDW, OSOURCE, MAILCODE, NOEXCH, DOB | Donor’s information |
| LIFESRC | Life style data source |
| ZIP | Zip code |
| SOLIH, SOLP3 | Solicitation code |

**8.3 Mapping** [**Back to the top**](#backmap)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BIBLE | BOATS | CARDS | CATLG | CDPLAY | COLLECT1 |
| CRAFTS | GARDENIN | FISHER | HOMEE | KIDSTUFF | PCOWNERS |
| PETS | STEREO | PHOTO | VETERANS | PLATES | WALKER |

**8.4 Replace missing values** [**Back to the top**](#backmiss)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute filter type** | **Value type** | **Attribute** | **Replace with** | **Replenishment value** |
| Value\_type | Numeric | AGE | Average | -1 |
| NUMCHILD | Zero | -1 |
| Single |  | GENDER |  | U |
| DOMAIN\_SES | 0 |
| DOMAIN\_Type | M |
| CLUSTER | 0.0 |

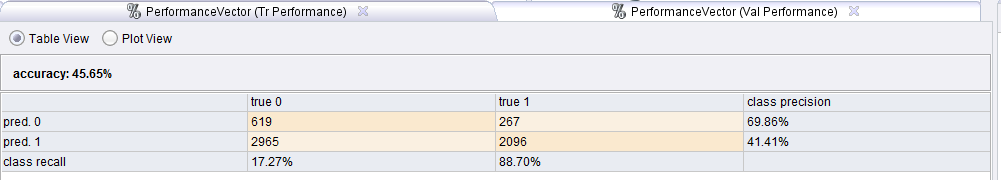
**8.5 Table of Excluded variables** [**Back to the top**](#backexc)

|  |  |
| --- | --- |
| **Variables** | **Meaning** |
| AC1 to AC2, AGEC1 to AGEC7 | Age |
| ADI, DMA, MSA | Code |
| AFC1 to AFC6, VC1 to VC4 | Percent in military or veterans |
| ANC1 to ANC15 | Ancestry |
| CHIL1 to CHIL3, CHILC1 to CHILC5, HHD1 to HHD12 | Percent of children |
| DW1 to DW9, HU1 to HU5, HUPA1 to HUPA7, HUR1 to HUR2, HV1 to HV4, HVP1 to HVP6, RHP1 To RHP4 | Home structure |
| EC1 to EC8, SEC1 to SEC5 | Education |
| EIC1 to EIC16, OCC1 to OCC13, OEDC1 to OEDC7, PEC1 to PEC2 | Employment |
| HC1 to HC21 | Residence type, utilities, etc |
| HHAGE1 to HHAGE3 | Elderly person in house |
| HHAS1 to HHAS4 | Social status |
| HHN1 to HHN6, HHP1 to HHP2, MC1 to MC3 | Number of persons |
| IC1 to IC23, MHUC1 to MHUC2, RP1 to RP4 | Income |
| LFC1 to LFC10 | Labor Force |
| LSC1 to LSC4 | Language |
| MARR1 to MARR4 | Marital status |
| POBC1 to POBC2 | Native country |
| TPE1 to TPE13, VOC1 to VOC3 | Vehicle |

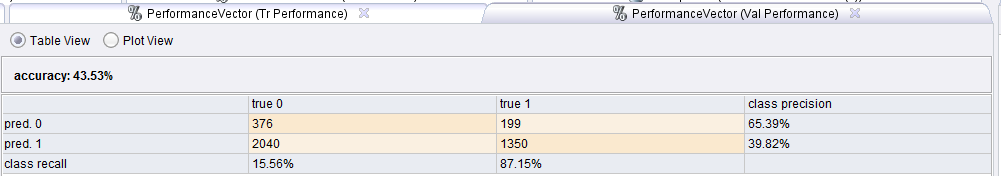
**8.6 Naive Bayes model performance:** [**Back to the top**](#backnaive)

**8.6.1 All PCA values considered:**

**Training Accuracy**

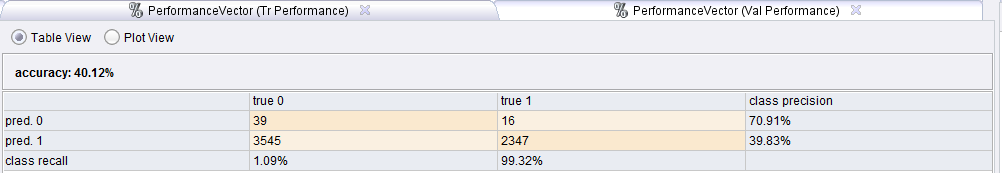


**Validation Accuracy**

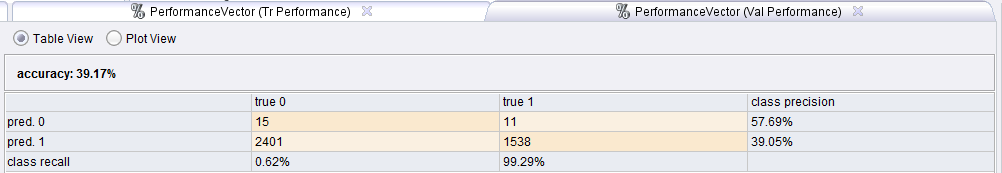


**8.6.2 Neighborhood PCA values removed:**

**Training Accuracy:**

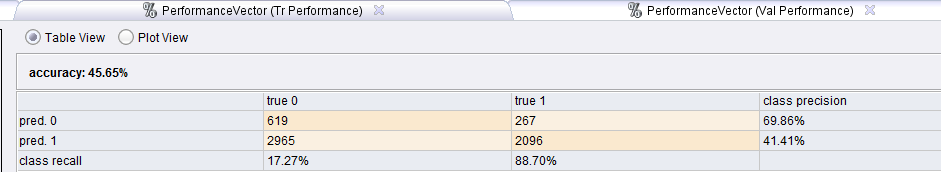


**Validation Accuracy:**

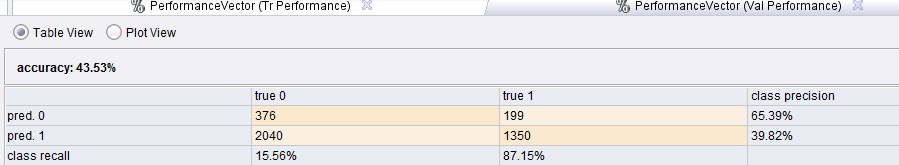


**8.6.3 Other Type mail PCA values removed:**

**Training Accuracy:**

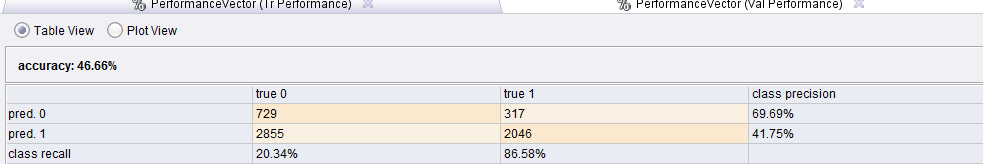


**Validation Accuracy:**

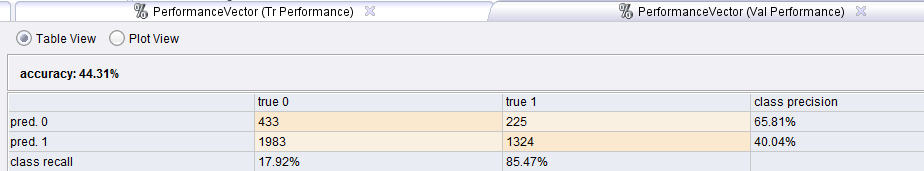


**8.6.4 Interest PCA values removed:**

**Training Accuracy**

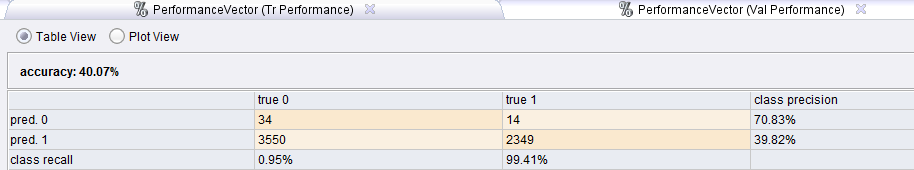


**Validation Accuracy**

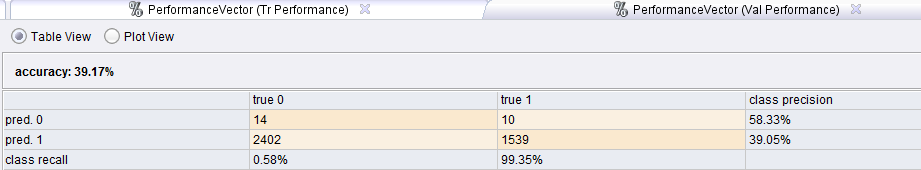


**8.6.5 All three PCA subsets removed:**

**Training Accuracy**



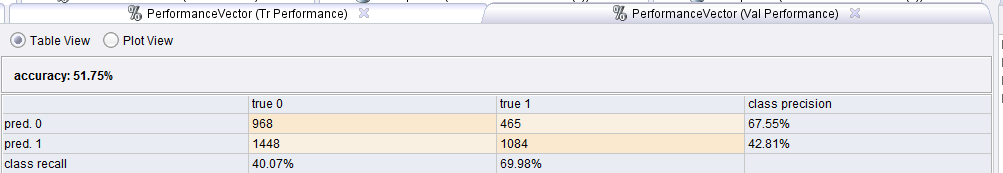
**Validation Accuracy**



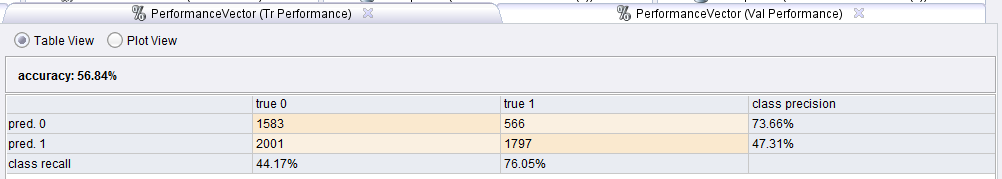
**8.7 K-Nearest Neighbor model performance** [**Back to the top**](#backknn)

**8.7.1 All PCA values considered**

**Training accuracy**

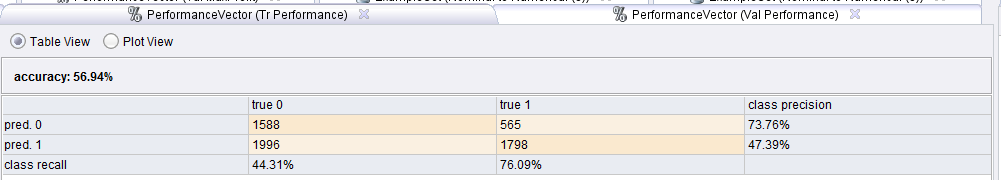


**Validation accuracy**

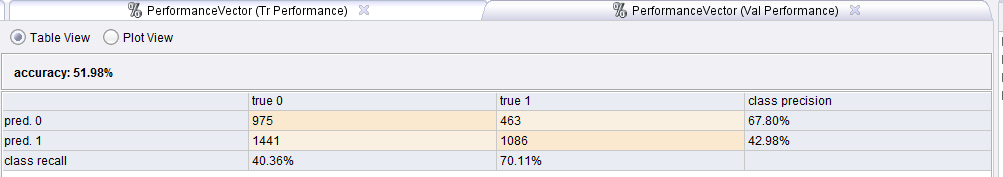


**8.7.2 Neighborhood PCA values removed**

**Training accuracy**

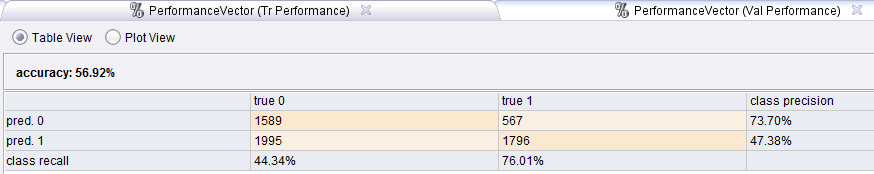


**Validation accuracy**

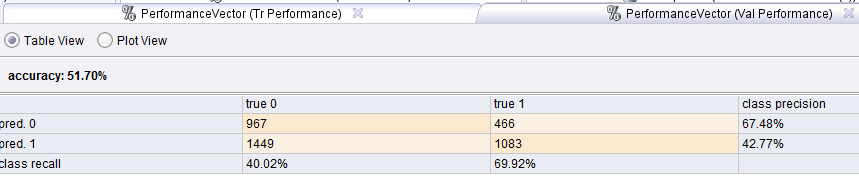


**8.7.3 Interest PCA values removed**

**Training accuracy**

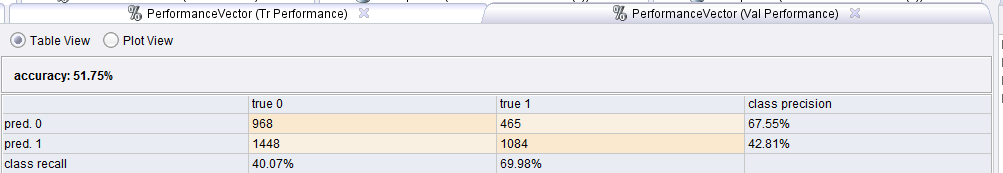


**Validation accuracy**

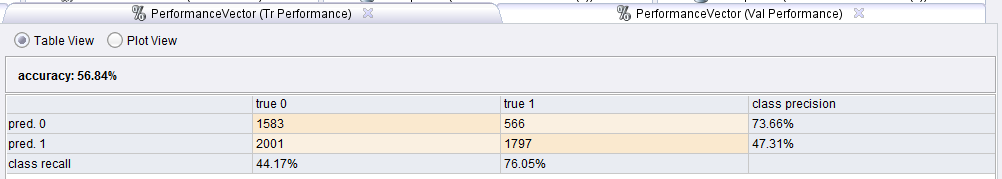


**8.7.4 Other type mail PCA values removed**

**Training accuracy**

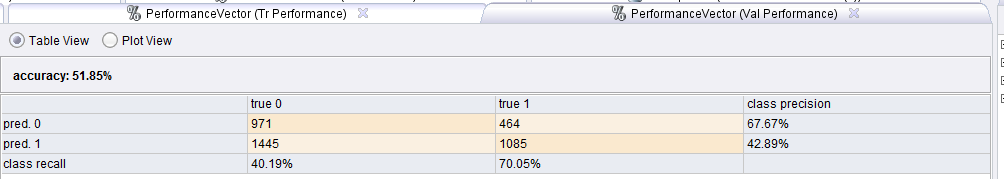


**Validation accuracy**

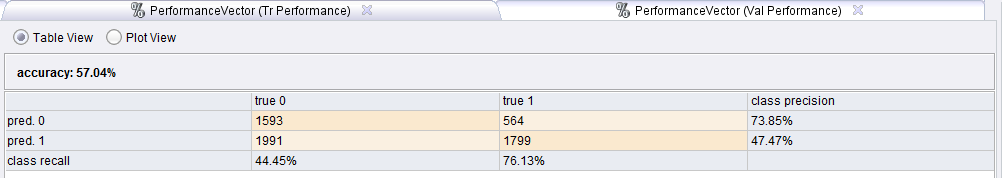


**8.7.5 Remove all PCA values**

**Training accuracy**

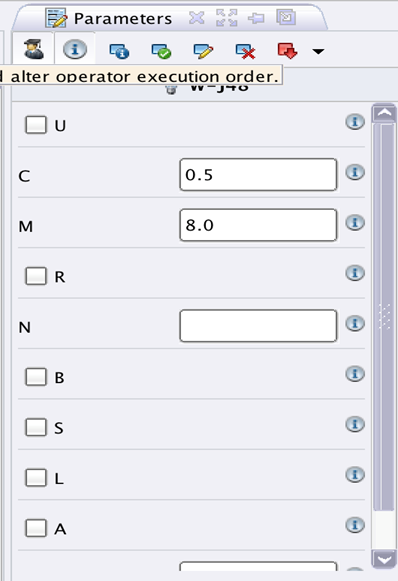


**Validation accuracy**



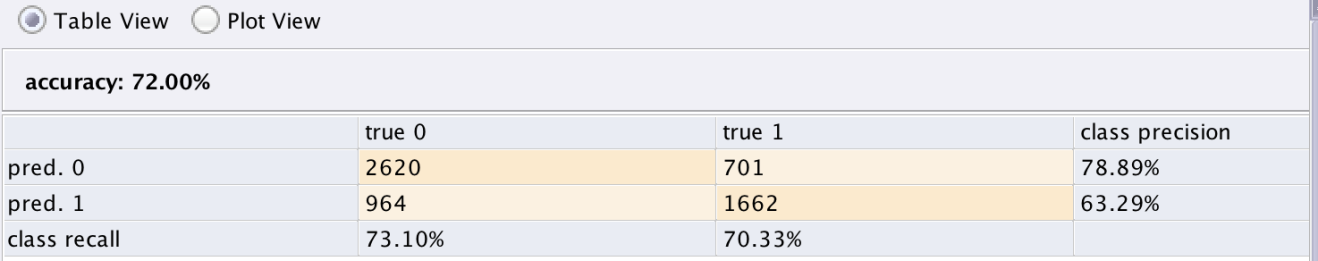
**8.8 Decision Tree model performance:** [**Back to the top**](#backdecision)

**8.8.1 Parameters:**

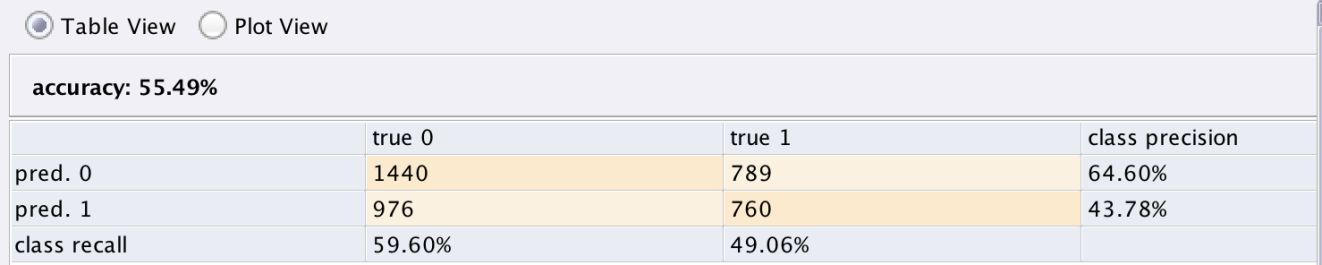
****

**8.8.2 All PCA values considered**

**Training accuracy**

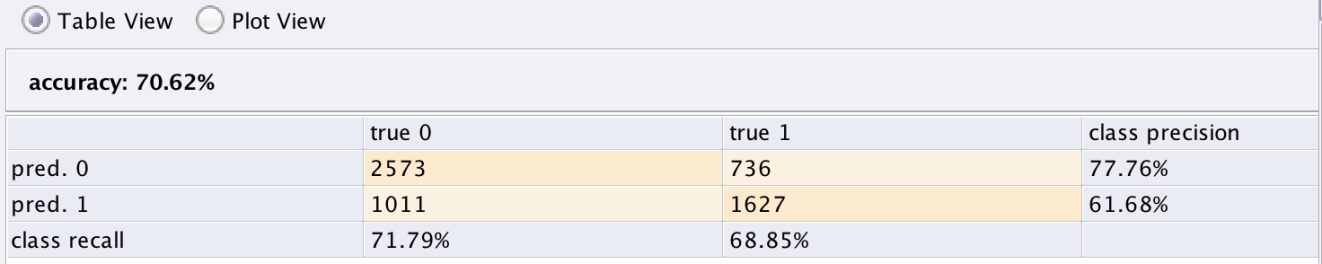


**Validation accuracy**

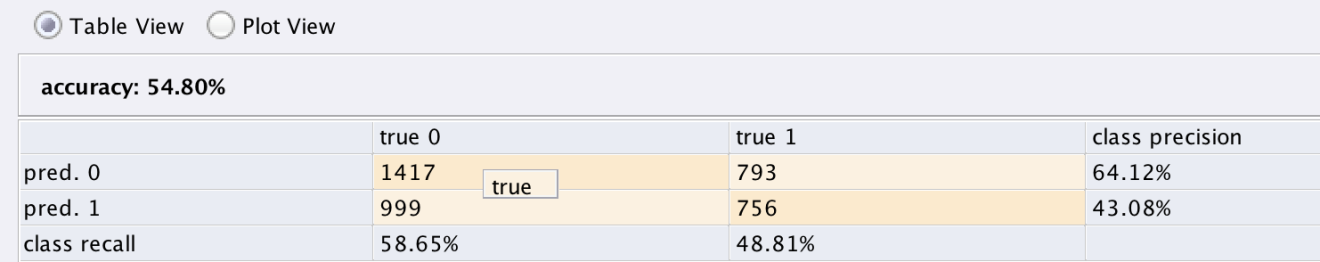


**8.8.3 Neighborhood PCA values removed:**

**Training accuracy**

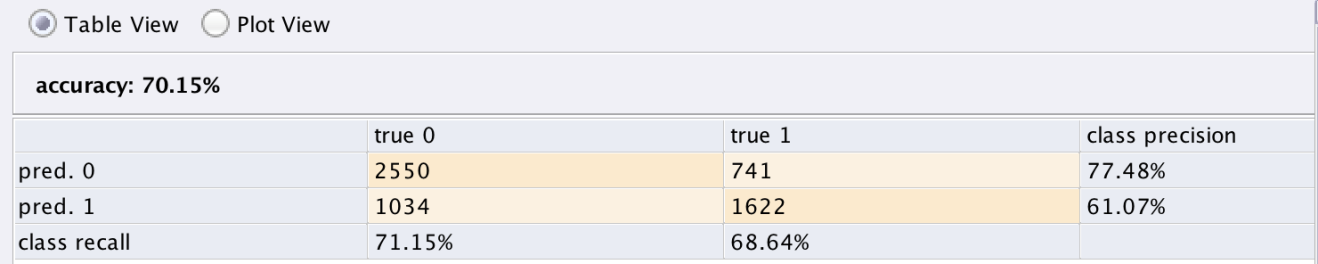


**Validation accuracy**

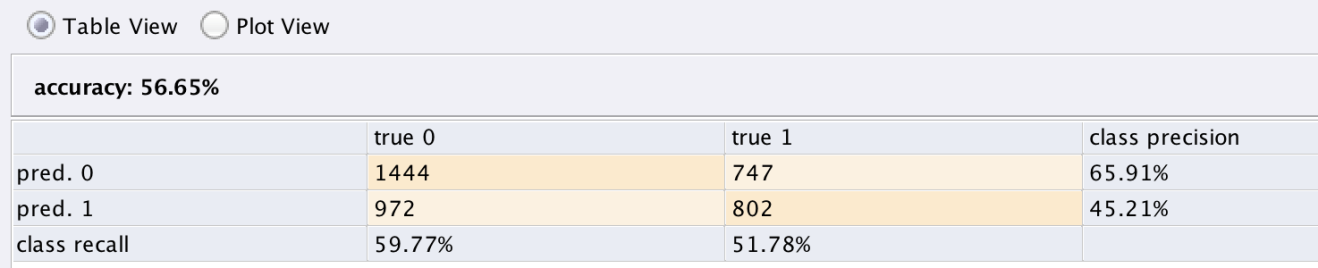


**8.8.4 Interest PCA values removed**

**Training accuracy**

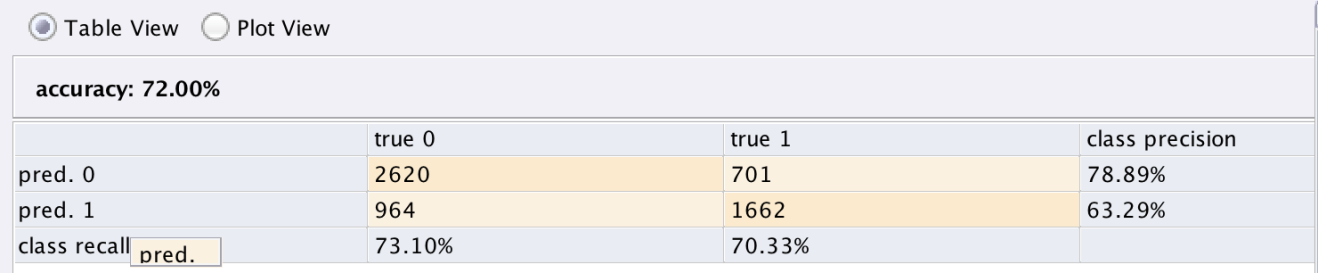


**Validation accuracy**

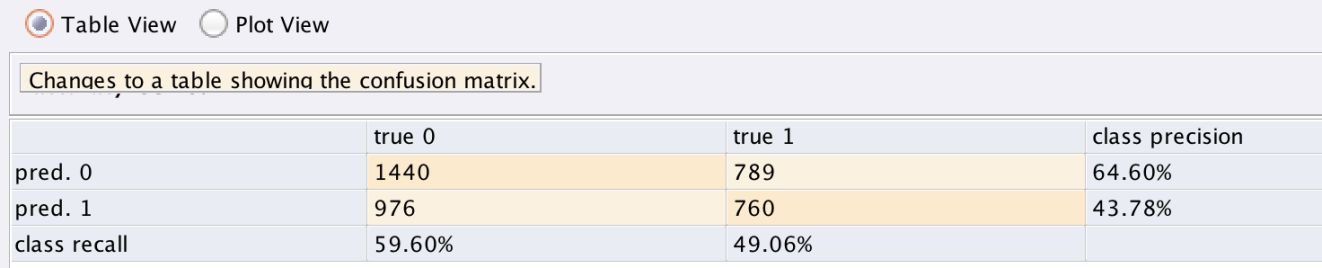


**8.8.5 Other type mail PCA values removed**

**Training accuracy**

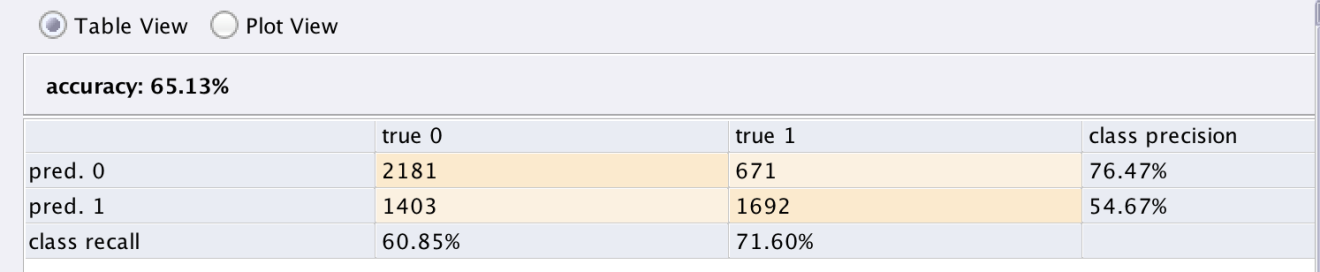


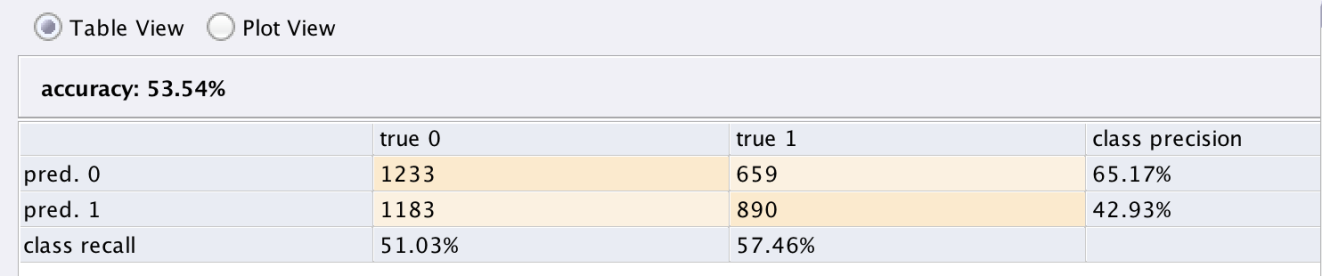
**Validation accuracy**



**8.8.6 Remove all PCA values of three categories**

**Training accuracy**

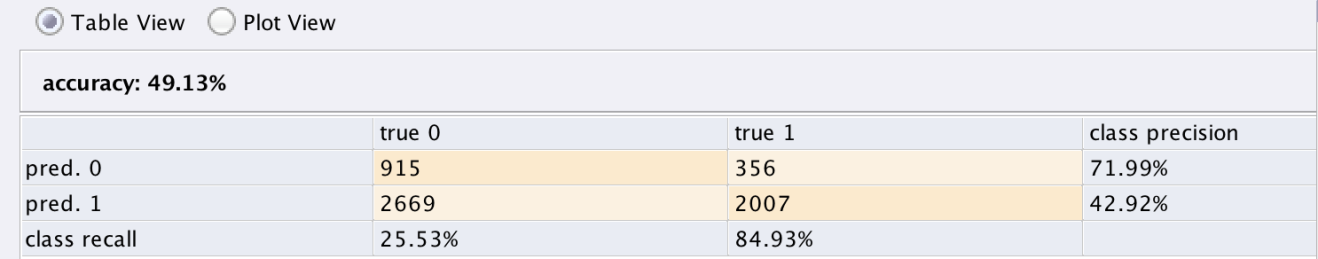


**Validation accuracy**

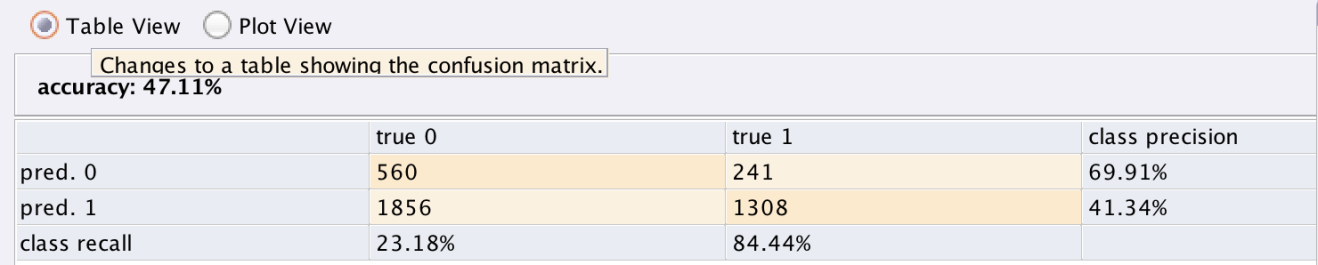
**8.9 W-Logistic Regression** [**Back to the top**](#backlogistic)

**8.9.1 All PCA values considered**

**Training accuracy**

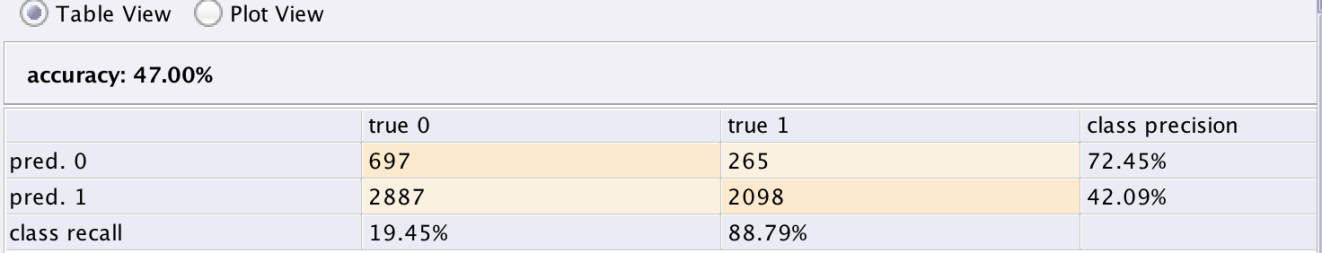


**Validation accuracy**

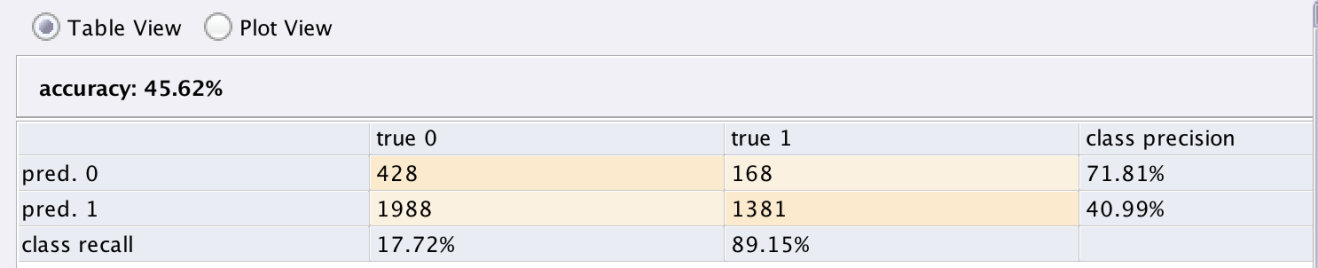


**8.9.2 Neighborhood PCA values removed**

**Training accuracy**

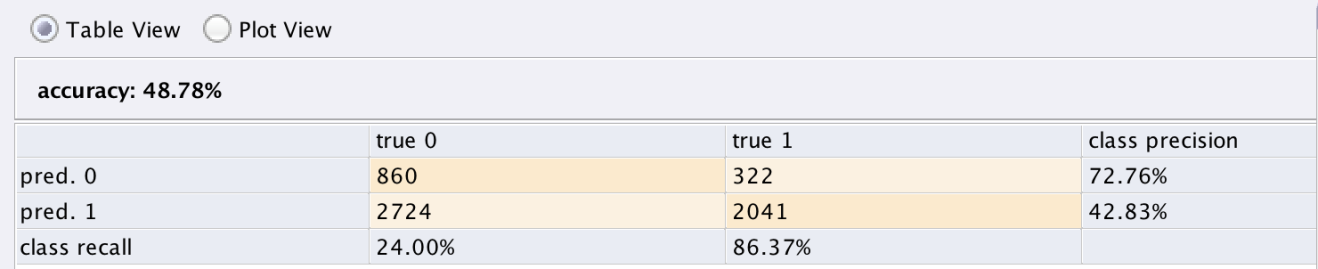


**Validation accuracy**

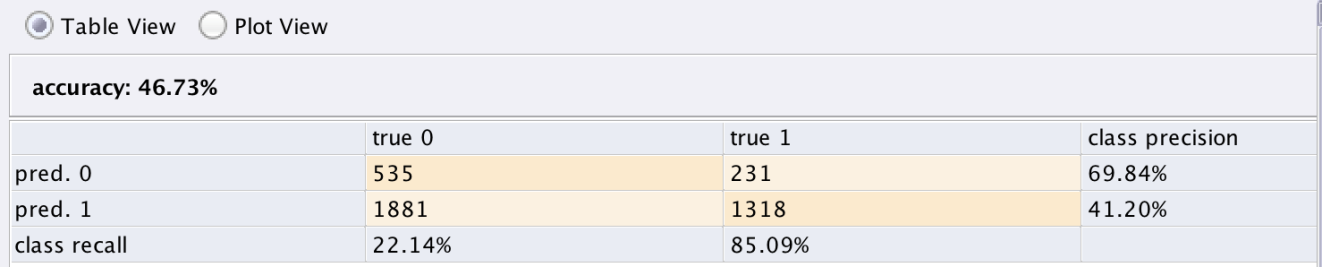


**8.9.3 Interest PCA values removed**

**Training accuracy**

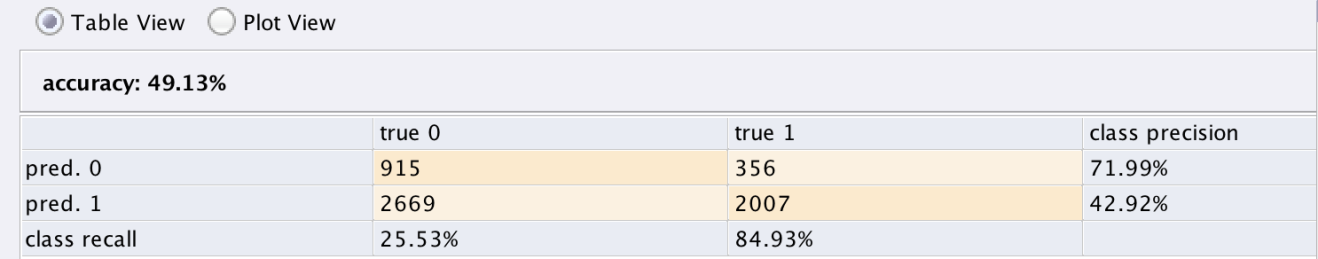


**Validation accuracy**

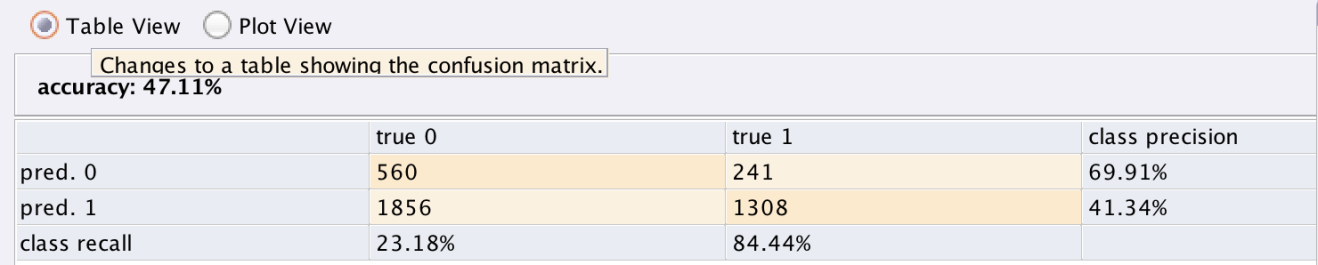


**8.9.4 Other mail type PCA values removed**

**Training accuracy**

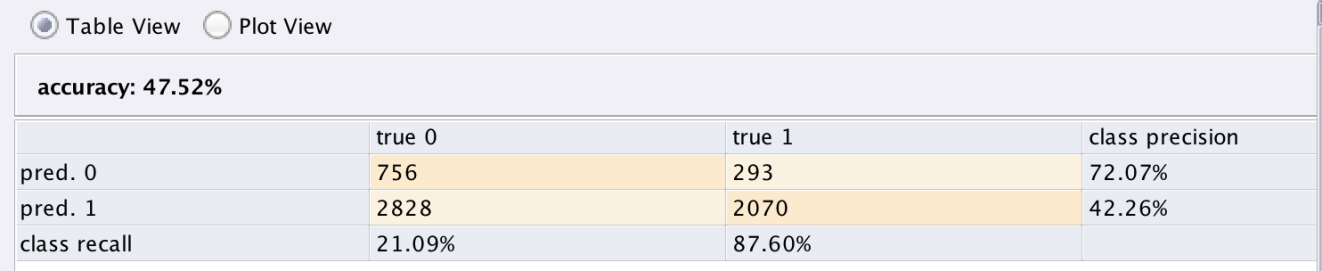


**Validation accuracy**

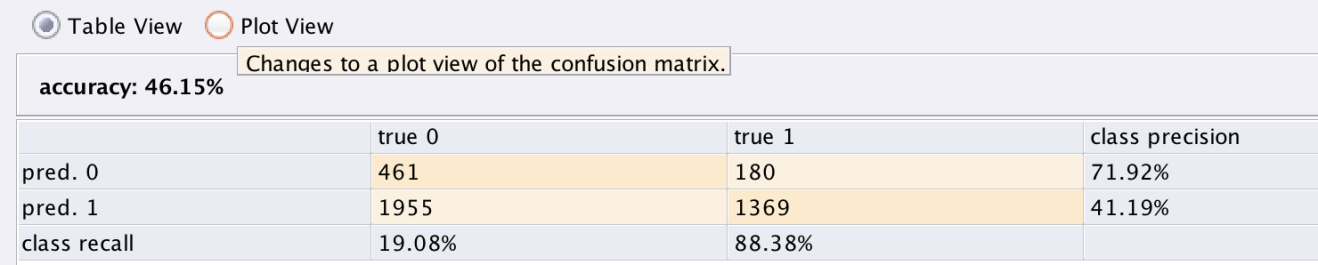


**8.9.5 Remove PCA values of all categories**

**Training accuracy**



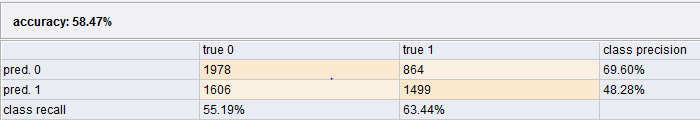
**Validation accuracy**



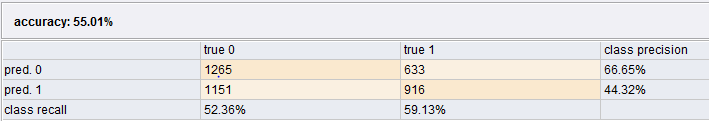
**8.10 Random forests model performance** [**Back to the top**](#backrandom)

**8.10.1 All PCA values considered**

**Training Accuracy:**

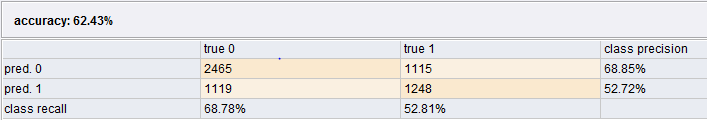


**Validation Accuracy:**

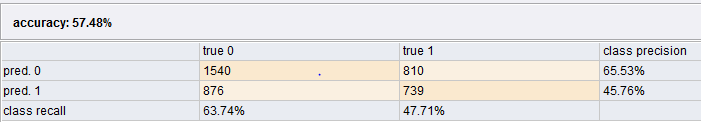


**8.10.2 Neighborhood PCA values removed**

**Training Accuracy:**

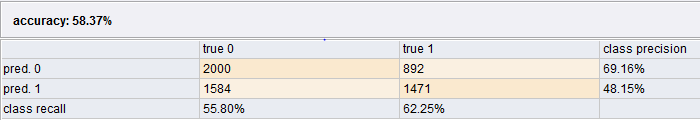


**Validation Accuracy:**

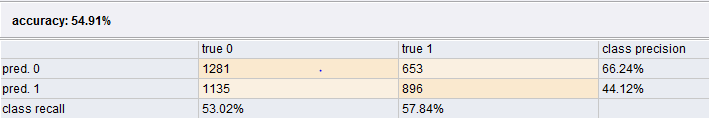


**8.10.3 Interests PCA values removed**

**Training Accuracy:**

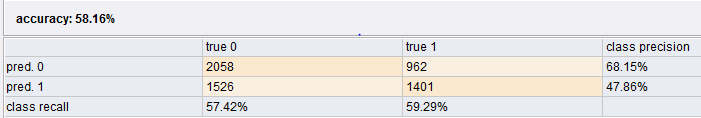


**Validation Accuracy:**

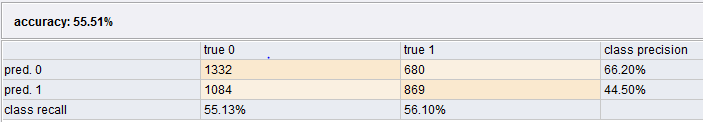


**8.10.4 Other type mail PCA values removed**

**Training Accuracy:**

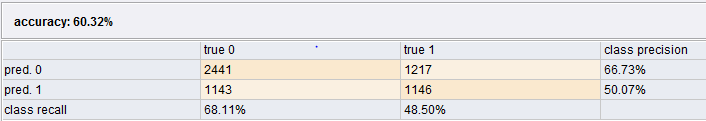


**Validation Accuracy:**

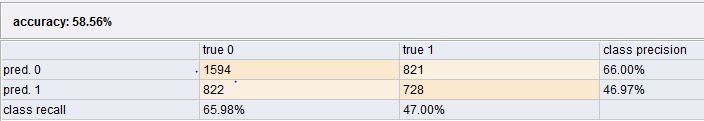


**8.10.5 Remove all PCA categories**

**Training Accuracy:**



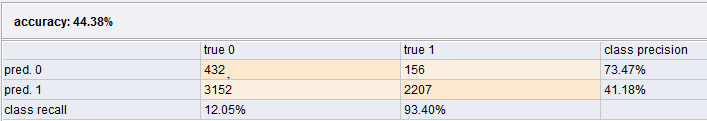
**Validation Accuracy:**



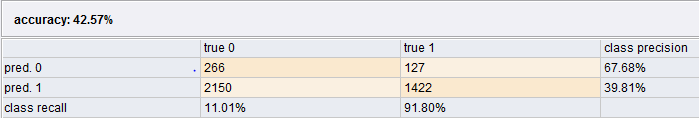
**8.11 Support Vector Machines model performance** [**Back to the top**](#backsvm)

**8.11.1 All PCA values considered**

**Training accuracy**

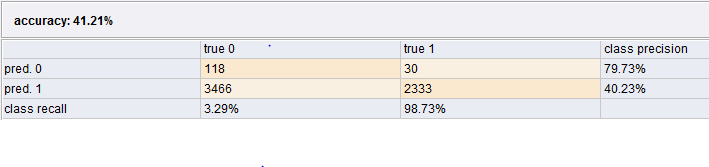


**Validation Accuracy:**

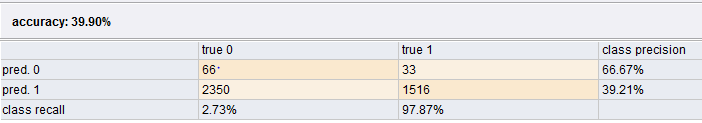


**8.11.2 Neighborhood PCA values removed**

**Training Accuracy**

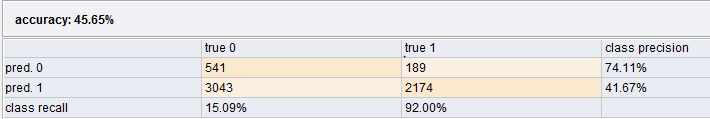


**Validation Accuracy**

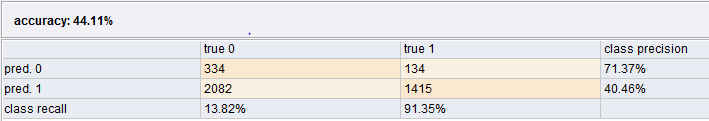


**8.11.3 Interests PCA values removed**

**Training accuracy**

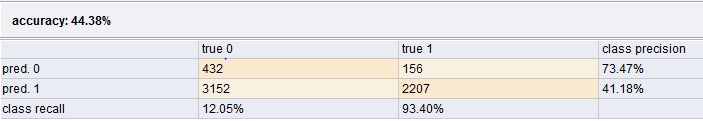


**Validation accuracy**

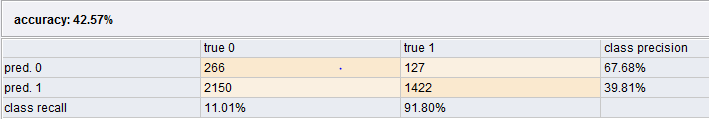


**8.11.4 Other type mail PCA values removed**

**Training accuracy**

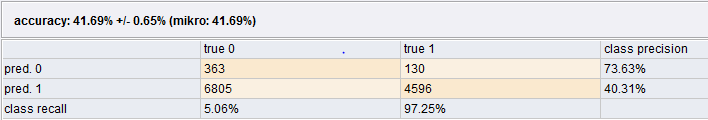


**Validation accuracy**



**8.11.5 Remove all PCA categories**

**Training accuracy**



**Validation accuracy**

