

MAIN STREET QUANT



PRESENTS

**PHILANTHROPIC
ANALYTICS
SHOWDOWN**



BACKGROUND



- Expansion upon a graduate project / case competition
- Real life data: prominent veteran's charity, 13M+ donors
- Data:
 - Sample for model development, 50/50 balanced response, 60/40 partition
 - Frequency, Recency, Worth, Demographics
 - Costs for each mail piece
- Problem: losing money on the 'spray & pray'
 - Best expected response rate: 5.1%
- Benchmark logistic regression vs. classification tree in SAS JMP to predict likely donors
 - Maximize profit using classification under an asymmetric response
 - Generating the label printing list of likely donors



BACKGROUND



Journal of Machine Learning Research 15 (2014) 3133-3181

Submitted 11/13; Revised 4/14; Published 10/14

Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?

Manuel Fernández-Delgado

MANUEL.FERNANDEZ.DELGADO@USC.ES

Eva Cernadas

EVA.CERNADAS@USC.ES

Senén Barro

SENEN.BARRO@USC.ES

CITIUS: Centro de Investigación en Tecnologías da Información da USC

University of Santiago de Compostela

Campus Vida, 15872, Santiago de Compostela, Spain

Dinani Amorim

DINANIAMORIM@GMAIL.COM

Departamento de Tecnologia e Ciências Sociais- DTCS

Universidade do Estado da Bahia

Av. Edgard Chastinet S/N - São Geraldo - Juazeiro-BA, CEP: 48.305-680, Brasil

Editor: Russ Greiner

- Almost always parallel random forest (R/Caret), if not, then Gaussian SVM (libSVM)
- *“This is consistent with our experience running hundreds of Kaggle competitions: for most classification problems, some variation on ensembles decision trees (random forests, gradient boosted machines, etc.) performs the best.”*

- Ben Hammer



NARRATIVE



- Imagine that YOU are the Executive Director of your favorite non-profit...
 - Education, health, faith, politics, social good, etc.
 - Maximize donations, minimize costs
- Do you have more than a \$1M in your treasury?
 - University foundations
 - Political campaigns
 - Major national charities
- Do you have the means to hire a Fundraising Manager?
 - Not a programmer, uses point-and-click
- Do you have the means to hire a Data Scientist/Programmer?
 - Higher salary



NET PROFIT



	Baseline (No Sort)	
Role		
Projected Take	\$8,619,000	
Mailer Costs (\$0.68 Each)	\$8,840,000	
Pieces to Send	13,000,000	
Expected Response Rate	5.10%	
Misclassification Rate	94.90%	
Gross Profit Lift		
Gross Profit	-\$221,000	
Labor		
Burden (50%)		
Software		
Net Profit	-\$221,000	



NET PROFIT



	Baseline (No Sort)	Logistic Regression (Excel)
Role		Consultant
Projected Take	\$8,619,000	\$3,967,760
Mailer Costs (\$0.68 Each)	\$8,840,000	3,130,948
Pieces to Send	13,000,000	1,0834
Expected Response Rate	5.10%	6.51%
Misclassification Rate	94.90%	35.99%
Gross Profit Lift		445.74%
Gross Profit	-\$221,000	\$764,080
Labor		\$4,300
Burden (50%)		
Software		
Net Profit	-\$221,000	\$759,780



NET PROFIT



	Baseline (No Sort)	Logistic Regression (Excel)	Bootstrap Tree* (SAS JMP)
Role		Consultant	Fundraiser
Projected Take	\$8,619,000	\$3,967,760	\$6,187,845
Mailer Costs (\$0.68 Each)	\$8,840,000	3,130,948	1,805,338
Pieces to Send	13,000,000	1,0834	6,247
Expected Response Rate	5.10%	6.51%	17.93%
Misclassification Rate	94.90%	35.99%	17.63%
Gross Profit Lift		445.74%	2,083.04%
Gross Profit	-\$221,000	\$764,080	\$4,382,507
Labor		\$4,300	\$48,500
Burden (50%)			\$24,250
Software			\$11,000
Net Profit	-\$221,000	\$759,780	\$4,298,757



NET PROFIT



	Baseline (No Sort)	Logistic Regression (Excel)	Bootstrap Tree* (SAS JMP)	Radial SVM (R)
Role		Consultant	Fundraiser	Programmer
Projected Take	\$8,619,000	\$3,967,760	\$6,187,845	\$8,558,011
Mailer Costs (\$0.68 Each)	\$8,840,000	3,130,948	1,805,338	517,586
Pieces to Send	13,000,000	1,0834	6,247	1,791
Expected Response Rate	5.10%	6.51%	17.93%	86.49%
Misclassificatio n Rate	94.90%	35.99%	17.63%	0.83%
Gross Profit Lift		445.74%	2,083.04%	3,738.20%
Gross Profit	-\$221,000	\$764,080	\$4,382,507	\$8,040,425
Labor		\$4,300	\$48,500	\$80,000
Burden (50%)			\$24,250	\$40,000
Software			\$11,000	
Net Profit	-\$221,000	\$759,780	\$4,298,757	\$7,920,425



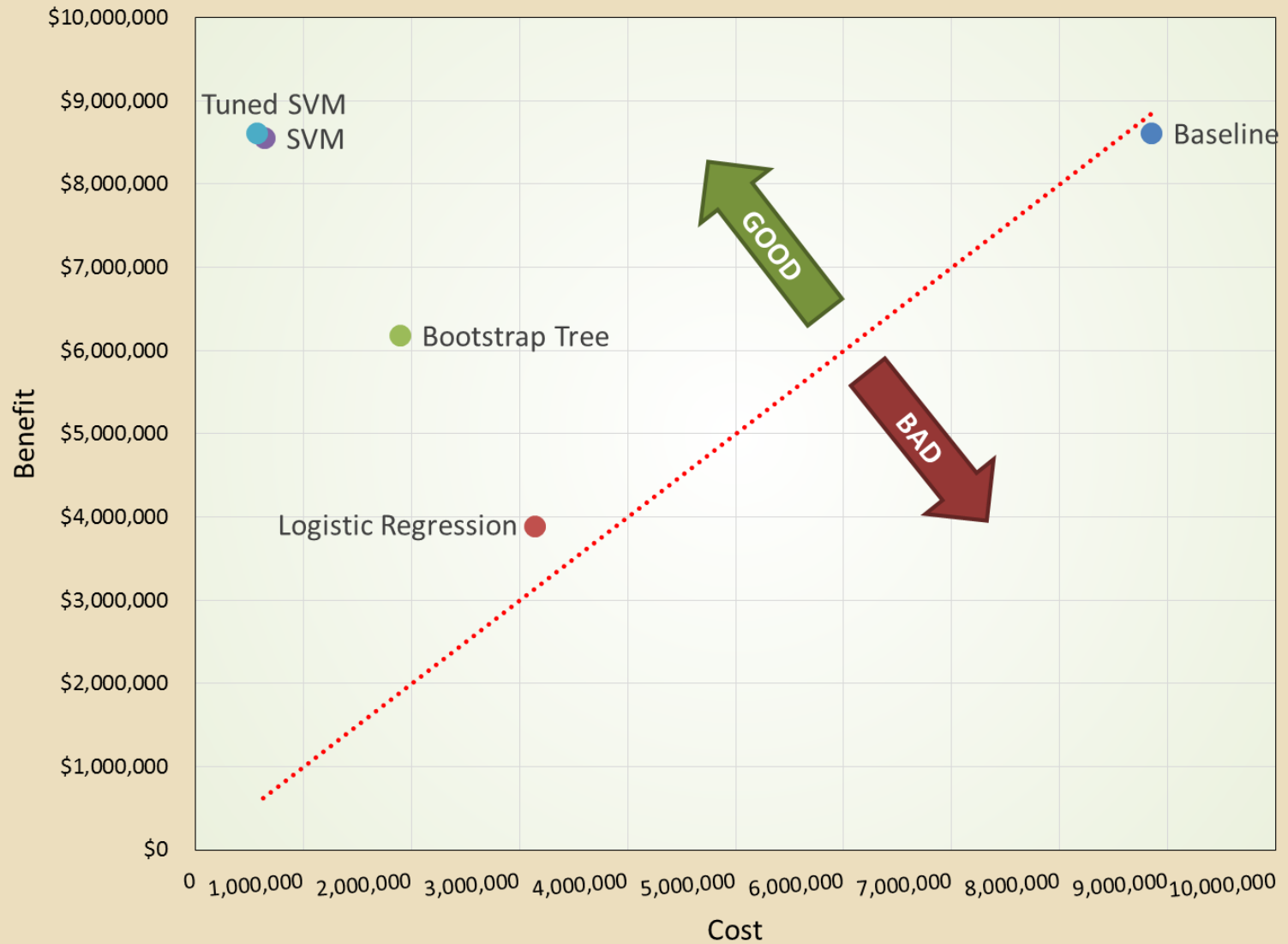
NET PROFIT



	Baseline (No Sort)	Logistic Regression (Excel)	Bootstrap Tree* (SAS JMP)	Radial SVM (R)	Tuned Radial SVM (R)
Role		Consultant	Fundraiser	Programmer	
Projected Take	\$8,619,000	\$3,967,760	\$6,187,845	\$8,558,011	\$8,618,785
Mailer Costs (\$0.68 Each)	\$8,840,000	3,130,948	1,805,338	517,586	\$450,829
Pieces to Send	13,000,000	1,0834	6,247	1,791	1,560
Expected Response Rate	5.10%	6.51%	17.93%	86.49%	100%
Misclassification Rate	94.90%	35.99%	17.63%	0.83%	0.00%
Gross Profit Lift		445.74%	2,083.04%	3,738.20%	3,795.91%
Gross Profit	-\$221,000	\$764,080	\$4,382,507	\$8,040,425	\$8,167,956
Labor		\$4,300	\$48,500	\$80,000	\$80,000
Burden (50%)			\$24,250	\$40,000	\$40,000
Software			\$11,000		
Net Profit	-\$221,000	\$759,780	\$4,298,757	\$7,920,425	\$8,047,956



COST/BENEFIT





TAKE-AWAYS



- Even small improvements in misclassification rates can lead to big financial gains.
- Expensive services do not necessarily yield the best results.
- Expensive software does not necessarily yield the best results.
- The world is attempting to automate and democratize statistical functions presently executed with programming:
 - Pro: Saves time and effort
 - Con: Greater use can lead to greater misuse. To wit:
 - Data Cleaning
 - Checking for Normality, Heteroskedasticity, Multicollinearity, Endogeneity, Variable Reduction
 - Drawing Statistical Inference from Machine Learning



GITHUB REPO



<https://github.com/JD-Freeman/Philanthropic-Analytics-Showdown>



CLICK PATH - JMP



Fundraising - JMP Pro

File Edit Tables Rows Cols DOE Analyze Graph Tools View Window Help

Notes E:\Booz Aller

1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082

Columns (24/2)
Row Id

INCOME gender dummy WEALTH HV lamed lavg IC15 NUMPRO

4 1 3 513 295 338 12
7 0 8 886 436 528 21
2 1 8 521 282 357 12
2 1 8 587 195 256 40

689 280 326 31
517 270 318 25
981 263 419 9
735 310 358 14
479 167 225 45
2463 406 460 15
484 314 347 13
809 421 444 9
384 122 150 59
1225 484 570 6

Recursively partition the data to predict a response. Classification and regression trees.

Modeling
Partition
Neural
Model Comparison
Nonlinear
Gaussian Process
Time Series
Screening
Response Screening

Partition - JMP Pro

Recursive partitioning

Select Columns
24 Columns
Row Id
Row Id.
zipconvert_2
zipconvert_3
zipconvert_4
zipconvert_5
homeowner dummy
NUMCHLD
INCOME
gender dummy
WEALTH
HV
lamed
lavg
IC15

Cast Selected Columns into Roles
Y, Response TARGET_B optional
X, Factor zipconvert_2 zipconvert_3 zipconvert_4 zipconvert_5
Weight optional numeric
Freq optional numeric
Validation Partition
By optional

Action
OK
Cancel
Remove
Recall
Help

☒ Informative Missing
Method Bootstrap Forest
Validation Portion 0

Bootstrap Forest

Bootstrap Forest Specification

Number of rows: 3120
Number of terms: 20

Number of trees in the forest 100
Number of terms sampled per split: 5
Bootstrap sample rate: 1
Minimum Splits Per Tree: 10
Maximum Splits Per Tree: 2000
Minimum Size Split: 5

☒ Early Stopping
☐ Multiple Fits over number of terms:
Max Number of terms: 10

OK Cancel



RESULTS - JMP



Bootstrap Forest for TARGET_B

Specifications

Target Column:	TARGET_B	Training rows:	3120
		Validation rows:	0
Number of trees in the forest:	100	Test rows:	0
Number of terms sampled per split:	5	Number of terms:	20
		Bootstrap samples:	3120
		Minimum Splits Per Tree:	10
		Minimum Size Split:	5

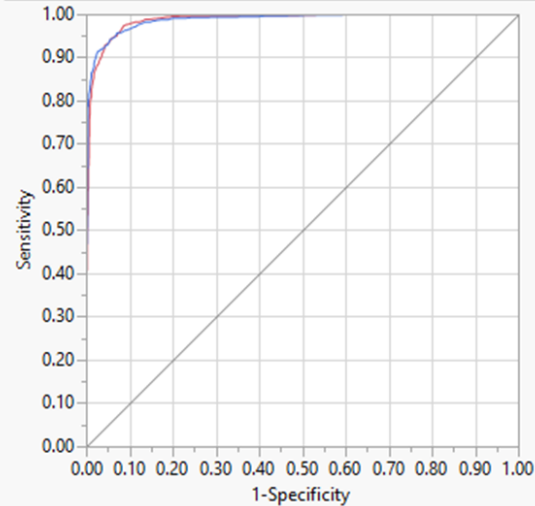
Overall Statistics

Measure	Training	Definition
Entropy RSquare	0.3183	$1 - \text{Loglike}(\text{model}) / \text{Loglike}(0)$
Generalized RSquare	0.4757	$(1 - (L(0)/L(\text{model}))^{(2/n)}) / (1 - L(0)^{(2/n)})$
Mean -Log p	0.4725	$\sum -\log(p[j])/n$
RMSE	0.3797	$\sqrt{\sum (y[j] - p[j])^2 / n}$
Mean Abs Dev	0.3716	$\sum y[j] - p[j] / n$
Misclassification Rate	0.0564	$\sum (p[j] \neq \text{pMax}) / n$
N	3120	n

Confusion Matrix

Actual \ Predicted	0	1
Training 0	1483	77
1	99	1461

Receiver Operating Characteristic

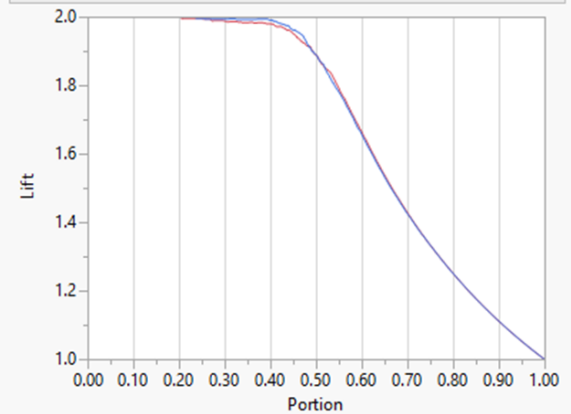


TARGET_B	Area
0	0.9889
1	0.9889

Column Contributions

Term	Number of Splits	G^2	Portion
NUMPROM	2076	9510.49667	0.0851
AVGGIFT	1917	9059.63772	0.0810
RAMNTALL	1953	9043.14548	0.0809
HV	1946	9006.7556	0.0806
IC15	2045	8690.55212	0.0777
totalmonths	1977	8629.61547	0.0772
lcmcd	1885	8557.62277	0.0766
lcavg	1809	8055.04086	0.0721
TIMELAG	1899	7643.88654	0.0684
LASTGIFT	1681	7397.7376	0.0662
MAXRAMNT	1672	7268.52299	0.0650
INCOME	1720	5626.2538	0.0503
WEALTH	1186	4077.8348	0.0365
gender dummy	1081	1996.14547	0.0179
homeowner dummy	790	1509.15954	0.0135
zipconvert_5	809	1477.9782	0.0132
zipconvert_2	645	1163.33475	0.0104
zipconvert_4	650	1131.61257	0.0101
zipconvert_3	603	1060.59827	0.0095
NUMCHLD	258	878.5888	0.0079

Lift Curve



TARGET_B	Area
0	0.9889
1	0.9889



R CODE



```
# This is the run of the parallel random forest. It did not beat SAS JMP bootstrap forest.

# Remember to set your working directory, install needed libraries, and set seed to 1.

> library("randomForest", lib.loc=~R/win-library/3.1")
> library("foreach", lib.loc=~R/win-library/3.1")
> library("doParallel", lib.loc=~R/win-library/3.1")

> model <- read.csv("model.csv")
> response <- as.factor(model$TARGET_B)
> predictors <- read.csv('predictors.csv')
> MyRF <- train(predictors, response, method = "parRF")
> getTree(MyRF$finalModel)
> head(MyRF$finalModel$predicted)
> MyRFresult <- MyRF$finalModel$predicted
> write.csv(MyRFresult, file = "MyRFresult.csv", row.names = FALSE)



---


# This is the run of the SVM and the tuned SVM

> library("e1071", lib.loc=~R/win-library/3.1")
> dataframe <- data.frame(x=predictors, y=response)
> svmfit <- svm(y~., data=dataframe, kernel="radial", gamma=1, cost=1)
> str(svmfit)
> write.csv(svmresult, file = "svmresult.csv", row.names = FALSE)
> head(svmfit$fitted)
> svmresult <- svmfit$fitted
> tune.out=tune(svm, y~., dat=dataframe, kernel="radial", ranges=list(cost=c(0.1,1,10,100,1000),gamma=c(0.5,1,2,3,4)))
> summary(tune.out)
#best performance found is cost 100, gamma 0.5
> svmfit <- svm(y~., data=dataframe, kernel="radial", gamma=.5, cost=100)
> tunedsvmresult <- svmfit$fitted
> write.csv(tunedsvmresult, file = "tunedsvmresult.csv", row.names = FALSE)
```




MATRICES



Logistic Regression

Confusion Matrix

Actual \ Predicted	0	1
Training 0	854	706
1	679	881

Reweighted:

		Predicted		
		0	1	
Actual	0	15891	13137	29,028
	1	679	881	1,560
		16570	14018	30,588
misclassification rate		0.45168		

SAS JMP Bootstrap Forest*

Confusion Matrix

Actual \ Predicted	0	1
Training 0	1483	77
1	99	1461

Reweighted:

		Predicted		
		0	1	
Actual	0	27595	1433	29,028
	1	99	1461	1,560
		27694	2894	30,588
misclassification rate		0.050085		

R, Best of RF Runs

Confusion Matrix

	0	1	
0	867	693	1560
1	707	853	1560
	1574	1546	3120

Reweighted:

		Predicted		
		0	1	
Actual	0	16,133	12895	29,028
	1	707	853	1560
		16,840	13748	30,588
Misclassification Rate:		0.445		

R Support Vector Machine

Count of x	Column Labels		
Row Labels	0	1 (blank)	Grand Total
0	1547	13	1560
1	11	1549	1560
(blank)			
Grand Total	1558	1562	3120
Reweighted:			
Actual	0	1	
	28,786	242	29,028
1	11	1549	1560
	28,797	1791	30,588
Misclassification Rate:		0.0083	

R Support Vector Machine Tuned

Count of Tuned	Column Labels		
Row Labels	0	1 (blank)	Grand Total
0	1560		1560
1		1560	1560
(blank)			
Grand Total	1560	1560	3120
Reweighted:			
Actual	0	1	
	29,028	0	29,028
1	0	1560	1560
	29,028	1560	30,588
Misclassification Rate:		0	