Data Mining

March 9, 2016

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Premnath Ramanathan |  | Sriram Santhanakrishnan |  | Vignesh Shankar |

[Praman3@uic.edu](mailto:Praman3@uic.edu) [Ssanth5@uic.edu](mailto:Ssanth5@uic.edu) [Vshank3@uic.edu](mailto:Vshank3@uic.edu)

Table of Contents

1. [Data Cleaning, Reduction and PCA 3](#_Toc445527932)
2. [Question 1 3](#_Toc445527933)
3. [Question 2 10](#_Toc445527934)
4. [Question 3 11](#_Toc445527935)
5. [Question 4 11](#_Toc445527936)
6. [Question 5 12](#_Toc445527937)
7. [Appendix 13](#_Toc445527938)

# Data Cleaning, Reduction and PCA

The final set of variables for this assignment was changed after deep exploration of data and to arrive at the best model. We carried out the same steps performed in the previous assignment to clean the data and reduce the number of variables. Also we have used different set of PCAs this time to get our best model. After exploring, cleaning, transforming and reducing the variables from the given dataset, there were 39 variables that seemed to be appropriate for modeling. Please refer the appendix section (Exhibit 1) of this document for the list of PCAs used and the variables contained in each PCA.

# Question 1

**Modeling - Consider the following classification techniques on the data: • decision Trees (you can use J48, or any other suitable type of decision tree) • logistic Regression • naïve-Bayes • k-nearest neighbors • random forests • support vector machines**

Our approach to find the best model was based on the selection of attributes and the performance results from training and validation by changing the parameters in each of the classification techniques. First we partitioned the dataset into 60% training and 40% validation. We also set the random seed value to 12345 as given in the problem to ensure that same results are obtained every time we run the process.

As the problem is dependent mainly on predicting the potential donors rather than the non-donors we set a threshold of 0.7 for both training and validation data last time. Which gave us an optimistic value for class precision for true 1’s (correct prediction of donors) while reducing the cost incurred for falsely predicted non-donors. But this time around, since finding the maximum profit and potential donors are the most significant measures, we have used the following operators to find the Maximum profit for training and validation data and did not use any threshold value.

1. Sort operator- To sort the records in decreasing order of the attribute -confidence (1).
2. Generate attributes- To calculate the profit and assign it to newly generated attribute.
   1. Profit was calculated using the below formula

if(TARGET\_B==1,1.7952,-0.9928)

1. Integrate – To generate the cumulative sum series.
2. Max Profit –To find the maximum profit using max statistics from the example set derived from the previous operator.

The basic structure of the profit validation process is available in Appendices section of this document (Please refer to Exhibit 2).

**What parameter values do you try for the different techniques, and what do you find to work best?**

The different classification techniques and the parameters used for those techniques are as follows.

1. **Decision Tree**

We compared the results between **W-J48** operator and the **Decision Tree** operator by trying out different parameter values for both the operators on the final set of (39) attributes.

We tried using different parameters for both the operators and the significance of those parameters are as given below

|  |  |  |
| --- | --- | --- |
| **Decision Tree Parameters** | **Best Model** | **Reasons/Significance** |
| Criterion | Information\_gain | Out of the four criterion, Information\_gain gave us the most optimized results |
| Max depth | 4 | Decreasing the depth to 4 gave us better results. A depth less than 4 and more than 15 produced drastic results. |
| Pruning confidence | 0.5 | Default value of 0.5 for confidence gave us the best model |
| Pre-pruning |  |  |
| Minimal gain | 0.01 | A minimal gain value from 0.1 to 0.001 gave us good results. But a value of 0.01 gave us the best model |
| Min leaf size | 2 | Leaf size of 1 gave us optimized output |
| Min size for split | 2 | A value ranging from 1 to 12 gave us varying results out of which a value of 2 gave us the best model. |
| No. of prepruning alts | 3 | This factor did not significantly change the accuracy of the end result. Hence, default value was used |

The performance results from this model for Testing data is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| Decision Tree - Validation | true 0 | true 1 | class precision |
| pred. 0 | 2060 | 966 | 68.08% |
| pred. 1 | 533 | 440 | 45.22% |
| class recall | 79.44% | 31.29% |  |

1. **Logistic Regression**

**We used W-Logistic** operator on our final set of variables and trained the model to provide the desired output. We derived our best model by using the following parameter values.

|  |  |  |
| --- | --- | --- |
| **W-Logistic Parameters** | **Best Model** | **Reasons/Significance** |
| D - Turn on Debugging output | Unchecked | Debugging is turned off by default. We used the default value because there was no significant difference otherwise |
| R - Ridge in the log-likelihood | 1.0E-8 | Changing the ridge value did not significantly change the performance. Default gave us the best model. |
| M - Maximum number of iterations | 7 | A value of 7 was used to arrive at the best model. Values above and below 7 changed the results slightly. |

The performance results from this model for Testing data is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| W-Logistic - Validation | true 0 | true 1 | class precision |
| pred. 0 | 2430 | 1315 | 64.89% |
| pred. 1 | 131 | 124 | 48.63% |
| class recall | 94.88% | 8.62% |  |
| Accuracy | 63.85% |  |  |

1. **Naïve Bayes classification**

We tried both Naïve Bayes and Naïve Bayes (Kernel) operators on our final set of attributes. Naïve Bayes operator gave us optimal results compared to Naïve Bayes (Kernel). Naïve Bayes produced the best model which could be because the method works better on categorical attributes. The above distinction could be due to Naïve Bayes being data dependent (Generative classifier).

The various parameters used and their significance are listed below.

|  |  |  |
| --- | --- | --- |
| **Naïve Bayes Parameters** | **Best Model** | **Reasons/Significance** |
| Laplace Correction | Checked | Laplace correction was turned on for smoothing. |
| Estimation mode-kernel density estimation | Greedy | Default mode greedy gave us the best model. Full estimation mode produces fixed bandwidth |
| bandwidth selection | None | Default value of Heuristic bandwidth selection gave us a decent model. Fixed bandwidth gave us an over fit model |
| Minimum bandwidth | 0.2 | Value of 0.2 was used to get the best model. Other values gave us decent models |
| Number of kernels | 54 | A value of 54 for number of kernels gave us a good model. Value above 70 and below 15 gave us over fit model. |
| application grid size | None | Change in grid size gave us poor results. |

The performance results using Naïve Bayes Kernel operator for Testing data is given below.

|  |  |  |  |
| --- | --- | --- | --- |
| Naïve Bayes Kernel- Validation | true 0 | true 1 | class precision |
| pred. 0 | 1552 | 723 | 68.22% |
| pred. 1 | 1009 | 716 | 41.51% |
| class recall | 60.60% | 49.76% |  |
| Accuracy | 56.70% |  |  |
| Max Profit | 354.35 |  |  |

1. **KNN:**

|  |  |  |
| --- | --- | --- |
| **KNN Parameter** | **Best Model** | **Reasons/Significance** |
| K – Number of nearest neighbors | 3 | Increasing the k value from 1 to 3 increased the performance by 75%. Performance degraded for any value more than that 5. So, 3 was our optimum k value. |
| Weighted Vote –Weight of Examples | Unchecked | Checking this value degraded the performance. The default option of unchecked gave us good results. |
| Measure Types –Type of measure to find the nearest neighbor | Mixed Measures | Default value was used as the input to the model has numerical attributes. The best value was obtained using the default value. |
| Mixed Measure | Mixed Euclidean Distance | Euclidean distance is the option available for mixed measure and numerical measures. |

The performance results using K-NN operator for Testing data is given below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | true 0 | true 1 | class precision |
| pred. 0 | 1649 | 815 | 66.92% |
| pred. 1 | 970 | 566 | 36.85% |
| class recall | 62.96% | 40.98% |  |

1. **Random Forest:**

**We used Random Forest operator** as it gave us better performance than W-Random Forest operator.

|  |  |  |
| --- | --- | --- |
| **Random Forest Parameters** | **Best Model** | **Reasons/Significance** |
| Number of trees | 1 | Number of trees more than 1 gave us significant change in outcome. Value of 1 gave us the best model |
| Criterian | Gini\_index | Out of the four criterion, Gini index gave us the most optimized results |
| Max Depth | 6 | Decreasing the depth to 6 gave us better results. A depth less than 4 and more than 15 produced drastic results. |
| Pruning confidence | 0.5 | Default value of 0.5 for confidence gave us the best model |
| Pre-pruning |  |  |
| Minimal gain | 0.001 | A minimal gain value from 0.1 to 0.001 gave us good results. But a value of 0.001 gave us the best model |
| Min leaf size | 40 | Leaf size of 1 gave us optimized output |
| Min size for split | 10 | A value ranging from 1 to 12 gave us varying results out of which a value of 10 gave us the best model. |
| No. of pre-pruning alts | 3 | This factor did not significantly change the accuracy of the end result. Hence, default value was used |
| Guess subset ratio | 0.92 | Values ranging from 0.6 to 0.9 gave us decent model but a value of 0.92 gave us the best model. Values below 0.6 gave us bad model. |

The performance results using Random Forest operator for Testing data is given below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | true 0 | true 1 | class precision |
| pred. 0 | 1922 | 867 | 68.91% |
| pred. 1 | 671 | 539 | 44.55% |
| class recall | 74.12% | 38.34% |  |

1. **SVM:**

|  |  |  |
| --- | --- | --- |
| **SVM Parameters** | **Best Model** | **Reasons/Significance** |
| C- Confidence threshold for pruning | 1 | Confidence value above and below 1 gave us poor results. A value of 1 gave us optimal result. |
| M- Minimum instances per leaf | 100,000 | Default value gave us the best model. |
| R – Use reduced error pruning | Checked | Option is checked to build a better model. |
| B – Binary Splits | Checked | Option is checked for binary split. |

The performance results using SVM operator for Testing data is given below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | true 0 | true 1 | class precision |
| pred. 0 | 1899 | 904 | 67.75% |
| pred. 1 | 694 | 502 | 41.97% |
| class recall | 73.24% | 35.70% |  |

**Provide a comparative evaluation of performance of your best models from each technique?**

The different subset of variables were chosen carefully by first analyzing the dataset and identifying variables that are significant and then packaging them into small subsets. A subset of the variables were chosen and run on each method to check their performance. We decided to take this subset of variables based on their significance from decision tree. The list of variables that were chosen as base subset and model applied on them are mentioned below.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Training** | | | | **Validation** | | | |
| **Class recall** | | **Class Precision** | | **Class recall** | | **Class Precision** | |
| True 0 | True 1 | True 0 | True 1 | True 0 | True 1 | True 0 | True 1 |
| Decision Tree | 85.64% | 45.27% | 74.50% | 62.80% | 79.44% | 31.29% | 68.08% | 45.22% |
| W-Logistics | 95.73% | 8.84% | 66.23% | 52.56% | 94.91% | 9.67% | 65.96% | 50.75% |
| KNN | 88.74% | 32.53% | 70.68% | 61.19% | 74.68% | 29.40% | 66.73% | 37.98% |
| Random Forest | 77.99% | 40.36% | 71.04% | 49.71% | 74.66% | 37.77% | 68.87% | 44.70% |
| SVM | 74.81% | 36.52% | 68.76% | 43.71% | 73.24% | 35.70% | 67.75% | 41.97% |
| Naïve Bayes | 65.63% | 64.10% | 77.35% | 49.96% | 59.35% | 51.18% | 69.20% | 40.25% |

**Does variable selection/PCA make a difference for the different models?**

We tried excluding combination of PCA’s and PCA’s individually and checked if it affects the performance significantly. The significance of PCA’s across the six techniques are given below.

**Decision Tree**:

We tried excluding different combinations of subset of PCAs and evaluated the model by comparing the results from our base model which includes all the PCAs. We found out that by

Excluding Neighborhood Population and Military PCAs gave us the best model based on performance of recall. Please refer Appendix section of the document for performance metrics in the attached excel sheet.

**W-J48**:

We tried excluding different combinations of subset off PCAs and evaluated the model comparing the results from our base model which includes all the PCAs. We identified that by excluding Gift History PCA increased the performance of recall and precision but it was not better than the model obtained from our base set of attributes.

**W-Logistic**:

We first tried to convert the numerical attributes to Binary attributes as Logistic Regression produces best results for categorical variables. However, Logistic regression requires large number of data to get a good model. A small dataset with less number of events will result in biased outcome. In spite of trying out different PCAs the output was biased towards 0s as the dataset contains more number of records in the same category. Hence, there was no significant improvements in the results compared to the best model obtained.

**Naïve Bayes**:

Excluding subset of PCA’s or individual PCAs did not improve the performance when compared to the base model. The performance reduced significantly without including PCAs. Highest performance was obtained by excluding PCA1- Military & PCA3-Neighbourhood Population but it was not better than the model obtained from our base set of attributes.

**KNN**:

Excluding Gift History PCA and Neighborhood Population PCAs gave us a decent model but when compared to base model the performance was degraded. The base model with all the three PCAs gave us the best model using KNN. KNN produced erratic results with and without PCAs.

**SVM**:

Excluding Neighborhood Population PCA gave us a good model but it was not better than the base model which consists of all the PCAs. The results of the performance are given in the Appendix section of this document in the attached excel sheet

**Random Forest**:

We tried excluding different combinations of subset of PCAs and evaluated the model by comparing the results from our base model which includes all the PCAs. We found out that by

Excluding Neighborhood Population and Gift History PCAs gave us the best model based on performance of recall. Please refer the attached excel sheet in Appendix section of the document.

# Question 2

Calculate Net Profit: For each method in Question 1 (choose the ‘best’ model for each method/technique, either with the full or reduced set of variables), calculate the lift of net profit for both the training and validation set based on the actual response rate (5.1%). Again, the expected donation, given that they are donors, is $13.00, and the total cost of each mailing is $0.68. (Hint: to calculate estimated net profit, we will need to ”undo” the effects of the weighted sampling, and calculate the net profit that would reflect the actual response distribution of 5.1% donors and 94.9% non-donors.)

The original dataset consisted of 5.1% of donors and 94.9% of non-donors. To calculate the Net Profit for our best model we have used 40% response to avoid random sampling because using same ratio in the sample would lead to biased output. We have calculated the Maximum Cumulative Profit and Net Profit for training and validation data for the best models from each technique. Maximum Cumulative Profit and Net Profit were calculated using the below formula.

**Actual Profit per donor** = $13 – $0.68 =$12.32

**Actual Cost** = -$0.68

To undo the weighting effects we performed the following calculations:

**Weighted Profit** = (12.32\* 0.051)/0.35 =$1.7952

**Weighted Cost** = (0.68\*0.949)/0.65 = $0.9928

The Maximum Cumulative Profit was calculated using these adjusted values. We have not included **opportunity cost** to calculate net profit.

The comparison between different models is given below.

|  |  |  |
| --- | --- | --- |
|  | Testing | |
| Model | Max Cumulative Profit | Net Profit |
| Decision Tree | 410.95 | 310.16 |
| Logistic Regression | 377.9 | 113.0976 |
| Naïve Bayes | 274.45 | 242.5424 |
| K-NN | 477.03 | 45.2472 |
| Random Forest | 462.81 | 303.8784 |
| SVM | 267.56 | 212.1872 |

Based on the above table it can be inferred that the maximum lift was obtained from Decision Tree model for the validation data.

# Question 3

Draw Lift Curves: Draw each model’s net profit lift curve for the validation set onto a single graph. Are there any models that dominate?

Net Profit lift curves for the validation set were plotted for the best classification models from each technique. The lift graph is given below. From the graph it can be inferred that the Decision Tree model dominates the lift curve with maximum cumulative profit of **410.95.**

# Question 4

Best Model: From your answers above, what do you think is the “best” model? (What criteria do you use to determine ‘best’?) Summarize the performance of the ‘best’ model from each method, in terms of Net Profit from predicting donors in the validation dataset; at what cutoff is the best performance obtained?

We used Maximum cumulative Profit criteria to choose our best model. The best model for each technique is given below along with their cut-off value.

|  |  |  |
| --- | --- | --- |
| **Model** | **Max Cumulative Profit** | **Cut-off level** |
| Decision Tree | 410.95 | 0.4 |
| Logistic Regression | 377.90 | 0.33 |
| Naïve Bayes | 274.45 | 0.41 |
| K-NN | 82.6608 | 0.45 |
| Random Forest | 355.7352 | 0.3 |
| SVM | 267.56 | 0.3 |

From the above table it can be inferred that ‘Decision Tree’ is our best model as the maximum cumulative profit is highest for Decision Tree model.

The cut-off value for our best model was **0.4**. The maximum profit was obtained at ‘1215’ data point. Hence our best model targets **30.4%** (1215/4000 = 0.30375) of individuals from the unseen data.

# Question 5

Testing -The file FutureFundraising.xls contains the attributes for future mailing candidates. Using your “best” model from Step 2 (#5), which of these candidates do you predict as donors and non-donors? List them in descending order of probability of being a donor. What cutoff do you use to predict donor/non-donor? Submit this file (xls format), with your best model’s predictions (prob of being a donor).

We used our best model identified from step 2 to predict future mailing candidates from unseen data- FutureFundraising.xls

Our best classification model predicted **5481** donors out of **20,000** mailing candidates. We used a cutoff/Threshold value of **0.4** to predict donors. Hence our best model would target **27.4%** (5481/20,000 = 0.2741) of individuals from unseen data.

Please refer to the attached excel file for the results.

# Appendix



Exhibit 1

|  |  |  |
| --- | --- | --- |
| PCA-1 | PCA-2 | PCA-3 |
| **Military & Veterans** | **Gift History** | **Neighborhood Population** |
| FEDGOV | AVGGIFT | POP901 |
| LOCALGOV | CARDGIFT | POP902 |
| MALEMILI | LASTGIFT | POP903 |
| MALEVET | MAXRAMNT | POP90C1 |
| STATEGOV | MINRAMNT | POP90C2 |
| VIETVETS | NGIFTALL | POP90C3 |
| WWIIVETS | RAMNTALL | POP90C4 |
|  |  | POP90C5 |

Exhibit 2

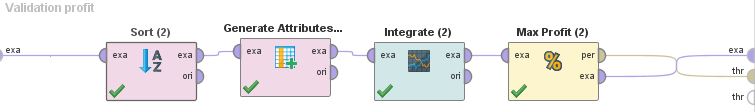


Exhibit 3

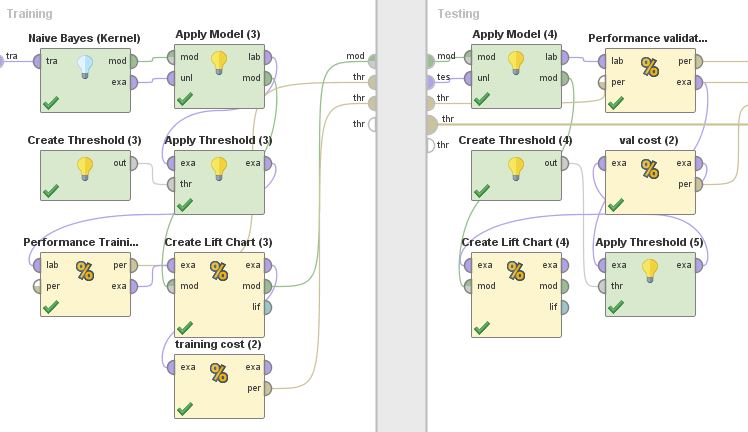
****

Exhibit 4



Exhibit 5



Exhibit 6

