Target Marketing - Fundraising

IDS 572: DATA MINING ASSIGNMENT-3

Ashuthosh Gowda
ASHUTHOSH GOWDA

LAAVANYA GANESH

AKSHAY MERCHANT

Assignment 3

Q.1. SOLUTION:

After partitioning the data into 60% Train data and 40% Validation data, and setting the seed to 12345, we developed the SVM (Support Vector Machine) Model for classification. We changed different parameters and got the best model in SVM for the following parameters.

Support Vector Machine

As we increase the L-pos in the range 1-1.5 along with an increase in L-neg in the range 0.8-0.9, we see the accuracy, recall and precision on the models fluctuate with a very small deviation. Hence, after trying different values for L-pos and L-neg in the above ranges, we chose the model which gave a higher accuracy, recall and precision. When we increase the Kernel gamma and cache, the models were tending towards overfit. So, we decided to keep the them constant at 0.005 and 200.

Shown below is the confusion matrix for the best SVM model

Confusion Matrix for Train Data

Accuracy=73%	True.0	True.1	**Class Precision**
Pred.0	2494	175	93.44%
Pred.1	1445	1885	**56.61%**
Class Recall	63.32%	**91.5%**	

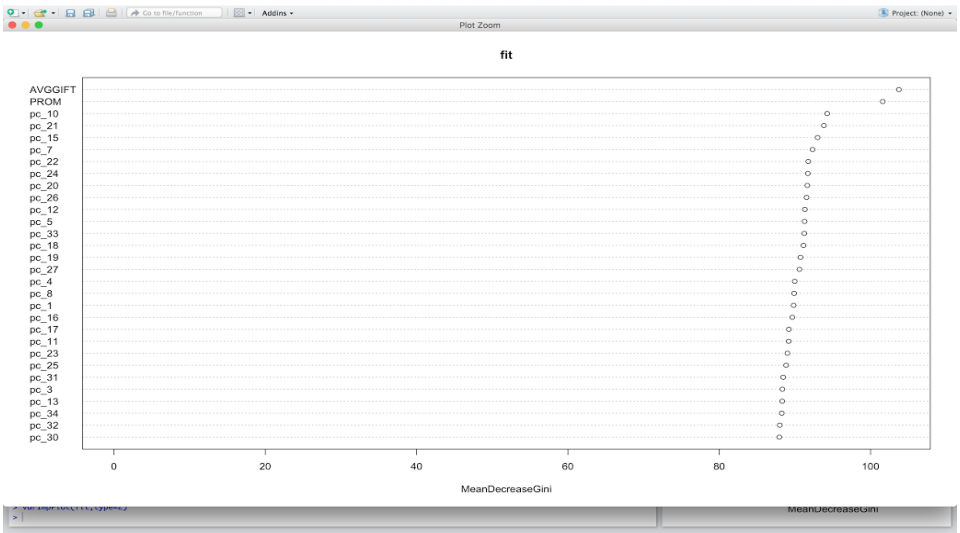
Confusion Matrix for Test Data

Accuracy=50.75%	True.0	True.1	**Class Precision**
Pred.0	959	368	72.27%
Pred.1	1602	1071	**40.07%**
Class Recall	37.45%	**74.43%**	

Variable Selection

We now had 38 RELEVANT transformed attributes after cleaning the original dataset plus the 40 PCA Transformed Variables, which gave us a total of 78 attributes on which we ran the Variable Importance R Code. We have used the “importance()” function in Random Forest, to get list of Variables arranged in the order of least Important to Most Important.

Below is the Variable Plot for Mean Decrease By Gini.



After getting the list of Variables in Order of least to most important, we used the Backward Selection method to get the best variables, that we used as the dataset in all our models.

FINAL ATTRIBUTE SET USED FOR MODELLING					
AGE	MALEMILI	RFA13left	pc_1	pc_12	pc_23
AVGGIFT	NEW.TCODE	RFA13mid	pc_2	pc_13	pc_24
CARDPM12	NUMPRM12	RFA3left	pc_3	pc_14	pc_25
CLUSTER	PEPSTRFL	RFA3mid	pc_4	pc_15	pc_26
DOM1	PROM1	RFA5left	pc_5	pc_16	pc_28
DOM2	PROM2	RFA5mid	pc_6	pc_17	pc_29
GEOCODE2	RECINHSE	RFA5right	pc_7	pc_18	pc_30
GOV1	RECPGVG	SOLIH	pc_8	pc_19	pc_31
GOV2	RECSWEEP	VET1	pc_9	pc_20	pc_32
HPHONE_D	RESPONSEtoOFFERS	VET2	pc_10	pc_21	pc_33
INCOME	RFA_2A	WEALTH2	pc_11	pc_22	pc_34
MINRAMNT	MAXRAMNT				

We used all the Classification models on the Variable Selection Dataset (Dataset with variables mentioned above) and the Complete Dataset. We found that using Variable Selection did make a difference in the model. As shown in the table below, when the Variable Selection Dataset s used on the models, we generally get a better Precision and Accuracy. This is why we preferred the Variable Selection Dataset over the complete one.

Classification Model	Variable Selection Dataset			Complete Dataset		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Decision Trees	58.3	42.31	43.78	59.48	40.22	25.99
Boosted Trees	57.07	42.32	62.61	53.52	41.21	68.45
Naïve Bayes	58.5	43.77	54.00	51.85	39.41	62.96
Logistic Regression	54.7	42.11	69.21	55.12	41.98	64.77
Random Forest	57.07	46.82	33.12	64.84	49.72	8.45
Support Vector Machine	50.75	40.07	74.43	49.98	39.47	73.18

For *Decision Trees*: The Classification Model gives a better Precision and Recall, but a lower Accuracy for the Variable Selection Dataset when compared to the Complete Dataset.

For *Boosted Trees*: The Classification Model gives a better Accuracy and Precision, but a lower Class Recall for the Variable Selection Dataset when compared to the Complete Dataset.

For *Naïve Bayes*: The Classification Model gives a better Accuracy and Precision, but a lower Class Recall for the Variable Selection Dataset when compared to the Complete Dataset.

For *Logistic Regression*: The Classification Model gives a lower Accuracy, but a higher Precision and significantly higher Class Recall for the Variable Selection Dataset when compared to the Complete Dataset.

For *Random Forest*: The Classification Model gives a lower Accuracy and Precision, but a higher Class Recall for the Variable Selection Dataset when compared to the Complete Dataset.

For *Support Vector Machine*: The Classification Model gives a lower Accuracy, Precision and Recall for the Variable Selection Dataset when compared to the Complete Dataset.

We changed the threshold for all the classification using Create Threshold in Rapid Miner and we got the following models for each.

Classification Models

Decision Tree

The parameters used are:

Criteria	Max Depth	Confidence	Min Gain	Min Leaf Size	Min Size of Split	Pre-pruning Alternatives	Threshold
Gini Index	35	0.5	0.0015	6	40	3	0.3

The Confusion Matrix below is for the Train Data

Accuracy=76.21%	True.0	True.1	**Class Precision**
Pred.0	3097	585	84.11%
Pred.1	842	1475	**63.66%**
Class Recall	78.62%	**71.60%**	

The Confusion Matrix below is for the Validation Data

Accuracy=58.3%	True.0	True.1	**Class Precision**
Pred.0	1702	809	67.78%
Pred.1	859	630	**42.31%**
Class Recall	66.46%	**43.78%**	

Naïve Bayes

The parameters used are:

Laplace Correction	Threshold
No	0.413

The Confusion Matrix below is for the Train Data

Accuracy=60.99%	True.0	True.1	**Class Precision**
Pred.0	2525	926	73.17%
Pred.1	1414	1134	**44.51%**
Class Recall	64.10%	**55.05%**	

The Confusion Matrix below is for the Validation Data

Accuracy=58.5%	True.0	True.1	**Class Precision**
Pred.0	1563	662	70.25%
Pred.1	998	777	**43.77%**
Class Recall	61.03%	**54.00%**	

Boosted Trees

The parameters used are:

No of Trees	Max Depth	Min Rows	Min Split Improvement	No of Bins	Learning Rate	Sample Rate	Threshold
40	3	10	0	20	0.1	1	0.323

The Confusion Matrix below is for the Train Data

Accuracy=64.41%	True.0	True.1	**Class Precision**
Pred.0	2374	570	80.64%
Pred.1	1565	1490	**48.77%**
Class Recall	60.27%	**72.3%**	

The Confusion Matrix below is for the Validation Data

Accuracy=57.07%	True.0	True.1	**Class Precision**
Pred.0	1382	538	71.98%
Pred.1	1179	901	**42.32%**
Class Recall	53.96%	**62.61%**	

Logistic Regression

The parameters used are:

Solver	Regularization	Lambda	Lambda Search	No of Lambdas	Lambda Min Ratio	Early Stopping	Stopping Rounds
Auto	Yes	0.005	Yes	5	0.01	Yes	3

Stopping Tolerance	Alpha	Standardize	Non Negative Coefficients	P Value	Remove Collinear Columns	Add Intercept	Threshold
0.001	0.1	Yes	No	Yes	Yes	MeanImpute	0.3

The Confusion Matrix below is for the Train Data

Accuracy=56.69%	True.0	True.1	**Class Precision**
Pred.0	1925	584	76.72%
Pred.1	2014	1476	**42.29%**
Class Recall	48.87%	**71.65%**	

The Confusion Matrix below is for the Validation Data

Accuracy=54.7%	True.0	True.1	**Class Precision**
Pred.0	1192	443	72.91%
Pred.1	1369	996	**42.11%**
Class Recall	46.54%	**69.21%**	

Random Forest

The parameters used are:

No of Trees	Max Depth	Cutoff
500	8	0.43

The Confusion Matrix below is for the Train Data

Accuracy=100%	True.0	True.1	**Class Precision**
Pred.0	3926	0	100%
Pred.1	0	2074	**100%**
Class Recall	100%	**100%**	

The Confusion Matrix below is for the Validation Data

Accuracy=57.07%	True.0	True.1	**Class Precision**
Pred.0	2038	953	68.14%
Pred.1	536	472	**46.82%**
Class Recall	79.17%	**33.12%**	

Support Vector Machine

The parameters used are:

Kernel Type	Kernel Gamma	Kernel Cache	C	Convergence Epsilon	Max Iterations
Radial	0.005	200	1	0.001	1000000

L positive	L negative	Epsilon	Epsilon	Threshold
1.4	0.85	0	0.8	0.2798

The Confusion Matrix below is for the Train Data

Accuracy=73%	True.0	True.1	**Class Precision**
Pred.0	2494	175	93.44%
Pred.1	1445	1885	**56.61%**
Class Recall	63.32%	**91.5%**	

The Confusion Matrix below is for the Validation Data

Accuracy=50.75%	True.0	True.1	**Class Precision**
Pred.0	959	368	72.27%
Pred.1	1602	1071	**40.07%**
Class Recall	37.45%	**74.43%**	

Comparative Evaluation of Performance

Classification Model	Accuracy	Class Precision	Class Recall
Decision Trees	58.3	42.31	43.78
Boosted Trees	57.07	42.32	62.61
Naïve Bayes	58.5	43.77	54.00
Logistic Regression	54.7	42.11	69.21
Random Forest	57.07	46.82	33.12
Support Vector Machine	50.75	40.07	74.43

From the above table, we feel that Boosted Trees and Logistic Regression give us the best models. But since the Boosted Trees model has a higher Accuracy and Class Precision, we prefer it over the Logistic Regression Model.

2.1(a)

SOLUTION:
The dataset that we received was oversampled to get 35% Donors and 65% Non-Donors. But the Original Data recorded had the actual response distribution of 5.1% donors and 94.9% non-Donors. We can calculate the Net Profit, using the Information given:
Expected Donation, given that they are Donors: $13.00
Total Cost of each mailing: $0.68

Since, this information provided was for the Original data, to use this information on our Oversampled data, we need to Calculate Weighted Donation and Cost of Each Mailing.

Case 1: The Total Profit obtained, if our Mailing get a Donation: $13.00-$0.68= $12.32
Modified Donation to use in our Cost Matrix: $12.32 * 5.1%/35%= $1.7952

Case 2: The Total Cost of Mailing, if we do not get a response: $0.68
Modified Total Cost of Mailing for Cost Matrix: $0.68 * 94.9%/65%= $0.9952

The Best Model for Each Method were evaluated on the highest Net Profit generated.
DECISION TREE
Parameters:

Criteria	Max Depth	Confidence	Min Gain	Min Leaf Size	Min Size of Split	Threshold
gini_index	35	0.5	0.0015	6	40	0.3

Train Data Performance Matrix				
NET PROFIT	76.21	true 0	true 1	class precision
$ 1811.982	pred. 0	3097	585	84.11%
	pred. 1	842	1475	63.66%
	class recall	78.62%	71.60%	

Validation Data Performance Matrix				
NET PROFIT	58.3	true 0	true 1	class precision
$ 278.1608	pred. 0	1702	809	67.78%
	pred. 1	859	630	42.31%
	class recall	66.46%	43.78%	

GRADIENT BOOSTED TREE
Parameters:

No of trees	Max Depth	Min rows	Min split Improv.	No. bins	Learn Rate	Sample rate	Threshold
20	5	10	0	20	0.1	1	0.323

Train Data Performance Matrix				
NET PROFIT	64.41	true 0	true 1	class precision
$ 1121.116	pred. 0	2374	570	80.64%
	pred. 1	1565	1490	48.77%
	class recall	60.27%	72.33%	

Validation Data Performance Matrix				
NET PROFIT	57.07	true 0	true 1	class precision
$ 446.964	pred. 0	1382	538	71.98%
	pred. 1	1179	901	43.32%
	class recall	53.96%	62.61%	

NAÏVE BAYES

Parameters:

Threshold				
				0.413
Train Data Performance Matrix				
NET PROFIT	60.99	true 0	true 1	class precision
631.9376	pred. 0	2525	926	73.17%
	pred. 1	1414	1134	44.51%
	class recall	64.10%	55.05%	
Validation Data Performance Matrix				
NET PROFIT	58.5	true 0	true 1	class precision
404.056	pred. 0	1563	662	70.25%
	pred. 1	998	777	43.77%
	class recall	61.03%	54.00%	

LOGISTIC REGRESSION

Parameters:

Solver	Regularization	Lambda	Lambda search	No of lambdas	Lambda min ratio	Early stopping	Threshold
Auto	Y	0.0005	Y	5	0.01	Y	0.3
Train Data Performance Matrix							
NET PROFIT	56.69	true 0	true 1	class precision			
$ 650.3636	pred. 0	1925	584	76.72%			
	pred. 1	2014	1476	42.29%			
	class recall	48.87%	71.65%				
Validation Data Performance Matrix							
NET PROFIT	54.7	true 0	true 1	class precision			
$ 428.9756	pred. 0	1192	443	72.91%			
	pred. 1	1369	996	42.11%			
	class recall	46.54%	69.21%				

RANDOM FOREST

Parameters:

Number of Trees	Interaction Depth	Threshold
500	8	0.422

Train Data Performance Matrix				
NET PROFIT	80.31%	true 0	true 1	class precision
$ 650.3636	pred. 0	1925	584	76.72%
	pred. 1	2014	1476	42.29%
	class recall	48.87%	71.65%	
Validation Data Performance Matrix				
NET PROFIT	62.77%	true 0	true 1	class precision
$ 315.1936	pred. 0	2038	953	31.86%
	pred. 1	536	472	46.83%
	class recall	20.82%	33.12%	

SVM

Parameters:

kernel type	kernel gamma	kernel cache	C	Lpos	Lneg	episilon	Threshold
Radial	0.005	200	1	1.5	0.8	0	0.264

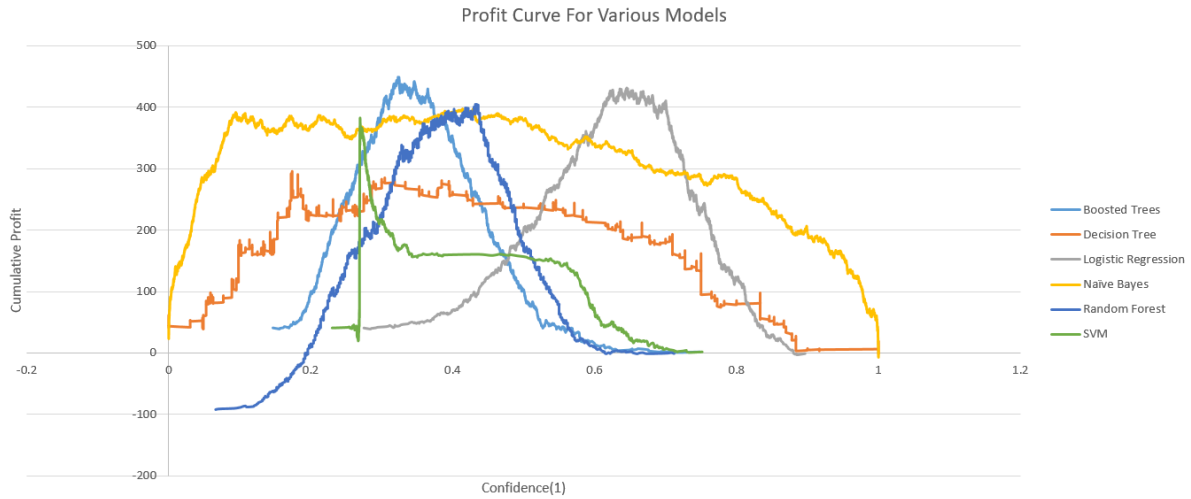
Train Data Performance Matrix				
NET PROFIT	76.43%	true 0	true 1	class precision
$ 2167.514	pred. 0	2683	158	94.44%
	pred. 1	1256	1902	60.23%
	class recall	68.11%	92.33%	

Validation Data Performance Matrix				
NET PROFIT	49.98%	true 0	true 1	class precision
$ 286.9736	pred. 0	946	386	71.02%
	pred. 1	1615	1053	39.47%
	class recall	36.94%	73.18%	

Summary of all the Best Models based on Net Profit in Validation Dataset.

MODEL TECHNIQUE	THRESHOLD	PRECISION	RECALL	NET PROFIT
DECISION TREE	0.3	42.31%	43.78%	$ 278.16
BOOSTED TREES	0.323	43.32%	62.61%	$ 446.96
LOGISTIC REG	0.3	42.11%	69.21%	$ 428.97
NAÏVE BAYES	0.413	43.77%	54.00%	$ 404.05
RANDOM FOREST	0.422	46.83%	33.12%	$ 315.19
SVM	0.264	39.47%	73.18%	$ 286.97

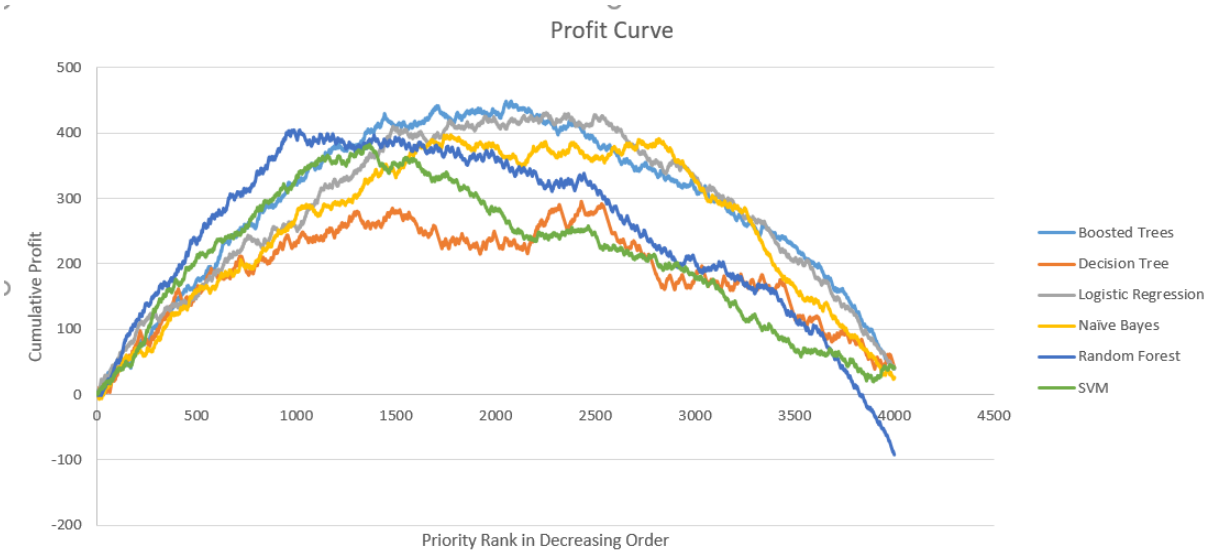
Find Below, the Profit Curve: Cumulative Profit vs Confidence for various models



Comparing the profit Curves of Different Techniques, we observe Different spread of Confidence(1)'s giving Cumulative profits.

Since, a Profit Curve must not have any sudden spikes in Cumulative Profit and must represent a smooth Single Maxima of Profit to calculate the corresponding Threshold, Logistic Regression and Boosted Trees give the most clear Picture of recognising a Threshold.

Shown below is the Profit Curve which compares Cumulative Profit vs Rank in Decreasing Order



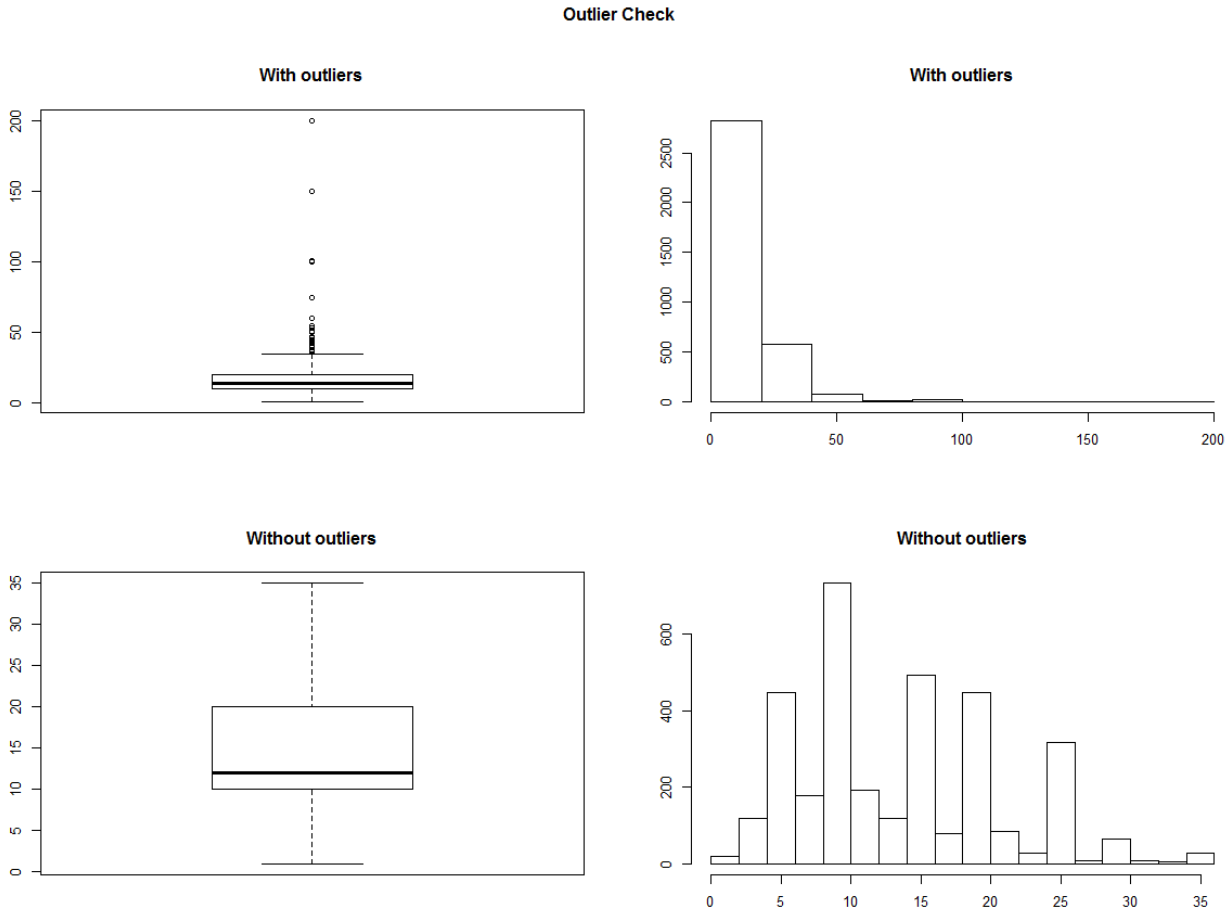
Q.2.2(a)

SOLUTION:

We have filtered out the Dataset by using rows with TARGET_B=1. From the 3500 rows, we checked for Outliers, i.e cases with rare and very large donation amounts. Based on the boxplots and histograms below, we saw that when there are very large donation amounts, the data is highly right skewed and has high side outliers. After removing these large amount donations, the boxplots and histograms illustrates a significant change and depict that large donation amounts should be removed.

R Output after running the Outlier Detection

```
Outliers identified: 134
Propotion (%) of outliers: 4
Mean of the outliers: 57.06
Mean without removing outliers: 15.58
Mean if we remove outliers: 13.92
```



We built a Linear Regression Model with 104 Variables for 3366 rows (refer the Table of Co-efficients obtained from the lm() function using R). We used the Step AIC function on the following 104 attributes which uses both forward and backward selection for Variable selection

ATTRIBUTES CHOSEN FOR STEP AIC FUNCTION							
NEW.TCODE	RESPONSEtoOFFERS	NUMPRM12	RFA5left	pc_10	pc_23	RAMNTALL	RAMNT_13
NOEXCH	MALEMILI	LASTGIFT	RFA13left	pc_11	pc_24	MAXRAMNT	RAMNT_14
RECINHSE	VET1	AVGGIFT	RFA13mid	pc_12	pc_25	MINRAMNT	RAMNT_15
RECP3	VET2	HPHONE_D	RFA13right	pc_13	pc_26	RAMNT_3	RAMNT_16
RECPGVG	GOV1	RFA_2F	pc_1	pc_14	pc_27	RAMNT_4	RAMNT_17
RECSWEEP	GOV2	RFA_2A	pc_2	pc_15	pc_28	RAMNT_5	RAMNT_18
DOM1	SOLIH	CLUSTER2	pc_3	pc_16	pc_29	RAMNT_6	RAMNT_19
DOM2	WEALTH2	GEOCODE2	pc_4	pc_17	pc_30	RAMNT_7	RAMNT_20
CLUSTER	GEOCODE	RFA3left	pc_5	pc_18	pc_31	RAMNT_8	RAMNT_21
AGE	PEPSTRFL	RFA3mid	pc_6	pc_19	pc_32	RAMNT_9	RAMNT_22
NUMCHILD	PROM1	RFA3right	pc_7	pc_20	pc_33	RAMNT_10	RAMNT_23
GENDER	PROM2	RFA5mid	pc_8	pc_21	pc_34	RAMNT_11	RAMNT_24
INCOME	CARDPM12	RFA5right	pc_9	pc_22	TARGET_D	RAMNT_12	

The following variables where selected from the Step AIC function on which we ran the linear regression.

ATTRBUTES FOR LINEAR REGRESSION			
DOM1	RFA3left	pc_18	RAMNT_9
INCOME	RFA3mid	pc_22	RAMNT_10
PROM1	RFA5left	pc_27	RAMNT_11
PROM2	RFA13right	pc_28	RAMNT_12
CARDPM12	pc_1	pc_31	RAMNT_13
LASTGIFT	pc_2	RAMNTALL	RAMNT_14
AVGGIFT	pc_5	MAXRAMNT	RAMNT_16
HPHONE_D	pc_11	RAMNT_6	RAMNT_18
RFA_2F	pc_13	RAMNT_7	RAMNT_21
RFA_2A	pc_15	RAMNT_8	

2.2We obtained the following coefficients and p-values for the above attributes.

Coefficients: (1 not defined because of singularities)				
	Estimate	Std.Error	t value	Pr(>|t|)
(Intercept)	10.135155	0.941772	10.762	< 2e-16
DOM11	0.408737	0.535333	0.764	0.445208
DOM12	0.072649	0.536078	0.136	0.89221
DOM13	0.126012	0.536116	0.235	0.814187
DOM14	-0.001375	0.572767	-0.002	0.998085
DOM15	-0.635287	0.557616	-1.139	0.254665
INCOME	0.11922	0.054275	2.197	0.028119
PROM1	-0.043667	0.006587	-6.63	3.92E-11
PROM2	-0.020088	0.012649	-1.588	0.112345
CARDPM12	0.127843	0.085512	1.495	0.135001
LASTGIFT	0.070851	0.019439	3.645	0.000272
AVGGIFT	-0.036108	0.021128	-1.709	0.087541 .
HPHONE_D	0.26007	0.163045	1.595	0.110789
RFA_2F2	-1.247219	0.47172	-2.644	0.008232
RFA_2F3	-2.727664	0.626573	-4.353	1.38E-05
RFA_2F4	-4.511762	0.727889	-6.198	6.41E-10
RFA_2A4	1.35506	0.372844	3.634	0.000283
RFA_2A5	4.308994	0.458429	9.399	< 2e-16
RFA_2A6	8.378304	0.591045	14.175	< 2e-16
RFA3left1	-3.411683	0.787894	-4.33	1.53E-05
RFA3left2	-2.256271	0.865292	-2.608	0.009161
RFA3left3	0.729382	1.655576	0.441	0.65956
RFA3left4	-3.13197	0.899312	-3.483	0.000503
RFA3left5	-2.450216	0.816786	-3	0.002722
RFA3mid1	1.753514	0.727017	2.412	0.015923
RFA3mid2	0.06512	0.618323	0.105	0.916131

RFA3mid3	-0.231036	0.50058	-0.462	0.644444
RFA3mid4	NA	NA	NA	NA
RFA5left1	0.952138	0.301337	3.16	0.001594
RFA5left3	-0.360763	4.912417	-0.073	0.941461
RFA5left4	-0.243939	0.664912	-0.367	0.713736
RFA5left5	0.482007	0.404162	1.193	0.233108
RFA13right1	-1.190795	3.292478	-0.362	0.717621
RFA13right2	1.107232	1.446806	-0.765	0.444151
RFA13right3	-0.782574	0.34011	-2.301	0.021457
RFA13right46	-0.06354	0.252619	-0.252	0.801405
RFA13right5	0.892139	0.268539	3.322	0.000903
RFA13right6	0.574966	0.471847	1.219	0.223105
pc_1	0.004694	0.00184	2.551	0.010781
pc_2	-0.003415	0.001463	-2.334	0.019642
pc_5	0.003382	0.001722	1.964	0.049595
pc_11	0.004251	0.002871	1.481	0.138709
pc_13	-0.006466	0.00328	-1.971	0.048783
pc_15	-0.006326	0.003663	-1.727	0.084241 .
pc_18	-0.010408	0.003861	-2.696	0.007056
pc_22	-0.007953	0.004967	-1.601	0.109395
pc_27	-0.008278	0.005701	-1.452	0.146576
pc_28	0.008191	0.004172	1.963	0.049715
pc_31	0.012105	0.006107	1.982	0.047562
RAMNTALL	0.005542	0.001746	3.175	0.001514
MAXRAMNT	-0.01034	0.005178	-1.997	0.045927
RAMNT_6	0.125202	0.053569	2.337	0.019487
RAMNT_7	0.120008	0.022424	5.352	9.30E-08
RAMNT_8	0.06019	0.013736	4.382	1.21E-05
RAMNT_9	0.073101	0.015534	4.706	2.63E-06
RAMNT_10	0.028895	0.018256	1.583	0.113567

RAMNT_11	0.103922	0.016569	6.272	4.03E-10
RAMNT_12	0.053414	0.013735	3.889	0.000103
RAMNT_13	0.052053	0.018246	2.853	0.00436
RAMNT_14	0.075666	0.014449	5.237	1.74E-07
RAMNT_15	0.068385	0.021407	3.195	0.001414
RAMNT_16	0.085699	0.01468	5.838	5.80E-09
RAMNT_18	0.036193	0.015201	2.381	0.017323
RAMNT_21	0.031992	0.02117	1.511	0.130833

We obtained an Adjusted R-squared of 0.5573 which shows that our model is 55.73% close to prediction. Our p-value 2.2e^16, which is smaller than all the standard alpha's i.e 1%, 5% and 10%. Our Residual Standard Error is 4.61 which shows that our predicted values are very much close to our actual values

Q 2.2(b)

SOLUTION

We took a subset of the data with TARGET_B=1. We then used the Linear Model on this subset to calculate the predicted donation amount for each individual in the whole dataset. We then used the best classification model we had obtained from the Lift Chart (Even though the Boosted Trees Model seemed to be the better model initially, after looking at the Lift Chart we realized that the Naïve Bayes Model had a wider spread and thus would be a better model to use with the Donation Amount Model) combined with the Donation Amount Model to identify targets. We used the Confidence(1) data obtained from the Naïve Bayes Classification Model. We then multiplied the Confidence(1) from the Naïve Bayes Classification Model with the Donation Amount from the Donation Amount Model to get the predicted amount each individual would donate should they actually donate to the campaign. This product is the Predicted Donation Amount.

Below is the Confusion Matrix obtained when we combined the Naïve Bayes Model with the Donation Amount Model.

Accuracy=57.53%	True.0	True.1	**Class Precision**
Pred.0	3887	1608	70.74%
Pred.1	2613	1891	**41.98%**
Class Recall	59.80%	**54.04%**	

We also tried other classification models to see if we could get a better combined model.

Shown below is a comparison of Precision, Recall and Accuracy for the different classification models we used.

Classification Model	Precision	Recall	Accuracy
Boosted Trees	36.55%	56.96%	50.34%
Logistic Regression	36.59%	52.30%	51.59%
Naïve Bayes	41.98%	54.04%	57.53%
Random Forest	65.18%	78.91%	77.87%

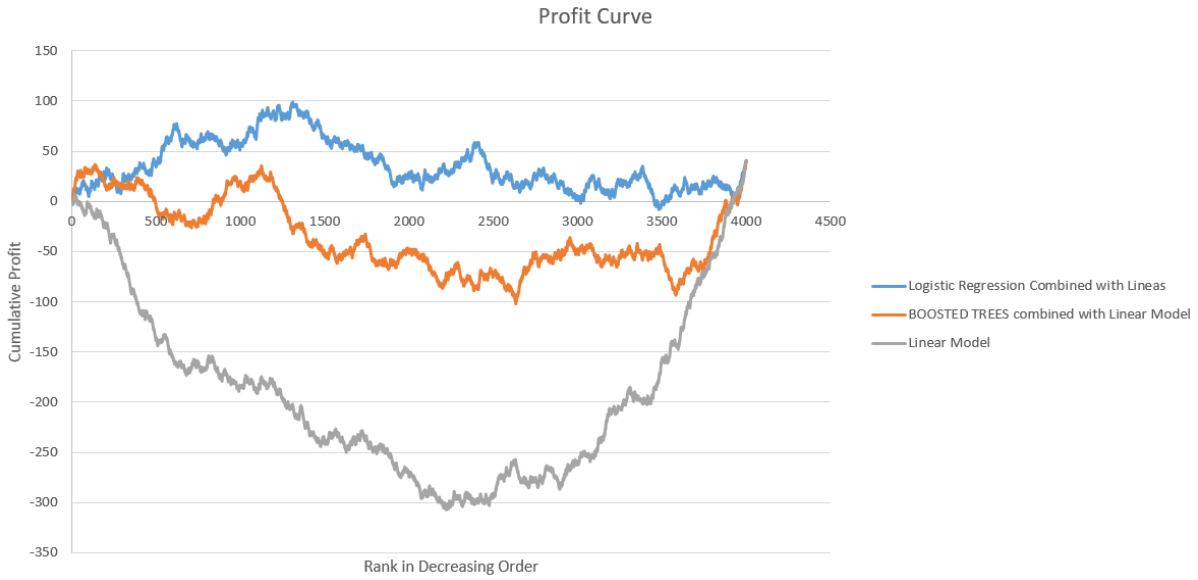
Although the Random Forest Model gave a better Class Precision, Class Recall and Accuracy, we decided not to use it since it was overfitting for the train data.

As seen above, the Naïve Bayes Model outperformed the Logistic Regression Model and the Boosted Trees Model.

Threshold

We used 0.68 as the threshold to identify which individuals to target. This means that all individuals with a Predicted Donation Amount over 0.68 were identified as targets. Since we under-sampled the data, in this case, we do not need to under-sample the cost. This is why we have used 0.68 as the threshold instead of 0.9928.

We have used 0.68 as the threshold to identify targets. The total cost of each mailing is 0.68. And the campaign would like to make a profit for each individual it targets. Thus we use 0.68 as the threshold. As any individual that donates below 0.68 would not be worth targeting, since it would become difficult to break even eventually.



We have ranked the Validation dataset by combining the Linear model with Logistic and Boosted Trees, then we took out the Profit curve for the models. Since, These Profit Curves were relatively poor as compared to the Classification model. We have used Classification Model for the Unseen Future Data.

Q.5

SOLUTION:

We will be using Boosted Trees as our Classification Model, to predict which IDs to send our Mailings. These are the parameters that we would be using to get the predictions:

No of trees	Max Depth	Min rows	Min split Improv.	No. bins	Learn Rate	Sample rate	Threshold
40	3	10	0	20	0.1	1	0.323

Please Refer to the Excel Sheet attached which contains the IDs, Predicted TARGET_B, Priority List of Mailing Columns.

We applied the model to the Unseen Data and got the following

Donors	9745
Non Donors	10255