ECE-9603A Project: Code

Joe Abley – jabley@uwo.ca

2018-12-07

Introduction

This document contains the R code with output and surrounding commentary used in the preparation of the project report[1] for course ECE-9603A, fall 2018, Western University.

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

R Setup

Source Dataset

Each observation in the source dataset corresponds to a five-minute interval during which responses were sent to a single client. Those clients are identified as "google", "facebook" and "other" in the "origin" column; other columns contain features that describe characteristics of the responses sent to that client during that window.

```
sourceData = read.csv("/Users/jabley/Downloads/data/summary.csv")
summary(sourceData)
```

```
##       origin          day              hour          responses
##  facebook: 923432   fri:395665   Min.   : 0.00   Min.   :      1.0
##  google  :1954409   mon:417453   1st Qu.: 6.00   1st Qu.:     99.0
##  other   :  17212   sat:397661   Median :12.00   Median :    223.0
##                     sun:397978   Mean   :11.54   Mean   :    406.8
##                     thu:399208   3rd Qu.:18.00   3rd Qu.:    365.0
##                     tue:443392   Max.   :23.00   Max.   :1133319.0
##                     wed:443696
##  max_labelsize    mean_labelsize     tlds_seen        slds_seen
##  Min.   :  2.00   Min.   : 1.250   Min.   :  1.00   Min.   :      0.0
##  1st Qu.: 25.00   1st Qu.: 5.300   1st Qu.: 10.00   1st Qu.:     73.0
##  Median : 31.00   Median : 5.562   Median : 16.00   Median :    159.0
##  Mean   : 36.73   Mean   : 5.631   Mean   : 20.49   Mean   :    258.9
##  3rd Qu.: 36.00   3rd Qu.: 5.949   3rd Qu.: 21.00   3rd Qu.:    260.0
##  Max.   :240.00   Max.   :21.000   Max.   :234.00   Max.   :1064196.0
##
##   prop_1_label        prop_2_label     prop_3_label     prop_4_label
##  Min.   :0.0000000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0000000   1st Qu.:0.2099   1st Qu.:0.4847   1st Qu.:0.03125
##  Median :0.0000000   Median :0.2724   Median :0.5865   Median :0.07073
##  Mean   :0.0007465   Mean   :0.3363   Mean   :0.5477   Mean   :0.07995
##  3rd Qu.:0.0000000   3rd Qu.:0.3780   3rd Qu.:0.6629   3rd Qu.:0.10156
##  Max.   :1.0000000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
```

[1] J. Abley, "Identifying the True Origin of DNS Traffic Without Reference to Client Source Address," ECE-9603A, Western University, Dec 2018. [Online]. Available: https://github.com/ableyjoe/uwo-mesc/tree/master/ECE-9603A-001-GF18/project

```
##
##   prop_rcode_0  prop_rcode_3  prop_qtype_1     prop_qtype_2
##  Min.   :1     Min.   :0     Min.   :0.0000   Min.   :0.000000
##  1st Qu.:1     1st Qu.:0     1st Qu.:0.5833   1st Qu.:0.000000
##  Median :1     Median :0     Median :0.7469   Median :0.002597
##  Mean   :1     Mean   :0     Mean   :0.7102   Mean   :0.010200
##  3rd Qu.:1     3rd Qu.:0     3rd Qu.:0.8182   3rd Qu.:0.011111
##  Max.   :1     Max.   :0     Max.   :1.0000   Max.   :1.000000
##
##   prop_qtype_5        prop_qtype_6      prop_qtype_10
##  Min.   :0.0000000   Min.   :0.000000   Min.   :0.0000000
##  1st Qu.:0.0000000   1st Qu.:0.000000   1st Qu.:0.0000000
##  Median :0.0000000   Median :0.000000   Median :0.0000000
##  Mean   :0.0008842   Mean   :0.016616   Mean   :0.0003181
##  3rd Qu.:0.0000000   3rd Qu.:0.007712   3rd Qu.:0.0000000
##  Max.   :1.0000000   Max.   :1.000000   Max.   :0.0485437
##
##  prop_qtype_12      prop_qtype_15     prop_qtype_16     prop_qtype_28
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.06593
##  Median :0.000000   Median :0.03185   Median :0.01744   Median :0.11017
##  Mean   :0.001139   Mean   :0.03839   Mean   :0.02289   Mean   :0.18204
##  3rd Qu.:0.000000   3rd Qu.:0.05728   3rd Qu.:0.03846   3rd Qu.:0.17994
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##  prop_qtype_33      prop_qtype_35       prop_qtype_37
##  Min.   :0.000000   Min.   :0.000e+00   Min.   :0.000e+00
##  1st Qu.:0.000000   1st Qu.:0.000e+00   1st Qu.:0.000e+00
##  Median :0.003623   Median :0.000e+00   Median :0.000e+00
##  Mean   :0.007861   Mean   :7.396e-05   Mean   :6.092e-06
##  3rd Qu.:0.013263   3rd Qu.:0.000e+00   3rd Qu.:0.000e+00
##  Max.   :1.000000   Max.   :7.692e-02   Max.   :1.724e-02
##
##  prop_qtype_38       prop_qtype_43      prop_qtype_44
##  Min.   :0.000e+00   Min.   :0.000000   Min.   :0.00e+00
##  1st Qu.:0.000e+00   1st Qu.:0.000000   1st Qu.:0.00e+00
##  Median :0.000e+00   Median :0.002481   Median :0.00e+00
##  Mean   :1.489e-05   Mean   :0.006567   Mean   :2.26e-06
##  3rd Qu.:0.000e+00   3rd Qu.:0.010152   3rd Qu.:0.00e+00
##  Max.   :1.053e-01   Max.   :1.000000   Max.   :4.00e-02
##
##  prop_qtype_46       prop_qtype_48        prop_qtype_52
##  Min.   :0.000e+00   Min.   :0.0000000   Min.   :0.000e+00
##  1st Qu.:0.000e+00   1st Qu.:0.0000000   1st Qu.:0.000e+00
##  Median :0.000e+00   Median :0.0000000   Median :0.000e+00
##  Mean   :8.442e-05   Mean   :0.0008687   Mean   :2.880e-06
##  3rd Qu.:0.000e+00   3rd Qu.:0.0000000   3rd Qu.:0.000e+00
##  Max.   :2.358e-02   Max.   :1.0000000   Max.   :5.263e-02
##
##  prop_qtype_99       prop_qtype_255     prop_qtype_256
##  Min.   :0.0000000   Min.   :0.000000   Min.   :0.000e+00
##  1st Qu.:0.0000000   1st Qu.:0.000000   1st Qu.:0.000e+00
##  Median :0.0000000   Median :0.000000   Median :0.000e+00
##  Mean   :0.0001934   Mean   :0.001486   Mean   :1.420e-06
```

```
##  3rd Qu.:0.0000000   3rd Qu.:0.000000   3rd Qu.:0.000e+00
##  Max.   :1.0000000   Max.   :1.000000   Max.   :4.348e-02
##
##  prop_qtype_257      prop_qtype_32769
##  Min.   :0.0000000   Min.   :0.000e+00
##  1st Qu.:0.0000000   1st Qu.:0.000e+00
##  Median :0.0000000   Median :0.000e+00
##  Mean   :0.0001625   Mean   :9.170e-06
##  3rd Qu.:0.0000000   3rd Qu.:0.000e+00
##  Max.   :1.0000000   Max.   :9.091e-02
##
```

Zero-Variance Predictors

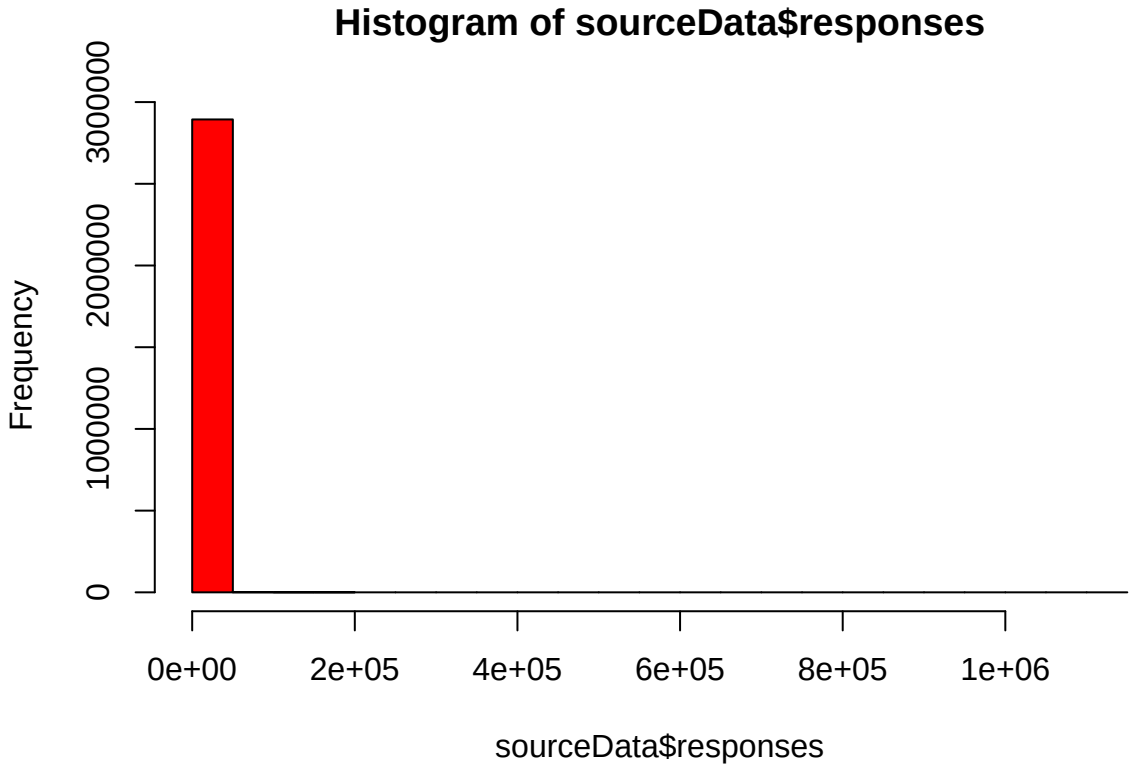
For reasons that are not entirely clear, this dataset contains only observations about responses with RCODE=0 (NOERROR), and none with RCODE=3 (NXDOMAIN). This seems indicative of some kind of error in the collection process, which is worthy of attention at some point. For the purposes of this project, however, we will simply remove those columns since they are effectively constants.

```
sourceData <- sourceData[, ! names(sourceData) %in% c("prop_rcode_0", "prop_rcode_3")]
```

Outliers

The *responses* predictor shows a maximum value dramatically larger than the median. It seems likely that there are outliers in the dataset that we could usefully eliminate in order to build a model that functions more sensibly over most test data.

```
hist(sourceData$responses, col="red")
```

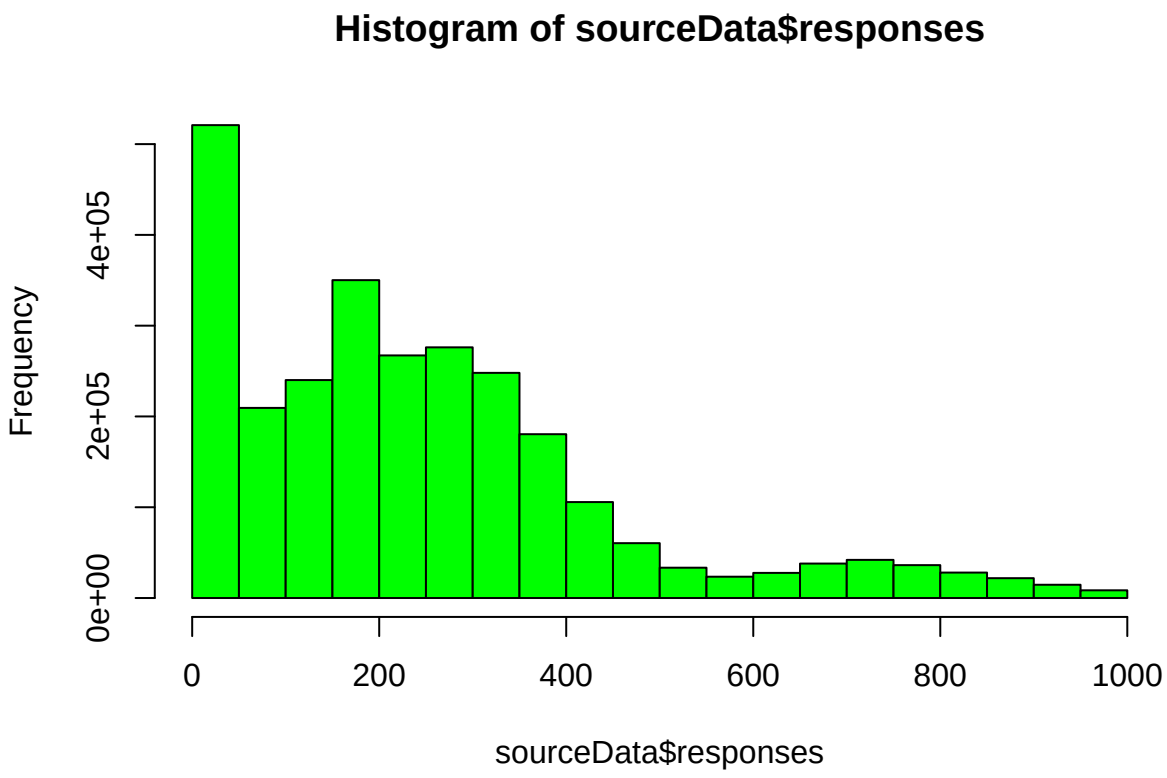


The vast majority of the values are less than 1000, so we shall eliminate all observations from the training set that have values that are higher:

```
sourceData <- sourceData[sourceData$responses < 1000,]
summary(sourceData$responses)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1      88     208     247     337     999
```

```
hist(sourceData$responses, col="green")
```



Addressing Class Imbalance

The dataset is currently inbalanced by "origin":

```
summary(sourceData$origin)
```

```
## facebook   google    other
##   917920  1807505     6811
```

Since we have a fairly large number of observations (more than sufficient for building plausible models, and still plenty left over for testing) we will balance these datasets by understampling the facebook and google observations. This will also have the happy side-effect of making the dataset smaller and more manageable for ad-hoc experimentation. We use random undersampling since there is no obvious difference in the relevance of each class; our data collection simply didn't collect an equal number of observations.

```
sourceData <- RandUnderClassif(origin ~ ., sourceData, "balance")
summary(sourceData$origin)
```

```
## facebook   google    other
##     6811     6811     6811
```

Normalisation

The *responses*, *hour*, *max_labelsize*, *mean_labelsize*, *tlds_seen* and *slds_seen* predictors are not scaled between [0, 1], so we rescale them.

Many of our proportional predictors are already scaled within a range of [0, 1]. Some predictors that relate to rare query types don't appear in observations with values very close to the upper bound of the possible range (i.e. they are much closer to zero) but their relative values compared to other predictors seem important to preserve, so we shall leave them as-is.

```
to_rescale = c("responses", "hour", "max_labelsize", "mean_labelsize", "tlds_seen")
summary(sourceData[to_rescale])
```

```
##    responses          hour       max_labelsize    mean_labelsize
##  Min.   :  1.0   Min.   : 0.00   Min.   :  2.00   Min.   : 1.287
##  1st Qu.: 26.0   1st Qu.: 5.00   1st Qu.: 19.00   1st Qu.: 5.168
##  Median :155.0   Median :11.00   Median : 26.00   Median : 5.527
##  Mean   :204.6   Mean   :11.33   Mean   : 27.74   Mean   : 5.509
##  3rd Qu.:311.0   3rd Qu.:17.00   3rd Qu.: 32.00   3rd Qu.: 5.945
##  Max.   :999.0   Max.   :23.00   Max.   :240.00   Max.   :15.000
##    tlds_seen
##  Min.   :  1.00
##  1st Qu.:  5.00
##  Median : 11.00
##  Mean   : 12.33
##  3rd Qu.: 17.00
##  Max.   :124.00
```

```
sourceData[to_rescale] <- lapply(sourceData[to_rescale], rescale)
summary(sourceData[to_rescale])
```

```
##    responses            hour        max_labelsize     mean_labelsize
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.02505   1st Qu.:0.2174   1st Qu.:0.07143   1st Qu.:0.2830
##  Median :0.15431   Median :0.4783   Median :0.10084   Median :0.3092
##  Mean   :0.20402   Mean   :0.4925   Mean   :0.10814   Mean   :0.3079
##  3rd Qu.:0.31062   3rd Qu.:0.7391   3rd Qu.:0.12605   3rd Qu.:0.3397
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000
##    tlds_seen
##  Min.   :0.00000
##  1st Qu.:0.03252
##  Median :0.08130
##  Mean   :0.09210
##  3rd Qu.:0.13008
##  Max.   :1.00000
```

Training and Test Datasets

We will split our source data into training and test datasets (we will do k-fold cross-validation across the training set without a separate validation set). We choose 75% of the source data for the training set and the remainder in the test set.

```
data_split <- initial_split(sourceData, prop = 0.75)
trainingData <- training(data_split)
testData <- testing(data_split)
```

```
summary(trainingData)
```

```
##       origin      day            hour          responses
##  facebook:5097   fri:2155   Min.   :0.0000   Min.   :0.00000
##  google  :5114   mon:2247   1st Qu.:0.2174   1st Qu.:0.02505
##  other   :5114   sat:2092   Median :0.4783   Median :0.15531
##                  sun:2052   Mean   :0.4913   Mean   :0.20536
##                  thu:2207   3rd Qu.:0.7391   3rd Qu.:0.31263
##                  tue:2379   Max.   :1.0000   Max.   :1.00000
##                  wed:2193
##  max_labelsize     mean_labelsize     tlds_seen         slds_seen
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.00000   Min.   :  0.0
##  1st Qu.:0.07143   1st Qu.:0.2832   1st Qu.:0.03252   1st Qu.: 18.0
##  Median :0.10084   Median :0.3094   Median :0.08130   Median :110.0
##  Mean   :0.10815   Mean   :0.3083   Mean   :0.09281   Mean   :136.5
##  3rd Qu.:0.12605   3rd Qu.:0.3399   3rd Qu.:0.13008   3rd Qu.:202.0
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.00000   Max.   :874.0
##
##   prop_1_label      prop_2_label     prop_3_label     prop_4_label
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.1842   1st Qu.:0.3750   1st Qu.:0.00000
##  Median :0.00000   Median :0.2854   Median :0.5535   Median :0.05907
##  Mean   :0.00423   Mean   :0.3425   Mean   :0.5164   Mean   :0.10077
##  3rd Qu.:0.00000   3rd Qu.:0.4519   3rd Qu.:0.6690   3rd Qu.:0.10796
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##
##   prop_qtype_1     prop_qtype_2       prop_qtype_5
##  Min.   :0.0000   Min.   :0.000000   Min.   :0.0000000
##  1st Qu.:0.4929   1st Qu.:0.000000   1st Qu.:0.0000000
##  Median :0.6667   Median :0.000000   Median :0.0000000
##  Mean   :0.6236   Mean   :0.007770   Mean   :0.0003981
##  3rd Qu.:0.8038   3rd Qu.:0.006369   3rd Qu.:0.0000000
##  Max.   :1.0000   Max.   :0.647619   Max.   :0.1428570
##
##   prop_qtype_6      prop_qtype_10       prop_qtype_12
##  Min.   :0.000000   Min.   :0.0000000   Min.   :0.0000000
##  1st Qu.:0.000000   1st Qu.:0.0000000   1st Qu.:0.0000000
##  Median :0.000000   Median :0.0000000   Median :0.0000000
##  Mean   :0.049889   Mean   :0.0001365   Mean   :0.0007508
##  3rd Qu.:0.003663   3rd Qu.:0.0000000   3rd Qu.:0.0000000
##  Max.   :1.000000   Max.   :0.0200000   Max.   :0.3208960
##
##  prop_qtype_15     prop_qtype_16     prop_qtype_28     prop_qtype_33
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.04167   1st Qu.:0.000000
##  Median :0.00000   Median :0.00000   Median :0.12336   Median :0.000000
##  Mean   :0.02141   Mean   :0.01127   Mean   :0.22582   Mean   :0.004385
##  3rd Qu.:0.03233   3rd Qu.:0.01702   3rd Qu.:0.46835   3rd Qu.:0.003040
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.000000
##
##  prop_qtype_35       prop_qtype_37       prop_qtype_38
##  Min.   :0.000e+00   Min.   :0.000e+00   Min.   :0.000e+00
##  1st Qu.:0.000e+00   1st Qu.:0.000e+00   1st Qu.:0.000e+00
##  Median :0.000e+00   Median :0.000e+00   Median :0.000e+00
```

```
##  Mean   :3.492e-05   Mean   :1.682e-06   Mean   :7.927e-06
##  3rd Qu.:0.000e+00   3rd Qu.:0.000e+00   3rd Qu.:0.000e+00
##  Max.   :1.324e-02   Max.   :5.181e-03   Max.   :2.564e-02
##
##  prop_qtype_43      prop_qtype_44       prop_qtype_46
##  Min.   :0.000000   Min.   :0.000e+00   Min.   :0.000e+00
##  1st Qu.:0.000000   1st Qu.:0.000e+00   1st Qu.:0.000e+00
##  Median :0.000000   Median :0.000e+00   Median :0.000e+00
##  Mean   :0.036722   Mean   :1.339e-06   Mean   :2.721e-05
##  3rd Qu.:0.008287   3rd Qu.:0.000e+00   3rd Qu.:0.000e+00
##  Max.   :1.000000   Max.   :7.143e-03   Max.   :1.149e-02
##
##  prop_qtype_48      prop_qtype_52       prop_qtype_99
##  Min.   :0.000000   Min.   :0.000e+00   Min.   :0.0000000
##  1st Qu.:0.000000   1st Qu.:0.000e+00   1st Qu.:0.0000000
##  Median :0.000000   Median :0.000e+00   Median :0.0000000
##  Mean   :0.008433   Mean   :3.241e-07   Mean   :0.0001127
##  3rd Qu.:0.000000   3rd Qu.:0.000e+00   3rd Qu.:0.0000000
##  Max.   :1.000000   Max.   :2.217e-03   Max.   :0.0666667
##
##  prop_qtype_255      prop_qtype_256     prop_qtype_257
##  Min.   :0.0000000   Min.   :0.00e+00   Min.   :0.000000
##  1st Qu.:0.0000000   1st Qu.:0.00e+00   1st Qu.:0.000000
##  Median :0.0000000   Median :0.00e+00   Median :0.000000
##  Mean   :0.0007642   Mean   :1.14e-06   Mean   :0.008413
##  3rd Qu.:0.0000000   3rd Qu.:0.00e+00   3rd Qu.:0.000000
##  Max.   :0.1428570   Max.   :1.25e-02   Max.   :1.000000
##
##  prop_qtype_32769
##  Min.   :0.000e+00
##  1st Qu.:0.000e+00
##  Median :0.000e+00
##  Mean   :1.137e-05
##  3rd Qu.:0.000e+00
##  Max.   :9.091e-02
##
```

```
summary(testData)
```

```
##       origin      day           hour          responses
##  facebook:1714   fri:754   Min.   :0.0000   Min.   :0.00000
##  google  :1697   mon:733   1st Qu.:0.2609   1st Qu.:0.02705
##  other   :1697   sat:680   Median :0.4783   Median :0.15130
##                  sun:695   Mean   :0.4960   Mean   :0.20002
##                  thu:739   3rd Qu.:0.7391   3rd Qu.:0.30561
##                  tue:770   Max.   :1.0000   Max.   :0.99900
##                  wed:737
##  max_labelsize     mean_labelsize        tlds_seen         slds_seen
##  Min.   :0.00000   Min.   :0.0008802   Min.   :0.00000   Min.   :  0.0
##  1st Qu.:0.07143   1st Qu.:0.2825443   1st Qu.:0.03252   1st Qu.: 18.0
##  Median :0.10084   Median :0.3085493   Median :0.08130   Median :108.0
##  Mean   :0.10812   Mean   :0.3068399   Mean   :0.08998   Mean   :133.3
##  3rd Qu.:0.12605   3rd Qu.:0.3389715   3rd Qu.:0.13008   3rd Qu.:200.0
##  Max.   :0.85714   Max.   :0.8298473   Max.   :0.98374   Max.   :832.0
##
```

```
##   prop_1_label       prop_2_label     prop_3_label     prop_4_label
##  Min.   :0.000000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.1897   1st Qu.:0.3803   1st Qu.:0.00000
##  Median :0.000000   Median :0.2839   Median :0.5568   Median :0.05671
##  Mean   :0.003639   Mean   :0.3448   Mean   :0.5196   Mean   :0.09739
##  3rd Qu.:0.000000   3rd Qu.:0.4490   3rd Qu.:0.6737   3rd Qu.:0.10764
##  Max.   :1.000000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##
##   prop_qtype_1     prop_qtype_2       prop_qtype_5       prop_qtype_6
##  Min.   :0.0000   Min.   :0.000000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:0.4920   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.000000
##  Median :0.6717   Median :0.000000   Median :0.000000   Median :0.000000
##  Mean   :0.6264   Mean   :0.007040   Mean   :0.000375   Mean   :0.048867
##  3rd Qu.:0.8035   3rd Qu.:0.006392   3rd Qu.:0.000000   3rd Qu.:0.003381
##  Max.   :1.0000   Max.   :0.291401   Max.   :0.050000   Max.   :1.000000
##
##  prop_qtype_10      prop_qtype_12       prop_qtype_15
##  Min.   :0.000000   Min.   :0.0000000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.0000000   1st Qu.:0.00000
##  Median :0.000000   Median :0.0000000   Median :0.00000
##  Mean   :0.000134   Mean   :0.0007137   Mean   :0.02216
##  3rd Qu.:0.000000   3rd Qu.:0.0000000   3rd Qu.:0.03390
##  Max.   :0.014925   Max.   :0.2500000   Max.   :1.00000
##
##  prop_qtype_16     prop_qtype_28    prop_qtype_33      prop_qtype_35
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.000000   Min.   :0.000e+00
##  1st Qu.:0.00000   1st Qu.:0.0400   1st Qu.:0.000000   1st Qu.:0.000e+00
##  Median :0.00000   Median :0.1200   Median :0.000000   Median :0.000e+00
##  Mean   :0.01108   Mean   :0.2248   Mean   :0.004531   Mean   :4.678e-05
##  3rd Qu.:0.01562   3rd Qu.:0.4691   3rd Qu.:0.003358   3rd Qu.:0.000e+00
##  Max.   :0.33613   Max.   :1.0000   Max.   :1.000000   Max.   :1.508e-02
##
##  prop_qtype_37       prop_qtype_38       prop_qtype_43
##  Min.   :0.000e+00   Min.   :0.000e+00   Min.   :0.000000
##  1st Qu.:0.000e+00   1st Qu.:0.000e+00   1st Qu.:0.000000
##  Median :0.000e+00   Median :0.000e+00   Median :0.000000
##  Mean   :2.932e-06   Mean   :5.691e-06   Mean   :0.037519
##  3rd Qu.:0.000e+00   3rd Qu.:0.000e+00   3rd Qu.:0.008138
##  Max.   :9.259e-03   Max.   :5.848e-03   Max.   :1.000000
##
##  prop_qtype_44       prop_qtype_46      prop_qtype_48
##  Min.   :0.000e+00   Min.   :0.00e+00   Min.   :0.000000
##  1st Qu.:0.000e+00   1st Qu.:0.00e+00   1st Qu.:0.000000
##  Median :0.000e+00   Median :0.00e+00   Median :0.000000
##  Mean   :1.718e-06   Mean   :2.02e-05   Mean   :0.007562
##  3rd Qu.:0.000e+00   3rd Qu.:0.00e+00   3rd Qu.:0.000000
##  Max.   :4.444e-03   Max.   :5.65e-03   Max.   :1.000000
##
##  prop_qtype_52       prop_qtype_99       prop_qtype_255
##  Min.   :0.000e+00   Min.   :0.000e+00   Min.   :0.0000000
##  1st Qu.:0.000e+00   1st Qu.:0.000e+00   1st Qu.:0.0000000
##  Median :0.000e+00   Median :0.000e+00   Median :0.0000000
##  Mean   :1.140e-06   Mean   :8.734e-05   Mean   :0.0008136
##  3rd Qu.:0.000e+00   3rd Qu.:0.000e+00   3rd Qu.:0.0000000
```

```
##  Max.   :4.167e-03   Max.   :2.632e-02   Max.   :0.1428570
##
##  prop_qtype_256 prop_qtype_257     prop_qtype_32769
##  Min.   :0      Min.   :0.000000   Min.   :0.000e+00
##  1st Qu.:0      1st Qu.:0.000000   1st Qu.:0.000e+00
##  Median :0      Median :0.000000   Median :0.000e+00
##  Mean   :0      Mean   :0.007825   Mean   :9.408e-06
##  3rd Qu.:0      3rd Qu.:0.000000   3rd Qu.:0.000e+00
##  Max.   :0      Max.   :1.000000   Max.   :8.264e-03
##
```

Classifiers

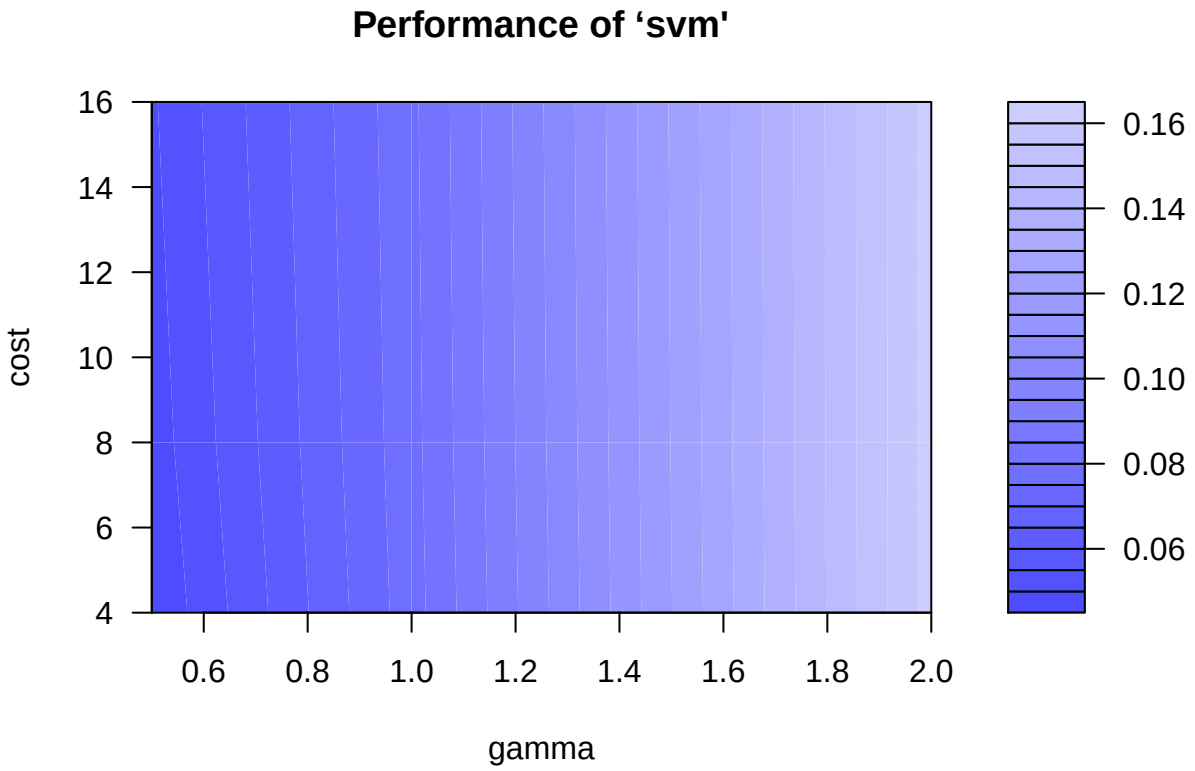
Multiclass Support Virtual Machine

We will use the e1071 package to train a multiclass SVM model based on our training data. We use the library defaults of C-classification and the RBF kernel, which exposes the hyperparameters γ and cost. We tune those parameters using a grid search.

```
svm_tuning <- tune(svm, origin ~ ., data = trainingData,
  ranges = list(gamma = 2^(-1:1), cost = 2^(2:4)),
  tunecontrol = tune.control(sampling = "fix"))
summary(svm_tuning)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: fixed training/validation set
##
## - best parameters:
##  gamma cost
##    0.5    4
##
## - best performance: 0.04560579
##
## - Detailed performance results:
##   gamma cost      error dispersion
## 1   0.5    4 0.04560579         NA
## 2   1.0    4 0.07770601         NA
## 3   2.0    4 0.16206694         NA
## 4   0.5    8 0.04736739         NA
## 5   1.0    8 0.07829321         NA
## 6   2.0    8 0.16206694         NA
## 7   0.5   16 0.04932472         NA
## 8   1.0   16 0.07888041         NA
## 9   2.0   16 0.16206694         NA
```

```
plot(svm_tuning)
```



Based on that tuning we select $\gamma = blah$ and $cost = 123$ and create a classifier accordingly for 10-fold cross-validation:

```
model_svm <- svm(origin ~ ., data = trainingData, gamma = 0.5, cost = 16, cross = 10)
summary(model_svm)
```

```
##
## Call:
## svm(formula = origin ~ ., data = trainingData, gamma = 0.5, cost = 16,
##     cross = 10)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  16
##       gamma:  0.5
##
## Number of Support Vectors:  6298
##
##  ( 2294 1088 2916 )
##
##
## Number of Classes:  3
##
## Levels:
##  facebook google other
##
```

```
## 10-fold cross-validation on training data:
##
## Total Accuracy: 95.09299
## Single Accuracies:
##  95.69191 94.52055 94.90862 95.95564 94.64752 94.19439 94.84334 95.30333 95.16971 95.69472
```

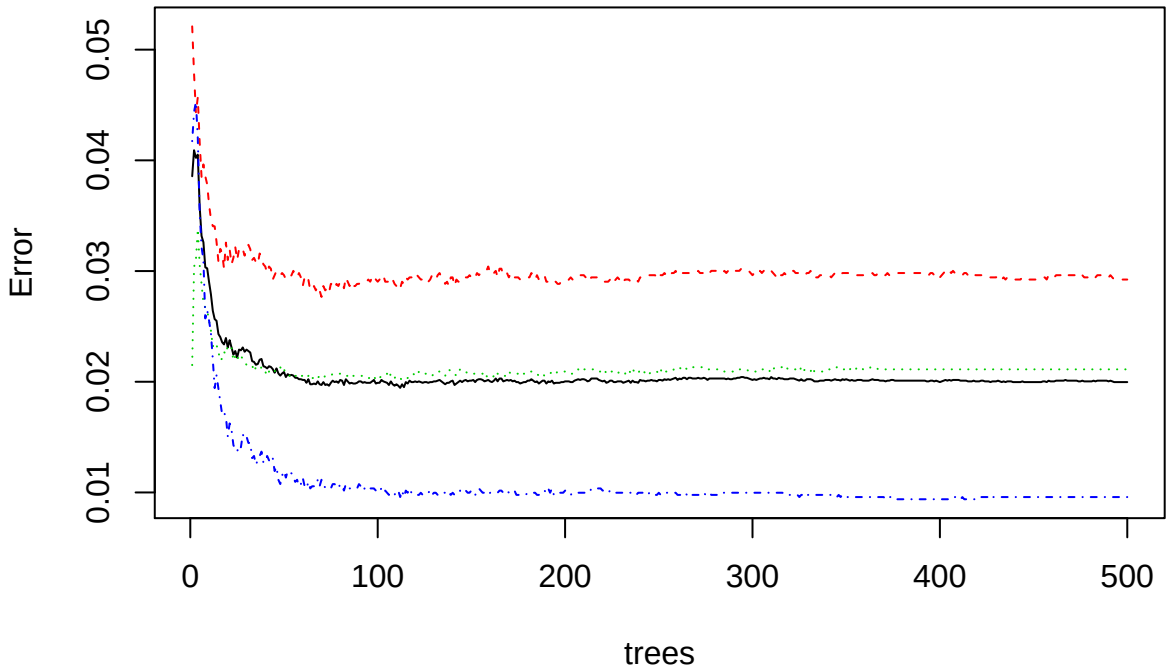
Random Forests

We use the randomForest package to train a multiclass classifier based on the same training data. We use the lbrary default parameters, since we have not been able to improve upon them. In particular the default number of trees seems more than sufficient.

```
model_rf <- randomForest(origin ~ ., trainingData)
model_rf
```

```
##
## Call:
##  randomForest(formula = origin ~ ., data = trainingData)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 2%
## Confusion matrix:
##          facebook google other class.error
## facebook     4948      4   145 0.029232882
## google         21   5006    87 0.021118498
## other          49      0  5065 0.009581541
```

```
plot(model_rf)
```



There is no need for cross-validation using Random Forests since an unbiased estimate of the test set error is generated internally during the construction of the model. Cross-validation's main function here is a guard against over-fitting, which Random Forests don't suffer from.

Model Testing

We use both models to classify traffic sources in the test dataset and assess their accuracy using a confusion matrix and a variety of calculated accuracy measures.

Multi-Class Support Vector Machine

```
prediction_svm  = predict(model_svm, testData)
confusionMatrix(data = prediction_svm, testData$origin)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction facebook google other
##   facebook     1607      4    35
##   google         36   1681    67
##   other          71     12  1595
##
## Overall Statistics
##
##                Accuracy : 0.956
##                  95% CI : (0.95, 0.9614)
##     No Information Rate : 0.3356
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9339
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: facebook Class: google Class: other
## Sensitivity                   0.9376        0.9906       0.9399
## Specificity                   0.9885        0.9698       0.9757
## Pos Pred Value                0.9763        0.9423       0.9505
## Neg Pred Value                0.9691        0.9952       0.9703
## Prevalence                    0.3356        0.3322       0.3322
## Detection Rate                0.3146        0.3291       0.3123
## Detection Prevalence          0.3222        0.3493       0.3285
## Balanced Accuracy             0.9630        0.9802       0.9578
```

Recall that:

$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

The accuracy of the classifier can hence be represented as follows:

$$A_{facebook} = \frac{1607 + (1681 + 67 + 12 + 1595)}{1607 + (4 + 35) + (1681 + 67 + 12 + 1595) + (36 + 71)} = 0.9714$$

$$A_{google} = \frac{1681 + (1607 + 35 + 71 + 1595)}{1681 + (4 + 12) + (1607 + 35 + 71 + 1595) + (4 + 12)} = 0.9936$$

$$A_{other} = \frac{1595 + (1607 + 4 + 36 + 1681)}{1595 + (71 + 12) + (1607 + 4 + 36 + 1681) + (35 + 67)} = 0.9653$$

Random Forests

```
prediction_rf = predict(model_rf, testData)
confusionMatrix(data = prediction_rf, testData$origin)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction facebook google other
##   facebook     1660      6    21
##   google          3   1674     2
##   other          51     17  1674
##
## Overall Statistics
##
##                Accuracy : 0.9804
##                  95% CI : (0.9762, 0.984)
##     No Information Rate : 0.3356
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9706
```

```
##  Mcnemar's Test P-Value : 1.31e-05
##
## Statistics by Class:
##
##                      Class: facebook Class: google Class: other
## Sensitivity                   0.9685        0.9864       0.9864
## Specificity                   0.9920        0.9985       0.9801
## Pos Pred Value                0.9840        0.9970       0.9610
## Neg Pred Value                0.9842        0.9933       0.9932
## Prevalence                    0.3356        0.3322       0.3322
## Detection Rate                0.3250        0.3277       0.3277
## Detection Prevalence          0.3303        0.3287       0.3410
## Balanced Accuracy             0.9803        0.9925       0.9833
```

Again, recalling:

$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

We obtain:

$$A_{facebook} = \frac{1660 + (1674 + 2 + 17 + 1674)}{1660 + (6 + 21) + (1674 + 2 + 17 + 1674) + (3 + 51)} = 0.9841$$

$$A_{google} = \frac{1674 + (1660 + 21 + 51 + 1674)}{1674 + (3 + 2) + (1660 + 21 + 51 + 1674) + (6 + 17)} = 0.9945$$

$$A_{other} = \frac{1674 + (1660 + 6 + 3 + 1674)}{1674 + (51 + 17) + (1660 + 6 + 3 + 1674) + (21 + 2)} = 0.9822$$