**Rapleaf Hackathon App Note**

Doug Chang [dougc@stanfordalumni.org](mailto:dougc@stanfordalumni.org)

References:

<http://avesbiodiv.mncn.csic.es/estadistica/bt1.pdf>

<http://cran.r-project.org/web/packages/dismo/vignettes/brt.pdf>

This document shows the contribution of the Rapleaf Demgraphic data using Boosted Regression Trees under R package “dismo.” The Rapleaf demographic data has an “importance” of 25.91% to a ROC curve accuracy of .748.

This translates to being 24.8% better than a random guess of who would become a subscriber.

As a comparison to the winning score, this is 5.2% less than the top entry.

Note: all terms in quotes are defined in the context of this document only and may conflict with definitions in other literature.

**Boosted Regression Trees:**

The advantage of BRTs over other general classification methods like SVMs is they are resistant to over fitting.

The biggest factor for choosing this package was the availability of the analytic functions available in this particular package which enables the user to see the relative importance of the different predictor variables.

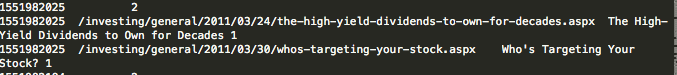
**Preprocessing the data:**

No features were added. Features from the weblogs were removed. The url data from the headline file was not used.

Processing consisted of changing the location variable in the demographics from a categorical variable to a numeric value because R couldn’t support the number of countries in the data set as a categorical variable. R returned an error message indicating there were “too many categories.”

**Methodology**

An example screenshot of the headline.tsv file is shown below. We use the first line for uid 1551982025 which has no url but a count of 2, the first line. This count 2 is extracted and renamed to pageV in file combining the demographics and headline files into 1 file. Threw away the urls visited information.



Preprocess the data by running MakeData.java .The program takes as input 4 data files, headlines.tsv, trainng.csv, example\_entry.csv, demographics.tsv and produces 2 output files, headlineTraining.txt and headlineEntry.txt. <https://gist.github.com/1288158> .

The most important contribution seems to be from page views from the headlines.tsv file with a 74.05% predictive importance to a total of 25.91% for all the demographics attributes.

<https://gist.github.com/1289173>

Lets combine the features, all of them into 1 file including converting the locations into a numeric range

***> headlineTraining<-read.csv(file="/Users/dc/Desktop/HeadlineTraining.txt", header=TRUE ,sep="\t")***

***> training<-read.csv(file="/Users/dc/Desktop/training.txt", header=TRUE, sep="\t")***

***> training1a<-data.frame(beh=training$beh, pageV=headlineTraining$pageV, age=training$age, sex=training$sex, child=training$child, home=training$homeval, income=training$houseincome, residlen=training$len\_resid, prop=training$prop\_type, marital=training$marital, own=training$owner, loc=training$loc)***

***> gbm1b<-gbm.step(data=training1a, gbm.x=2:12 ,gbm.y=1, fold.vector=NULL, tree.complexity=4, learning.rate=0.01)***

GBM STEP - version 2.9

Performing cross-validation optimisation of a boosted regression tree model

for beh with dataframe training1a and using a family of bernoulli

Using 201398 observations and 11 predictors

creating 10 initial models of 50 trees

folds are stratified by prevalence

total mean deviance = 0.4228

tolerance is fixed at 4e-04

ntrees resid. dev.

50 0.4012

now adding trees...

100 0.3919

150 0.3874

200 0.3852

250 0.3839

300 0.3832

350 0.3828

400 0.3826

450 0.3824

500 0.3823

550 0.3821

600 0.382

650 0.3819

700 0.3818

750 0.3817

800 0.3817

850 0.3816

900 0.3815

950 0.3814

1000 0.3814

1050 0.3813

1100 0.3812

1150 0.3811

1200 0.3811

1250 0.381

1300 0.381

1350 0.3809

1400 0.3808

1450 0.3808

1500 0.3807

1550 0.3807

1600 0.3806

1650 0.3806

1700 0.3805

1750 0.3805

1800 0.3804

1850 0.3804

1900 0.3803

1950 0.3803

2000 0.3803

2050 0.3802

2100 0.3802

2150 0.3801

2200 0.3801

2250 0.3801

2300 0.38

2350 0.38

fitting final gbm model with a fixed number of 2350 trees for beh

mean total deviance = 0.423

mean residual deviance = 0.376

estimated cv deviance = 0.38 ; se = 0.001

training data correlation = 0.223

cv correlation = 0.199 ; se = 0.002

training data ROC score = 0.76

cv ROC score = 0.748 ; se = 0.002

elapsed time - 358.72 minutes

The command summary graphs the relative predictive contribution from each variable. We can see location is the most important after page views to the home page.

***> summary(gbm1b)***

var rel.inf

1 pageV 74.0567852

2 loc 11.0801383

3 income 4.1565597

4 age 3.1426519

5 residlen 3.0813927

6 home 2.3308287

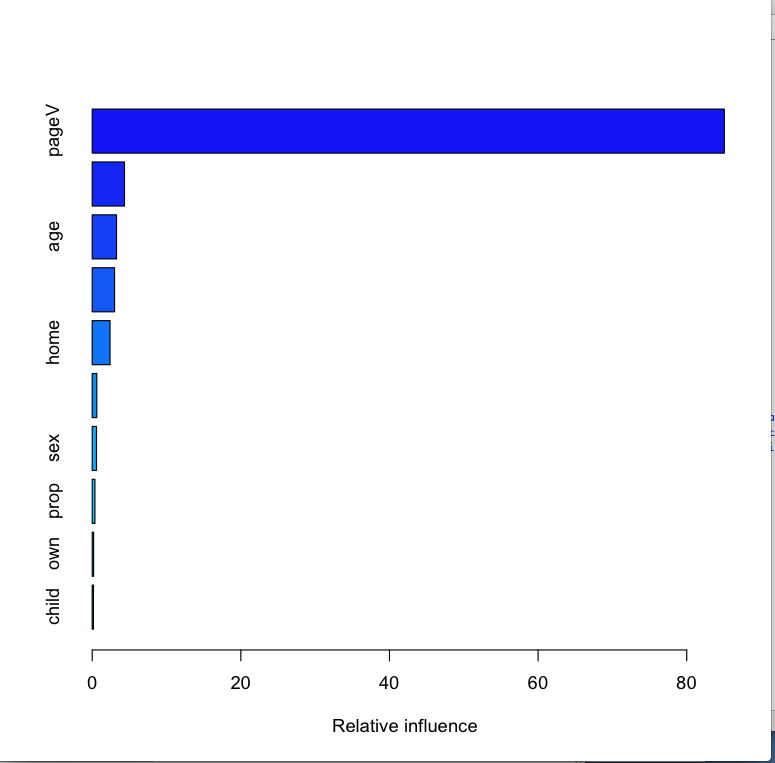
7 marital 0.6560258

8 sex 0.6476549

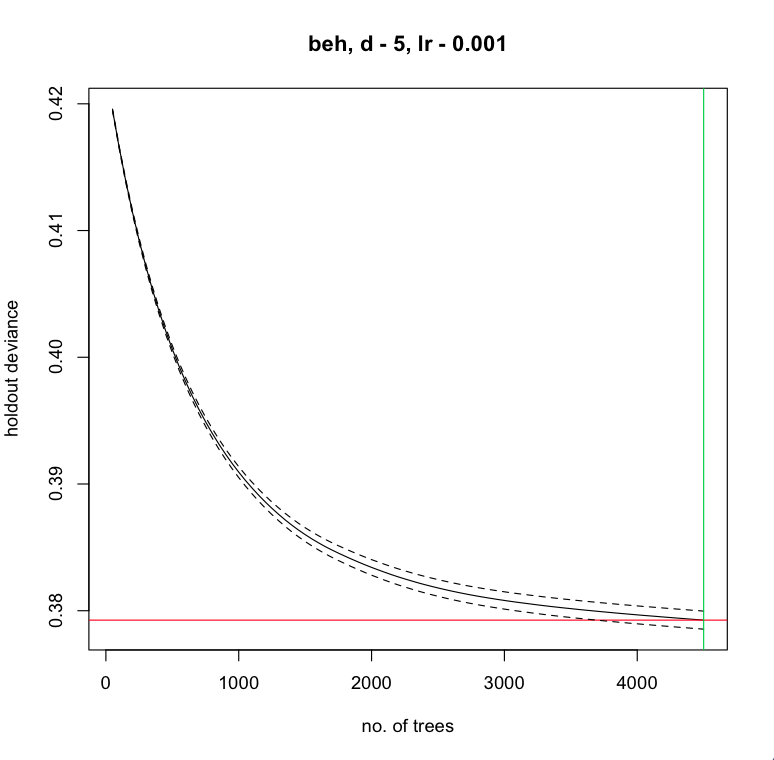
9 prop 0.3817017

10 child 0.2632598

11 own 0.2030012



Here is a graph of the difference in features, with location being the most important demographic feature because we converted this to a numeric attribute. This probably indicates we need to convert the other attributes into a numeric attribute and then measure the change.

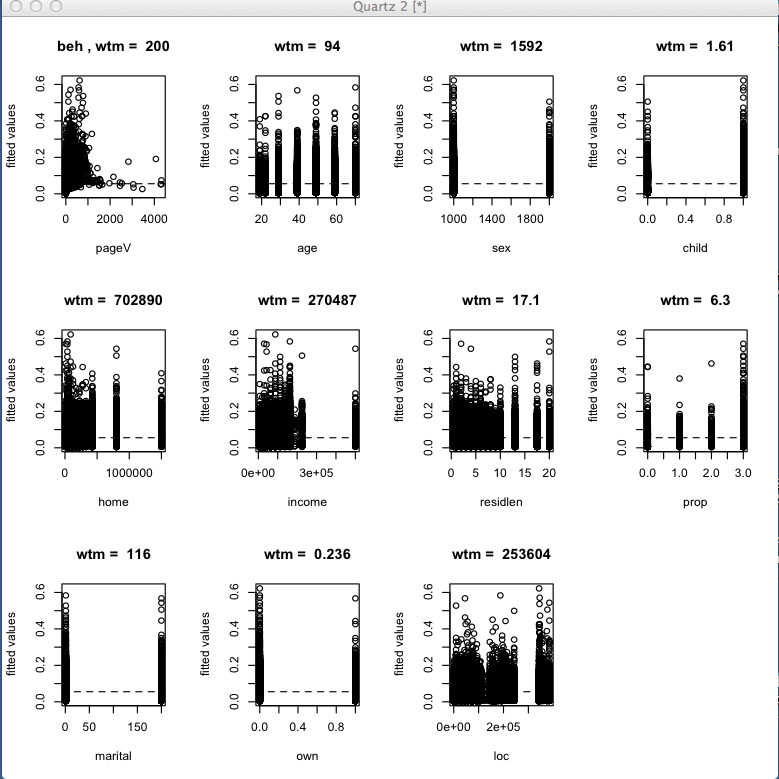


The intersection of the red and green lines indicate where the minimum is.

The graph above is bad in the sense we are looking for the holdout deviance to start rising again. If we see that then we know roughly how to pick the number of trees to run in our production model. We can see th holdout deviance decreasing with the number of trees still. This is also a sign we need to spend some time with cleaning the features given the amount the holdout deviance is decreasing as the number of trees increase. This is also an indication we have correlated features.

> gbm.plot(gbm2, nplots=12)

> gbm.plot.fits(gbm1b) The model performs better when converting the location from a categorical variable to a numerical one.



**Qualitatively we can look at the plots and try to discern Bernoulli probability distributions.**

**Find the Interactions between the predictor vars.**

* find.int<-gbm.interactions(gbm1b)