

Contents

[Introduction](#_30j0zll)

Problem Description [1](#_30j0zll)

[Project page: 2](#_1fob9te)

[Libraries 3](#_3znysh7)

[Error logging 3](#_2et92p0)

[Data Structure 4](#_tyjcwt)

[Data Cleaning, Selection, and Correlation Analysis 5](#_3dy6vkm)

[Transformation: Conversion Profiles 5](#_1t3h5sf)

[Oversampling 10](#_4d34og8)

[Correlation Matrix 11](#_2s8eyo1)

[Algorithm 13](#_17dp8vu)

[Early tabulation using StepAIC 13](#_3rdcrjn)

[newCSVs.R 18](#_26in1rg)

[Example plots from V8517 25](#_35nkun2)

[Actual Console Output 30](#_1ksv4uv)

# Executive Summary

# 

# 

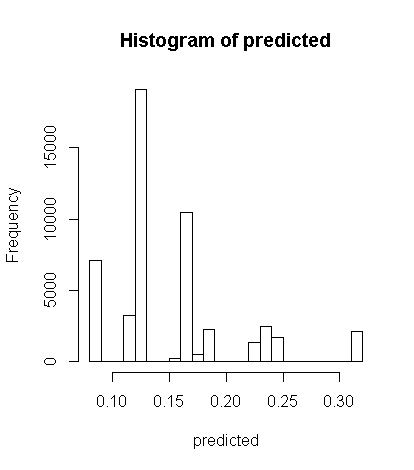
# 

# 

# 

# Introduction

# The American Public Health Association (APHA) needs to evaluate the behavior trends of drug use among adolescents annually. This research will provide resources for APHA to review substance abuse trends and study the impact of new trend (or new substance) among adolescents. Our data consists of poll research data from 2012-2017 from the Inter-University Consortium for Political and Social Research (ICPSR) and covers data from the 8th to 10th grade youth. The name of the study is titled “Monitoring the Future: A Continuing Study of American Youth (8th- and 10th-Grade Surveys), [Year]. The overall goal of this project is to identify the relationship between any lifetime use of various drugs to education, geographical location, mental and physical health, happiness, parental background, and the presence of deviant behaviors.

Our vision was to split the data into binary classes, to include the independent terms. This is what we term the “semiotic grid” since it consists of only 1’s and 0’s (presence or absence of something), the resulting factorial combinations are binary. I.e. 2 factors is a total of 4 combinations, 3 is 8, 4 is 16 and so on. 

Aside from this simple list of 1’s and 0’s. The model can be thought of as an easier to read and interpret binary logistic regression model which makes use of various sampling techniques to achieve model robustness.

# Problem Statement

The three questions which we intended to solve through this project are given below:

* What factors influences being included in above or equal to median value of GPA? (B+)
* What factors influence having a history of Gang Violence?
* What factors influence lifetime presence of psychedelic drug use?

# Project page:

<https://github.com/thistleknot/Capstone-577>

Our project was tracked on github for version control as well as accessibility purposes. The project has seen many evolutions and is over 400 commits. The readme, which is available online, details the projects evolutions over each milestone.

The files we based our final analysis on (cleandatacode.R’s: \*final.csv’s) were based on commit ca9100c. The steps to follow to run the code is as follows:

* Modify in cleandatacode.R
  + sourceDir to point to where you downloaded and extracted the sourcefolder to. This should be the same folder that output sits in.
* Run cleandatacode.R which starts the simulation up until the point where it starts to output tabulation results.
  + Then stop the simulation
    - You can let it run all the way through, but the files necessary to run the rest of the simulation (\*final.csv) are already in the output folder.
* Run saveCSVs.R and allow it to finish
* Run 4thpass.R

The system consists of 3 different main files.

* Cleandatacode.R
  + Derives final.csv which stores cross validated thresholds,
  + does factor reduction.
* saveCSVs.R
  + Ensures factor reduced terms work well with each other using Cross Validation and derives final terms
* 4thpass.R
  + Derive optimal class confusion matrix
  + ROC plots
  + Check VIF for collinearity

Other files related to cleaning, transforming, indexing, class balancing, and aggregating the data into dataframes are (and in their proper order of being called). All these files are assumed to be sourced from cleandatacode.R

* **newDF.R**
* **reseedboth.R**
* **reseedTest.R**
* **reseedTrain.R**
* **MCResampleTest.R**
* **MCResampleTrain.R**
* **redrawTrain.R**
* **redrawTest.R**

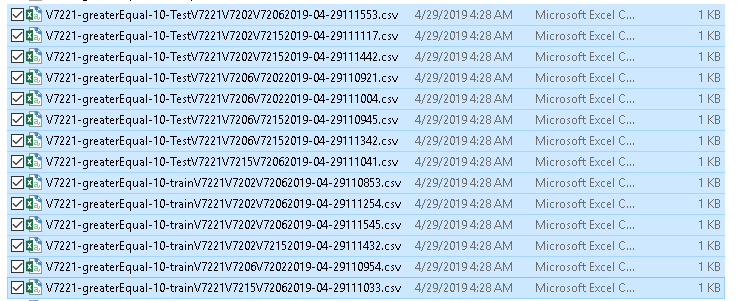
# Libraries

* dplyr
* plyr
* RPostgreSQL
* ggplot2
* anchors
* caret
* corrplot
* MASS
* car
* leaps
* bestglm
* compare
* R.utils
* tidyr
* stringr

# Error logging

During our project, we ran into various errors so we made sure we logged them. One error thrown by bestglm was hard to properly log in a tryCatch block so we resorted to outputting to csv to be sure the factors included in the final output wouldn’t’ have been affected as well as to inspect the curious dataset further at the population level if we so desired.





# 

# Data Loading

In this process we integrated all the received and combined them into a single dataset. The details of the raw datasets are as follows -

* Dataset for 2012 - 31106 records and 552 Variables.
* Dataset for 2013 - 28495 records and 550 variables.
* Dataset for 2014 - 28536 records and 563 variables
* Dataset for 2015 - 31162 records and 577 variables
* Dataset for 2016 - 32873 records and 571 variables
* Dataset for 2017 - 30181 records and 581 variables



We used the process of ETL(Extract, Transform, Load) by copying the data from the sources into a destination system which represented the data differently from the source. The original data was in SPSS format, which was finally converted to CSV’s. Furthermore, we transformed the extracted data using R.

# Data Cleaning

While exploring the combined data, we came across a lot of N/A values in the combined dataset. Also, we found a mixture of both numerical and categorical variables. The categorical variables are converted into numerical variables. There are 618 factors left after combining the datasets. We parsed through these factors and focus mainly on behavioral, habitual, and environmental factors.

The columns represent

* factor id
* conversion profile
* description
* category flag (no longer used aside from excluding 8’s as non response terms and 0 as response and all else as x’s)

To address the issue with NA’s we took two approaches. One we dealt with NA’s at a higher level when we class balanced the variables (which requires cleaning, which requires the combination of columns presented at the time of na cleaning). So we modified the algorithm to work with only 2 columns at a time which required us to rewrite the entire indexing structure of the dataframes.

## Conversion of Profiles

The transformation of the data occurs in NewDF.R where we transform according to 3 conversion profiles.

* convert1Index
  + Simple yes/no
* convert2Index
  + Median based
  + One deciding constant.
    - medianDirection
    - controls operant condition: >= or > median.
* convert3Index
  + split around a positive response (i.e. response was classed as 4+: mostly or higher).

It should be noted that these profiles convert to -1 for what will become 0 eventually. This is because we use 0 at this juncture to hold NA’s and then drop NA’s at the last possible level (after generating index’s for which columns we are going to include in a partition which is what pairedLists.R’s output is for, column indexes)

Within NewDF.R the basic logic for conversion is as follows

Pre conversion

Anything below 0 is considered NA which we code as 0.

*Convert1Index*

Simply binary presence (yes/no)



Rules

*Convert2Index*

Using a measure of central tendency, we converted to 1 if >= Median. The median is a common bifurcator in classification.

Our concern/hope was that there was no increasing or decreasing trend so we could state that the binary logistic regression equation could be inferred to apply to years beyond the dataset, but to be able to make that statement, we needed to check each year and derive confidence intervals for the median. The below is the code to check the confidence intervals for the median.

The medians were confirmed the same for each year. For V7563 in 2012 the median’s 95% upper confidence interval was 3 vs 2. But for our purposes, we assume the median is accurate for each year and remains as such and a population resample every 5 years would be in order to confirm the median remained the same.

**V7221,2,"R HS GRADE/D=1",0**

**Median: #7: B+ 95% conf interval confirmed**

**V7215,2,"FATHR EDUC LEVEL",0**

**#5: for college grad father, 95% conf confirmed**

**V7551,2,"#HR/W INTERNET S",0**

**#4: 3-5 Hours Internet #95% conf confirmed**

**V7552,2,"DALY WEB FACEBK",0**

**#5: 6-9 Hours Facebook # 95% conf confirmed**

**V7553,2,"#HR GAMING",0**

**#4 3-5 Hours Gaming # 95% conf confirmed**

**V7562,2,"#HR TEXT",0**

**#4 3-5 Hours Texting # 95% conf confirmed**

**V7563,2,"#HR TALK CELL",0**

**#2: <1 Hour talking on cell phone # 95% conf confirmed**



Rules

*Convert3Index*



Rules

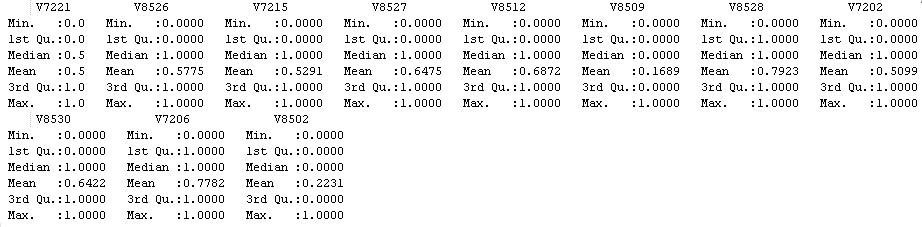
**

***Hypothesis***

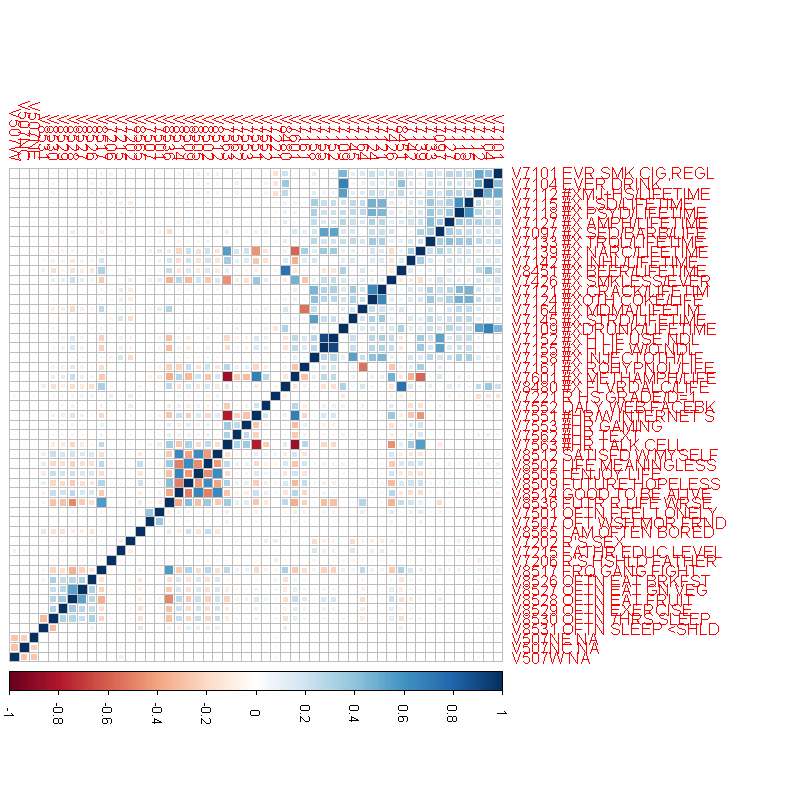
* **H0 sum(Co-efficients) = 0**
* **H1 sum(Co-efficients) > 0**
  + **Find robust subset**
  + **We set to show that H1 is true using cross validation and simulation to converge on significant terms using monte carlo class balanced resampling**

# Oversampling

We followed the procedure to size the set of 0’s to be equal to the set of 1’s. This is over/undersampling. We tried to balance the size of the sample to reflect the average of the number of rows between ones and zeros but the large number of 0’s in the data created errors when the sample size exceeded the population. We are reconstructing a distribution to fit a specific outcome response. When we look at the output of one such constructed dataset, we can immediately see population proportion of the other variables. (Shmueli, 130). You still randomly sample from the non-desired class at an equal amount as the desired class.



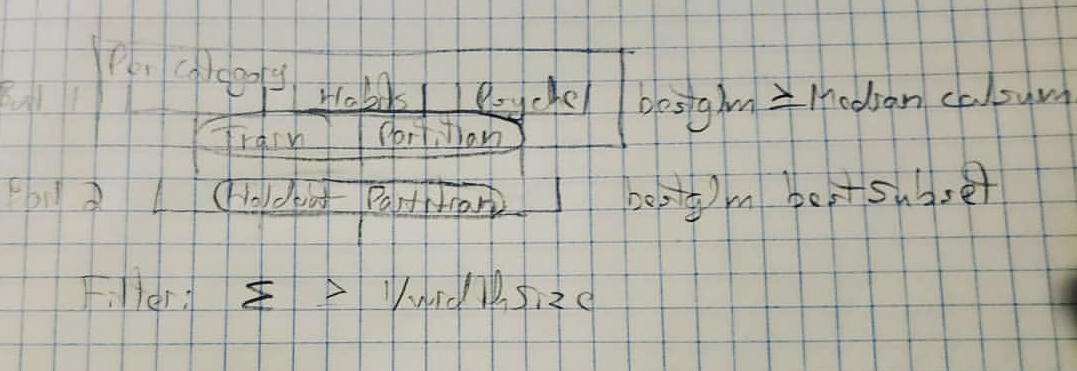
# Correlation Matrix

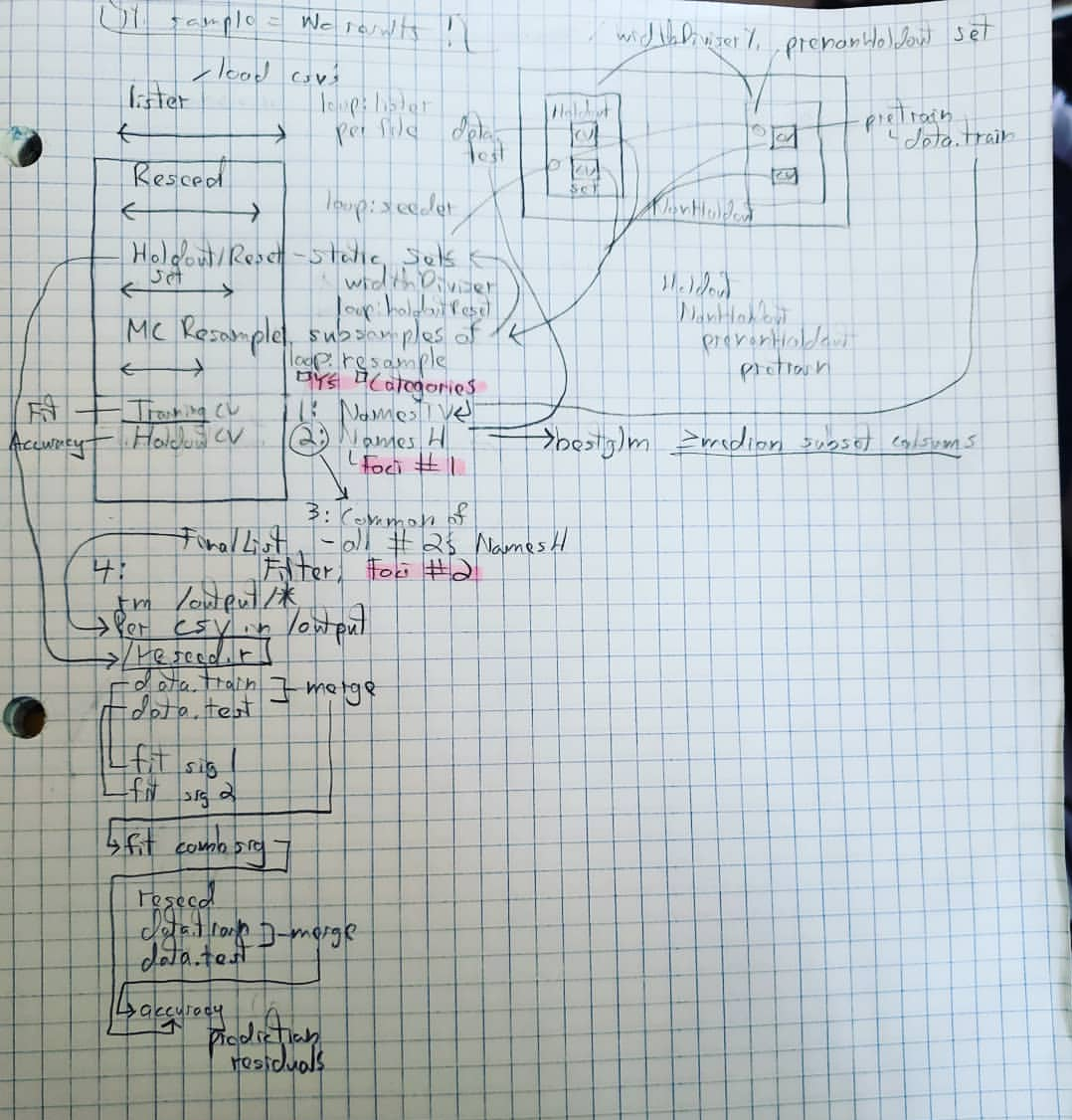


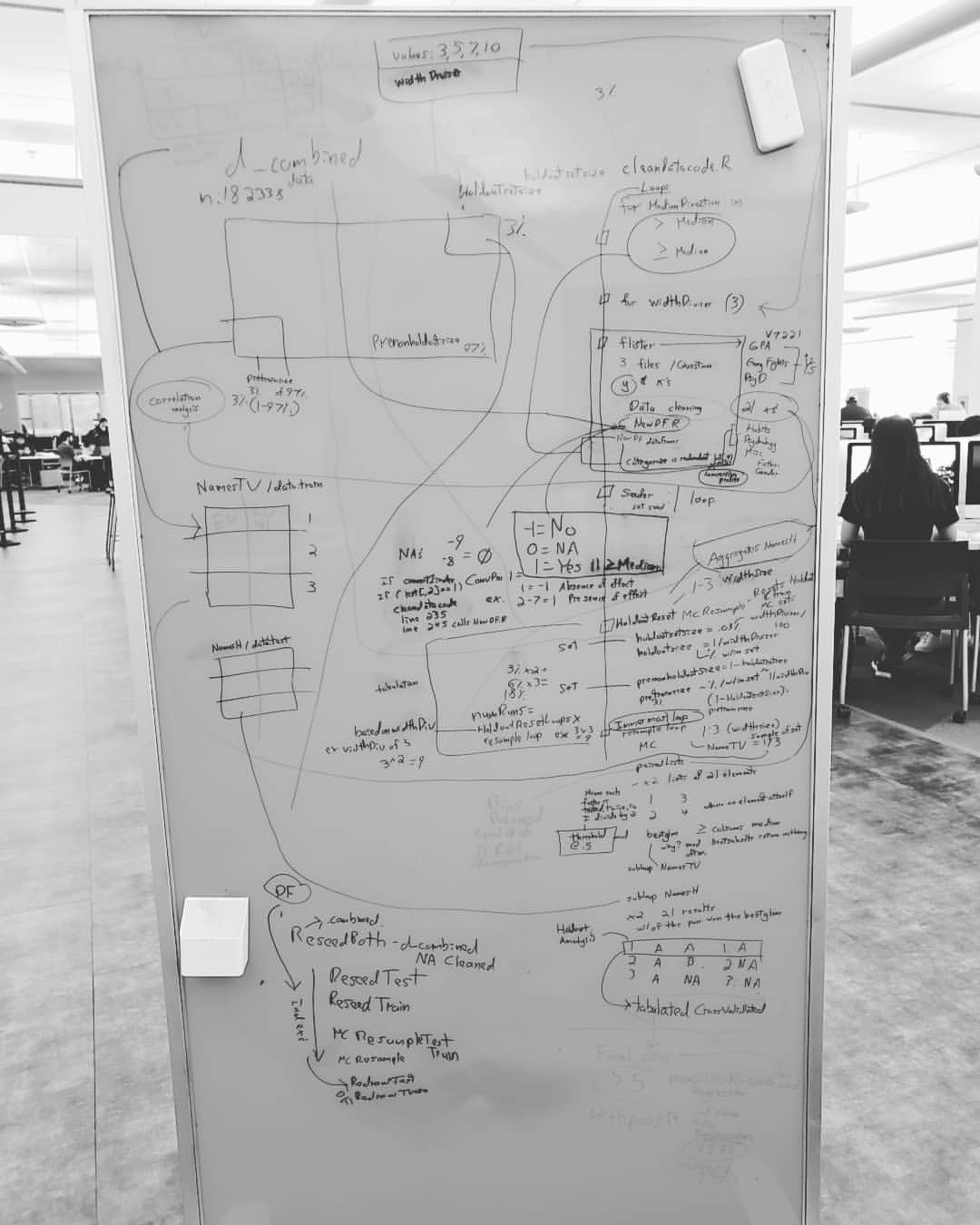
This is one of our earliest plots and the na’s were merely coded as 0. This plot was made to see what factors were relevant and how we should split up our search space. For example, V8512 through V8546 were seen to be highly correlated with each other and therefore were treated as a group to be included or excluded or pruned in pairs (psychology factors). V8517 through V8531 were grouped as Health. V7501 through V7206 were grouped as one-off’s. V7551 through V7563 were grouped as habits. The substances all clearly were correlated with each other and we considered aggregating them and to visually group into categories and used tabulation to do final determination of a good response term y. We referred to this chart to consider which class of variables to drop once we did decide on which to include (in this case our substances, V7101 to V8480). We dropped V7501 and V7507 later during development when we ran into an issue with bestglm not converging due to NA’s decimating records even with just 6 columns with those included as we saw that these columns had data with over 75% NA’s.

We understand the algorithm for checking for multicollinearity is to compare any pairwise x’s and see if they are greater than response y. (Shmueli, 206). However, we group similar terms together visually based on the correlation plot as well as to exclude group of variables. We then allowed the tabulation to do the filtering as we understood our goal was to find the best combination of factors and terms will initially be collinear going into the filtering algorithm (whether stepAIC or bestglm).

# Algorithm

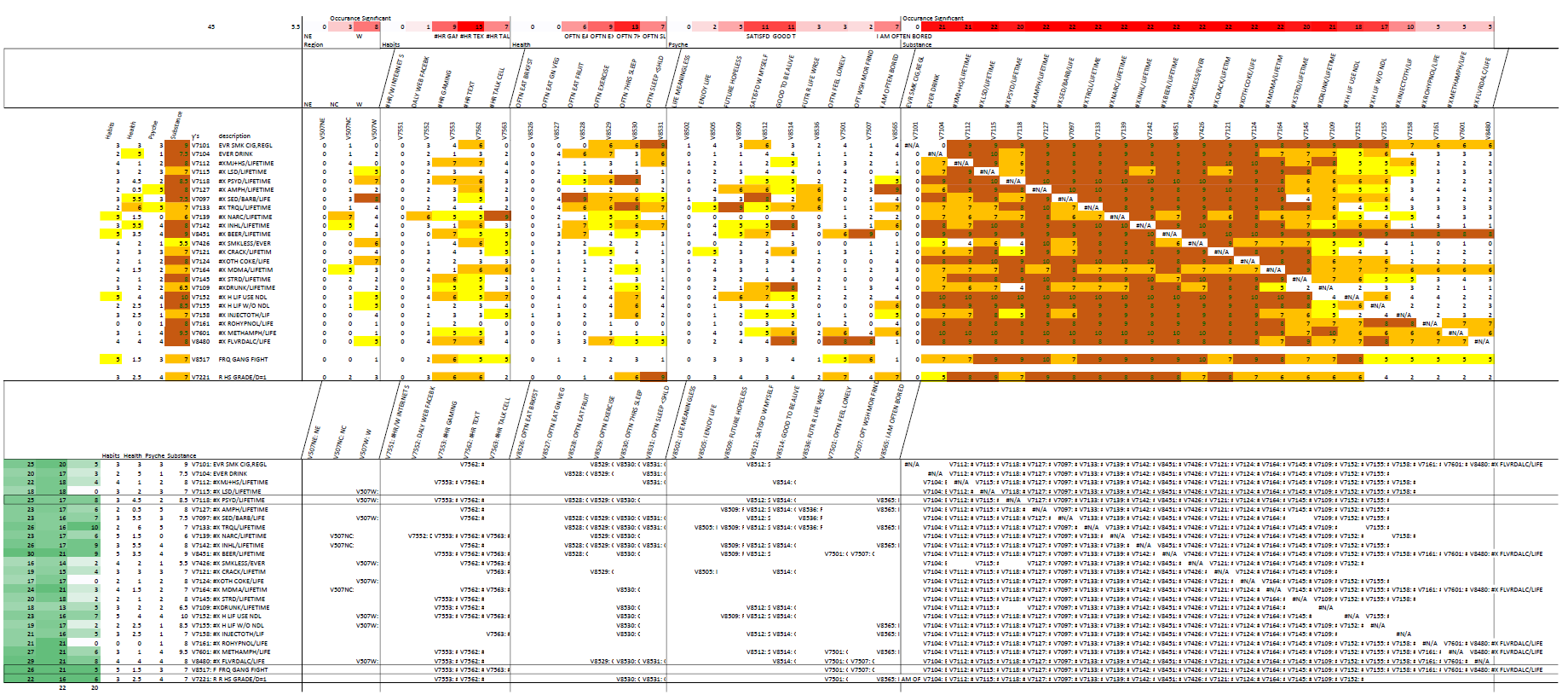






# Regression

The find the regression equation we first wanted to use the stepAIC of 10 runs. We dropped geographical categorical variables (which we hand coded) for V507NE, V507NC,and V507W as they were not found relevant under any cross tabulation results. We also focused on only one substance, pscyhadelic drug use based on the cross tabulated results.



So we were going to let whatever filtering algorithm we chose do the tabulation for us and decide on a threshold after viewing a few runs at various settings. a tabulation over data.train (training data) using cross validation and produce plausible terms understanding they will be collinear. We initially tried stepAIC then switched to bestglm.

We modified bestglm to expand on the results normally offered and have provided the code.

Instead of running the algorithm over the whole data, we did subsampling at two levels to arrive at two non repeated subsamples, one is a small sample percentage (10% in our analysis) of our population taken twice. The two sets are the holdout set and training/validation set and derives from the Carl Jung idea of the collective unconscious (the unknown information space from which we draw our limited samples from). These sets are monte carlo resamples and re initialized on our holdoutReset loops. The sets are resampled with replacement but the partitions are distinct from each other ensuring proper holdout analysis. This is to randomize and create more information space (variance) as well as mimic measures of central tendency of the desired observed class as well as meet the assignment simulation requirement.

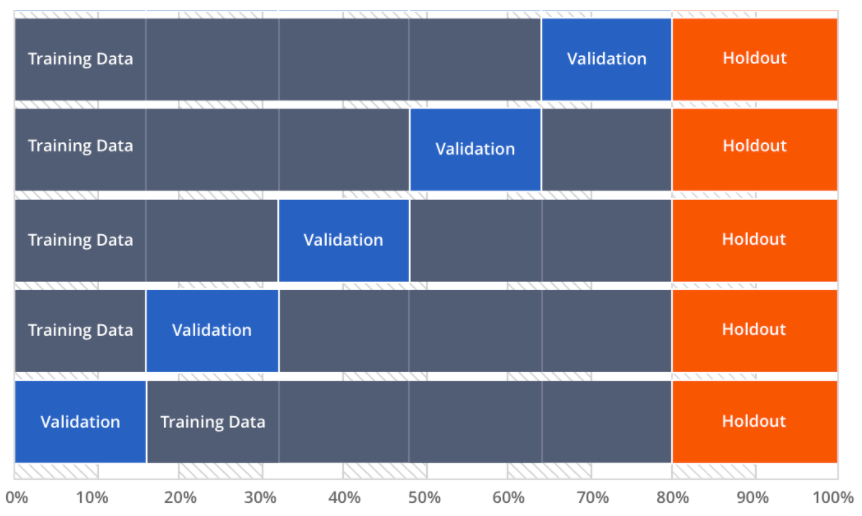
During our holdout loop, we [re]-initialize our holdoutSet. The holdoutSet.ones is the holdout set of ones used for testing from which we resample data.test from during our holdout reset loop. From within that loop we call reseedtrain. Then we ration off the prenonHoldoutset by simply delesecting the holdoutset via “availableset.prenonholdout.ones <- rangeOnes[-c(set.holdoutSet.ones)]”. From within this prenonHouldset we draw our pretrain set of indexes from inside our resample loop. Within the holdoutSet exists our holdout [set] (confusing I know) indexes. These two set of indexes are used for data.train (pretrain) and data.test (holdout) within the resample loop.

Within the inner most loop, we redraw (redrawtest.R) the holdout dataframe (data.test) from holdout. The outer [macro] loop holdoutreset monte carlo resets the Holdoutset at 10% each (10% for holdoutset and 10% for pretrain which is drawn from the prenonholdoutset set) 10 times. Then an inner resample loop monte carlo resamples from these sets (holdout and pretrain) at 33% over 3 loops creating data.test

(from holdout) and data.train (from pretrain). Both loops combined (10 x 3) makes for 30 runs (numRuns) total. The output only displays the tabulated results at the end of the holdoutreset loop.

We do this nested sampling to achieve two things. First is to mimic not seeing all the data, and second is to achieve monte carlo resampling techniques which help achieve more robust terms, but also while maintaining proper holdout analysis allows us to resample almost continuously for finer results.

Data.train makes up the training/validation you see below and data.test is the holdout. However, the holdout does its own validation akin to how you see the training doing it. This is because we treat both partitions the same and simply run the same created set known as pairedlist against both partitions and then compare them side by side record by record and note which factors passed both partitions. Each partition produces it’s own list, NamesTV and NameH respectively for data.train and data.test. These lists are the results of the pairedLists (factors) passes. These two lists are compared side by side record by record and are considered holdout filtered if the same pair survived both partitions which are then tabulated each resample pass. Each holdout pass these tabulations are outputted.



Example of pairedLists.R using a size of 3 which is used for column indexing, offers unique non repeating pairs (i.e. no pairs of the same #).

pool1 set

picked 2 3

picked.1 1 2

picked.2 3 1

As a result of pairedlists.R creating two distinct paired lists, the initial list of factors are tested twice which results in the chance that the factor could score twice each run. So each resample run, we divide this summation by 2. This results in more NA’s (we expect lots of NA’s, each paired list can result in 2 NA’s, something like that, for our analysis, it’s not essential, we merely note the NA’s show at a greater quantity than the factors of interest). This results in a threshold of 1 as the max and .5 as the factor showing up at least once between the pairedList. At the end of all the holdout passes, a final.csv is printed with a descending order of the scores stored as Freq.

Sample output of a final.csv’s tabulated results.



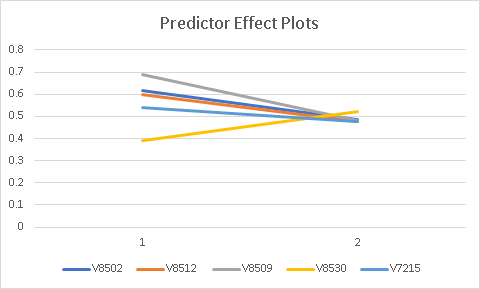
We set our final threshold for >.25. Which means the factor shows up more than every other run at a minimum (i.e. > .25).

Then use bestglm’s actual subset filter to arrive at real Cross Validated result using these tabulated (threshold cutoff filtered) terms during a subsequent pass on a monte carlo aggregated population class balanced dataframe (newCSVs.R) to find always terms that will always converge as significant at the population level. We then compare model parameter’s derived from the bestglm output from the class balanced monte carlo partition we created against the population’s best fit using train.control’s CV of those terms.

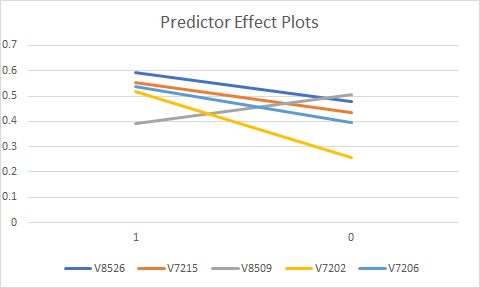
During 4thpass.R, we then use those reduced terms against the population, derive ROC plots, optimal cutoff thresholds for an outcome of 1 or 0, and derive classification matrix’ accordingly.

# Logistic Regression Equations

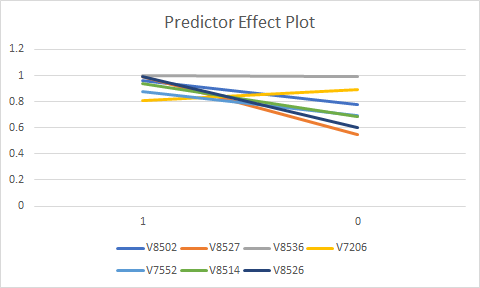
V7118=exp(-0.2801+O.5424\*V8502+0.5023+V8512+0.8561\*V8509-0.5247\*V8530+0.2496\*V7215)/1+exp(-0.2801+O.5424\*V8502+0.5023+V8512+0.8561\*V8509-0.5247\*V8530+0.2496\*V7215)



V7221=exp(-1.7364+0.4629\*V8526+0.4801\*V7215-0.4573\*V8509+1.1307\*V7202+0.5737\*V7206)/1+exp(-1.7364+0.4629\*V8526+0.4801\*V7215-0.4573\*V8509+1.1307\*V7202+0.5737\*V7206)



V8517=exp(-2.7165+2.0157\*V8502+5.3567\*V8527+4.9321\*V8536-O.7356\*V7206+1.1383\*V7552+1.8867\*V8514+4.9424\*V8526)/1+exp(-1.7364+0.4629\*V8526+0.4801\*V7215-0.4573\*V8509+1.1307\*V7202+0.5737\*V7206)



[1] "C:/Users/user/Documents/School/CSUF/ISDS577/projects/Capstone-577//output/V7118-greaterEqual-10-final.csv"

[1] "y:" "V7118"

[[1]]

[1] "final: "

$tabulatedCrossValidated

[1] <NA> V8502 V8512 V8509 V8514 V8530 V8505 V7215 V8536 V8526 V7206 V7202 V8528 V7562 V7552 V8529 V8531 V8527 V8565 V7553 V7563 V7551

Levels: V7202 V7206 V7215 V7551 V7552 V7553 V7562 V7563 V8502 V8505 V8509 V8512 V8514 V8526 V8527 V8528 V8529 V8530 V8531 V8536 V8565

$Freq

[1] 3.517 0.433 0.383 0.350 0.350 0.333 0.317 0.267 0.267 0.233 0.217 0.200 0.150 0.133 0.117 0.117 0.117 0.100 0.100 0.083 0.083 0.067

tabulatedCrossValidated Freq

2 V8502 0.433

3 V8512 0.383

4 V8509 0.350

5 V8514 0.350

6 V8530 0.333

7 V8505 0.317

8 V7215 0.267

9 V8536 0.267

10 V8526 0.233

11 V7206 0.217

12 V7202 0.200

13 V8528 0.150

14 V7562 0.133

15 V7552 0.117

16 V8529 0.117

17 V8531 0.117

18 V8527 0.100

19 V8565 0.100

20 V7553 0.083

21 V7563 0.083

22 V7551 0.067

[1] "keep: > " "0.25" "8" "V8502" "V8512" "V8509" "V8514" "V8530" "V8505" "V7215" "V8536"

V8502 V8512 V8509 V8530 V7215

0.5423997 0.5023144 0.8560780 -0.5246812 0.2495730

V7118 V8502 V8512 V8509 V8514 V8530 V8505 V7215 V8536

Min. :0.0 Min. :0.0000 Min. :0.0000 Min. :0.00000 Min. :0.00 Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.00000

1st Qu.:0.0 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.00 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.00000

Median :0.5 Median :0.0000 Median :0.0000 Median :0.00000 Median :0.00 Median :0.0000 Median :0.0000 Median :0.0000 Median :0.00000

Mean :0.5 Mean :0.1299 Mean :0.2203 Mean :0.09739 Mean :0.25 Mean :0.1432 Mean :0.2289 Mean :0.4157 Mean :0.02048

3rd Qu.:1.0 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:0.00000 3rd Qu.:0.25 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:0.00000

Max. :1.0 Max. :1.0000 Max. :1.0000 Max. :1.00000 Max. :1.00 Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.00000

Call:

imcdiag(x = x, y = y, method = method, corr = FALSE, vif = vif,

tol = tol, conf = conf, cvif = cvif, leamer = leamer, all = all)

VIF Multicollinearity Diagnostics

VIF detection

V8502 1.6330 0

V8512 3.4039 0

V8509 1.6528 0

V8514 3.8443 0

V8530 1.2904 0

V8505 3.4856 0

V7215 1.0190 0

V8536 1.0749 0

NOTE: VIF Method Failed to detect multicollinearity

0 --> COLLINEARITY is not detected by the test

===================================

[1] "MC summary"

Call:

glm(formula = y ~ ., family = family, data = data.frame(Xy[,

c(bestset[-1], FALSE), drop = FALSE], y = y))

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0068 -1.0610 -0.1622 1.1904 1.5326

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.28009 0.04102 -6.828 8.58e-12 \*\*\*

V8502 0.54240 0.11385 4.764 1.90e-06 \*\*\*

V8512 0.50231 0.08016 6.266 3.69e-10 \*\*\*

V8509 0.85608 0.13398 6.390 1.66e-10 \*\*\*

V8530 -0.52468 0.09441 -5.558 2.73e-08 \*\*\*

V7215 0.24957 0.05926 4.212 2.53e-05 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 6903.7 on 4979 degrees of freedom

Residual deviance: 6664.0 on 4974 degrees of freedom

AIC: 6676

Number of Fisher Scoring iterations: 4

[1] "optCutOff\_sens:" "0"

[1] "error rate sens: 0.5"

[1] "yhat.transformed\_center sens matrix"

[1] "n:" "4980"

1

0 0.5

1 0.5

[1] "optCutOff\_center" "1"

[1] "error rate c: 0.4159"

[1] "yhat.transformed\_center conf matrix"

[1] "n:" "4980"

0 1

0 0.4257 0.0743

1 0.3416 0.1584

[1] "optCutOff\_spec" "0.01"

[1] "error rate spec: 0.4159"

[1] "yhat.transformed\_spec conf matrix"

[1] "n:" "4980"

0 1

0 0.4257 0.0743

1 0.3416 0.1584

[1] "MC model applied to Pop :" "0.634457431104765"

[1] "Pop model applied to pop :" "0.646058282469799"

[1] "CV Model applied to population"

Call:

glm(formula = holderOfData, family = binomial(link = "logit"))

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0546 -0.3351 -0.3020 -0.3020 2.4941

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -3.06467 0.08879 -34.517 < 2e-16 \*\*\*

V8512 0.77495 0.12557 6.172 6.76e-10 \*\*\*

V8509 3.80169 0.12489 30.441 < 2e-16 \*\*\*

V8530 0.25677 0.14231 1.804 0.0712 .

V7215 0.21286 0.10867 1.959 0.0502 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 3846.6 on 4978 degrees of freedom

Residual deviance: 2570.2 on 4974 degrees of freedom

AIC: 2580.2

Number of Fisher Scoring iterations: 5

V7118 V8502 V8512 V8509 V8530 V7215

Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.0000

1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.0000

Median :0.0000 Median :0.0000 Median :0.0000 Median :0.00000 Median :0.0000 Median :0.0000

Mean :0.4508 Mean :0.1234 Mean :0.2038 Mean :0.09284 Mean :0.1352 Mean :0.4111

3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:0.00000 3rd Qu.:0.0000 3rd Qu.:1.0000

Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.00000 Max. :1.0000 Max. :1.0000

[1] "C:/Users/user/Documents/School/CSUF/ISDS577/projects/Capstone-577//output/V7221-greaterEqual-10-final.csv"

[1] "y:" "V7221"

[[1]]

[1] "final: "

$tabulatedCrossValidated

[1] <NA> V8526 V7215 V8527 V8512 V8509 V8528 V7202 V8530 V7206 V8502 V8505 V8514 V8529 V7562 V8565 V7553 V7551 V7552 V8536 V7563

Levels: V7202 V7206 V7215 V7551 V7552 V7553 V7562 V7563 V8502 V8505 V8509 V8512 V8514 V8526 V8527 V8528 V8529 V8530 V8536 V8565

$Freq

[1] 2.067 0.600 0.567 0.550 0.450 0.433 0.417 0.367 0.367 0.350 0.317 0.250 0.250 0.250 0.233 0.183 0.083 0.050 0.050 0.050 0.017

tabulatedCrossValidated Freq

2 V8526 0.600

3 V7215 0.567

4 V8527 0.550

5 V8512 0.450

6 V8509 0.433

7 V8528 0.417

8 V7202 0.367

9 V8530 0.367

10 V7206 0.350

11 V8502 0.317

12 V8505 0.250

13 V8514 0.250

14 V8529 0.250

15 V7562 0.233

16 V8565 0.183

17 V7553 0.083

18 V7551 0.050

19 V7552 0.050

20 V8536 0.050

21 V7563 0.017

[1] "keep: > " "0.25" "10" "V8526" "V7215" "V8527" "V8512" "V8509" "V8528" "V7202" "V8530" "V7206" "V8502"

V8526 V7215 V8509 V7202 V7206

0.4628940 0.4800828 -0.4572628 1.1307374 0.5737264

V7221 V8526 V7215 V8527 V8512 V8509 V8528 V7202 V8530 V7206 V8502

Min. :0.0 Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000

1st Qu.:0.0 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000

Median :0.5 Median :0.0000 Median :1.0000 Median :0.0000 Median :0.0000 Median :0.00000 Median :0.0000 Median :1.0000 Median :0.0000 Median :1.0000 Median :0.0000

Mean :0.5 Mean :0.1468 Mean :0.5186 Mean :0.1816 Mean :0.2809 Mean :0.08137 Mean :0.2247 Mean :0.9141 Mean :0.1683 Mean :0.7074 Mean :0.1146

3rd Qu.:1.0 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:0.00000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:0.0000

Max. :1.0 Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000

Call:

imcdiag(x = x, y = y, method = method, corr = FALSE, vif = vif,

tol = tol, conf = conf, cvif = cvif, leamer = leamer, all = all)

VIF Multicollinearity Diagnostics

VIF detection

V8526 1.9812 0

V7215 1.0284 0

V8527 2.8743 0

V8512 1.3416 0

V8509 1.4923 0

V8528 3.3344 0

V7202 1.1240 0

V8530 2.0541 0

V7206 1.1066 0

V8502 1.5148 0

NOTE: VIF Method Failed to detect multicollinearity

0 --> COLLINEARITY is not detected by the test

===================================

[1] "MC summary"

Call:

glm(formula = y ~ ., family = family, data = data.frame(Xy[,

c(bestset[-1], FALSE), drop = FALSE], y = y))

Deviance Residuals:

Min 1Q Median 3Q Max

-1.5805 -1.1639 0.1812 1.0379 2.1445

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.73639 0.03818 -45.48 <2e-16 \*\*\*

V8526 0.46289 0.02544 18.20 <2e-16 \*\*\*

V7215 0.48008 0.01775 27.04 <2e-16 \*\*\*

V8509 -0.45726 0.03263 -14.02 <2e-16 \*\*\*

V7202 1.13074 0.03901 28.99 <2e-16 \*\*\*

V7206 0.57373 0.02030 28.26 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 76634 on 55279 degrees of freedom

Residual deviance: 72472 on 55274 degrees of freedom

AIC: 72484

Number of Fisher Scoring iterations: 4

[1] "optCutOff\_sens:" "0"

[1] "error rate sens: 0.5"

[1] "yhat.transformed\_center sens matrix"

[1] "n:" "55280"

1

0 0.5

1 0.5

[1] "optCutOff\_center" "1"

[1] "error rate c: 0.399"

[1] "yhat.transformed\_center conf matrix"

[1] "n:" "55280"

0 1

0 0.3482 0.1518

1 0.2472 0.2528

[1] "optCutOff\_spec" "0.01"

[1] "error rate spec: 0.399"

[1] "yhat.transformed\_spec conf matrix"

[1] "n:" "55280"

0 1

0 0.3482 0.1518

1 0.2472 0.2528

[1] "MC model applied to Pop :" "0.64510140728372"

[1] "Pop model applied to pop :" "0.760487828258346"

[1] "CV Model applied to population"

Call:

glm(formula = holderOfData, family = binomial(link = "logit"))

Deviance Residuals:

Min 1Q Median 3Q Max

-0.8397 -0.6062 -0.5256 -0.4543 2.2228

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.38210 0.04933 -48.292 < 2e-16 \*\*\*

V7215 0.30891 0.02471 12.503 < 2e-16 \*\*\*

V8509 0.73996 0.03699 20.004 < 2e-16 \*\*\*

V7202 0.16294 0.05126 3.179 0.00148 \*\*

V7206 0.30926 0.02928 10.561 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 46112 on 55278 degrees of freedom

Residual deviance: 45406 on 55274 degrees of freedom

AIC: 45416

Number of Fisher Scoring iterations: 4

V7221 V8526 V7215 V8509 V7202 V7206

Min. :0.0000 Min. :0.0000 Min. :0.000 Min. :0.00000 Min. :0.0000 Min. :0.0000

1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.000 1st Qu.:0.00000 1st Qu.:1.0000 1st Qu.:0.0000

Median :1.0000 Median :0.0000 Median :1.000 Median :0.00000 Median :1.0000 Median :1.0000

Mean :0.5783 Mean :0.1506 Mean :0.533 Mean :0.07944 Mean :0.9209 Mean :0.7226

3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1.000 3rd Qu.:0.00000 3rd Qu.:1.0000 3rd Qu.:1.0000

Max. :1.0000 Max. :1.0000 Max. :1.000 Max. :1.00000 Max. :1.0000 Max. :1.0000

[1] "C:/Users/user/Documents/School/CSUF/ISDS577/projects/Capstone-577//output/V8517-greaterEqual-10-final.csv"

[1] "y:" "V8517"

[[1]]

[1] "final: "

$tabulatedCrossValidated

[1] <NA> V8502 V8509 V8527 V8536 V8512 V7206 V7552 V8514 V8526 V8530 V8505 V7215 V8528 V8565 V8529 V8531 V7202

Levels: V7202 V7206 V7215 V7552 V8502 V8505 V8509 V8512 V8514 V8526 V8527 V8528 V8529 V8530 V8531 V8536 V8565

$Freq

[1] 2.317 0.617 0.533 0.433 0.333 0.317 0.300 0.267 0.267 0.267 0.250 0.233 0.217 0.150 0.100 0.067 0.067 0.017

tabulatedCrossValidated Freq

2 V8502 0.617

3 V8509 0.533

4 V8527 0.433

5 V8536 0.333

6 V8512 0.317

7 V7206 0.300

8 V7552 0.267

9 V8514 0.267

10 V8526 0.267

11 V8530 0.250

12 V8505 0.233

13 V7215 0.217

14 V8528 0.150

15 V8565 0.100

16 V8529 0.067

17 V8531 0.067

18 V7202 0.017

[1] "keep: > " "0.25" "9" "V8502" "V8509" "V8527" "V8536" "V8512" "V7206" "V7552" "V8514" "V8526"

V8502 V8527 V8536 V7206 V7552 V8514 V8526

2.0156523 5.3566775 4.9321064 -0.7355915 1.1383145 1.8866605 4.9423597

V8517 V8502 V8509 V8527 V8536 V8512 V7206 V7552 V8514 V8526

Min. :0.0 Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000

1st Qu.:0.0 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000

Median :0.5 Median :0.0000 Median :0.0000 Median :0.0000 Median :0.00000 Median :0.0000 Median :1.0000 Median :1.0000 Median :0.0000 Median :0.0000

Mean :0.5 Mean :0.1825 Mean :0.1406 Mean :0.2702 Mean :0.04404 Mean :0.3733 Mean :0.7207 Mean :0.6955 Mean :0.4462 Mean :0.2443

3rd Qu.:1.0 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:0.00000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000

Max. :1.0 Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000

Call:

imcdiag(x = x, y = y, method = method, corr = FALSE, vif = vif,

tol = tol, conf = conf, cvif = cvif, leamer = leamer, all = all)

VIF Multicollinearity Diagnostics

VIF detection

V8502 1.4815 0

V8509 1.4874 0

V8527 1.5427 0

V8536 1.1356 0

V8512 2.4656 0

V7206 1.0040 0

V7552 1.1440 0

V8514 2.6250 0

V8526 1.5116 0

NOTE: VIF Method Failed to detect multicollinearity

0 --> COLLINEARITY is not detected by the test

===================================

[1] "MC summary"

Call:

glm(formula = y ~ ., family = family, data = data.frame(Xy[,

c(bestset[-1], FALSE), drop = FALSE], y = y))

Deviance Residuals:

Min 1Q Median 3Q Max

-5.0246 -0.4343 -0.1248 0.1200 2.6394

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.71647 0.11031 -24.626 <2e-16 \*\*\*

V8502 2.01565 0.10795 18.671 <2e-16 \*\*\*

V8527 5.35668 0.34380 15.581 <2e-16 \*\*\*

V8536 4.93211 0.59758 8.254 <2e-16 \*\*\*

V7206 -0.73559 0.08648 -8.506 <2e-16 \*\*\*

V7552 1.13831 0.10437 10.906 <2e-16 \*\*\*

V8514 1.88666 0.08269 22.817 <2e-16 \*\*\*

V8526 4.94236 0.32896 15.024 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 11173.5 on 8059 degrees of freedom

Residual deviance: 4032.3 on 8052 degrees of freedom

AIC: 4048.3

Number of Fisher Scoring iterations: 8

[1] "optCutOff\_sens:" "0"

[1] "error rate sens: 0.5"

[1] "yhat.transformed\_center sens matrix"

[1] "n:" "8060"

1

0 0.5

1 0.5

[1] "optCutOff\_center" "1"

[1] "error rate c: 0.1119"

[1] "yhat.transformed\_center conf matrix"

[1] "n:" "8060"

0 1

0 0.4697 0.0303

1 0.0816 0.4184

[1] "optCutOff\_spec" "0.01"

[1] "error rate spec: 0.1119"

[1] "yhat.transformed\_spec conf matrix"

[1] "n:" "8060"

0 1

0 0.4697 0.0303

1 0.0816 0.4184

[1] "MC model applied to Pop :" "0.252417878296369"

[1] "Pop model applied to pop :" "0.232831548837981"

[1] "CV Model applied to population"

Call:

glm(formula = holderOfData, family = binomial(link = "logit"))

Deviance Residuals:

Min 1Q Median 3Q Max

-1.8324 -0.5996 -0.5696 -0.3735 2.3226

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.51605 0.08985 -28.002 < 2e-16 \*\*\*

V8527 0.62375 0.07715 8.085 6.24e-16 \*\*\*

V8536 2.35310 0.12190 19.304 < 2e-16 \*\*\*

V7206 -0.11147 0.06745 -1.653 0.0984 .

V7552 0.89111 0.08310 10.724 < 2e-16 \*\*\*

V8514 0.09686 0.06843 1.416 0.1569

V8526 0.02348 0.07972 0.294 0.7684

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 7656.4 on 8058 degrees of freedom

Residual deviance: 6892.6 on 8052 degrees of freedom

AIC: 6906.6

Number of Fisher Scoring iterations: 5

V8517 V8502 V8527 V8536 V7206 V7552 V8514 V8526

Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.000000 Min. :0.00 Min. :0.0000 Min. :0.0000 Min. :0.00000

1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.000000 1st Qu.:0.00 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.00000

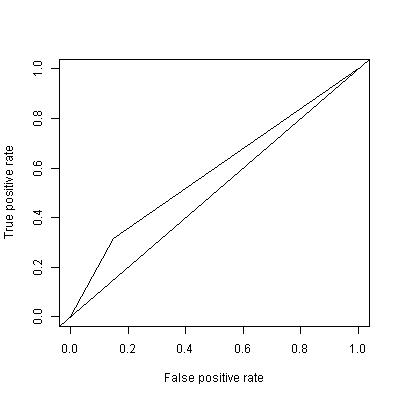
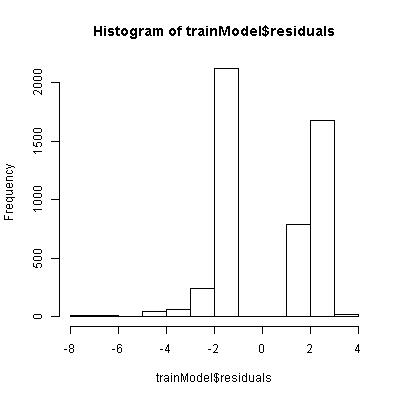
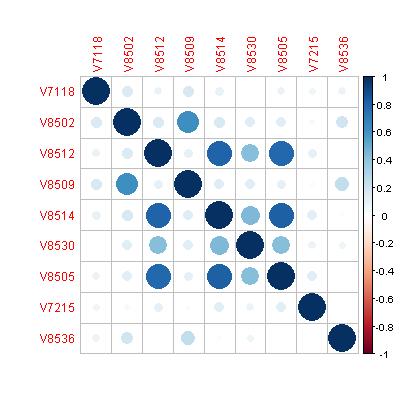
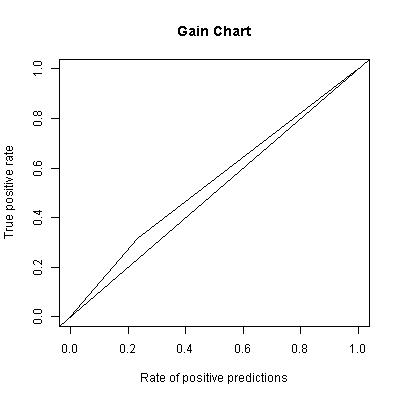
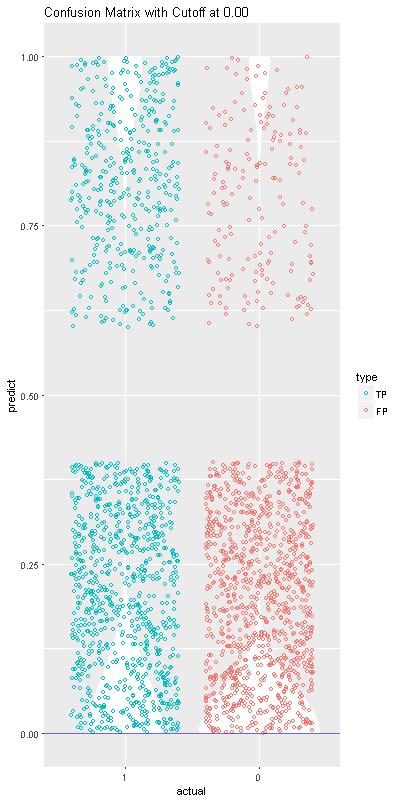
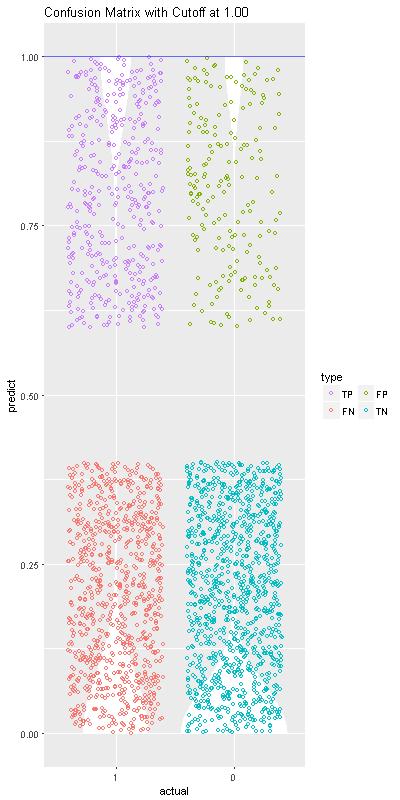
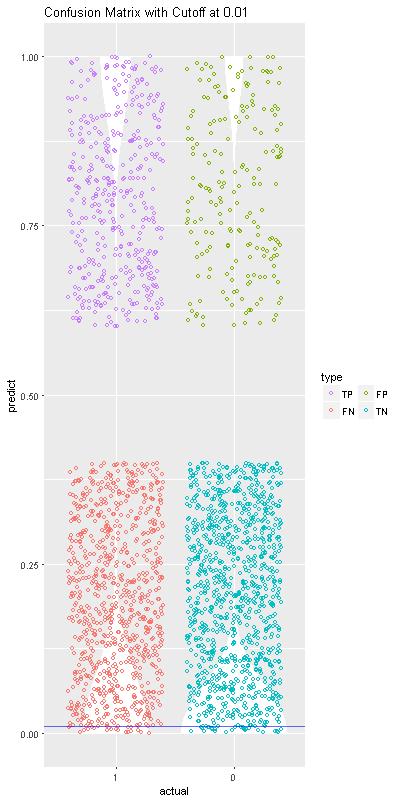
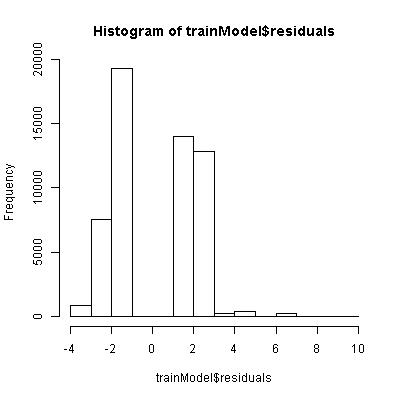
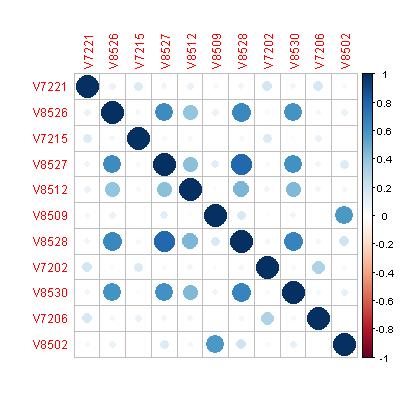
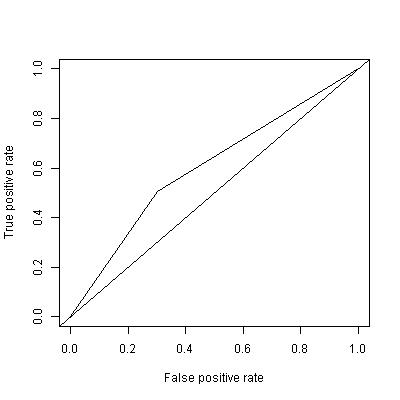
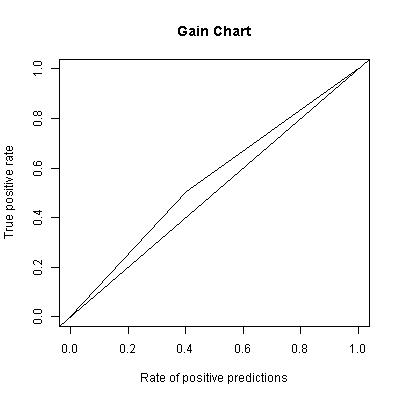
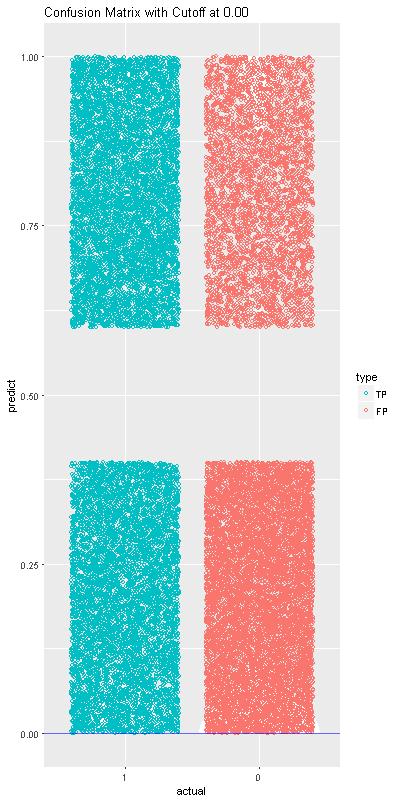
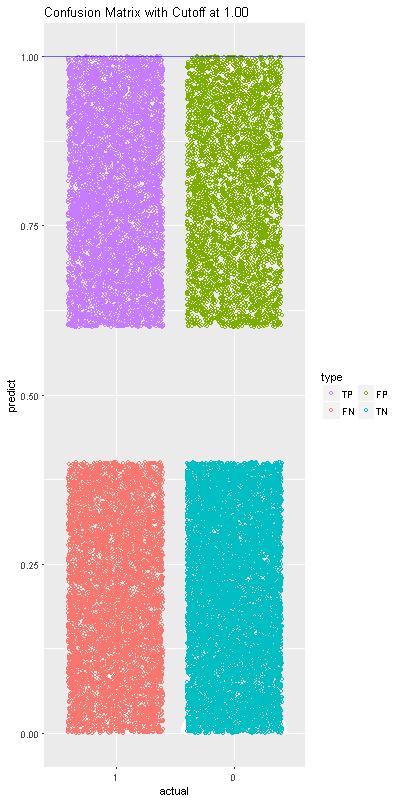
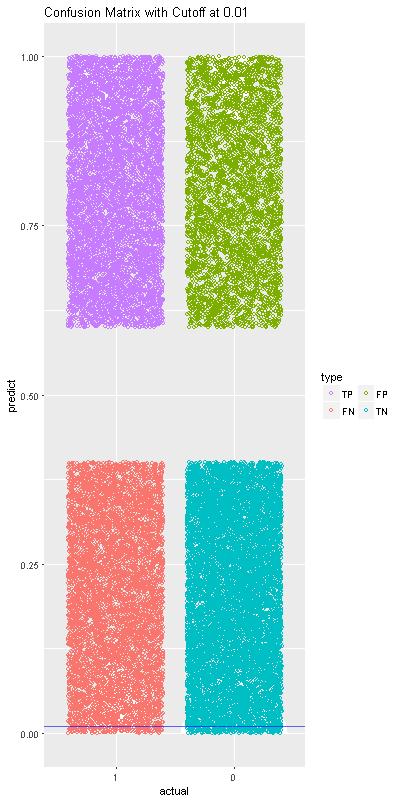
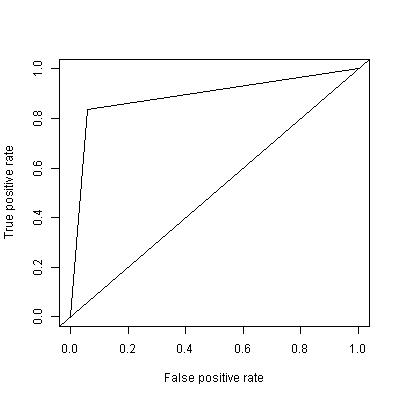
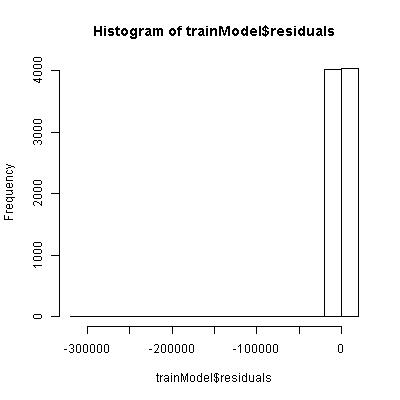
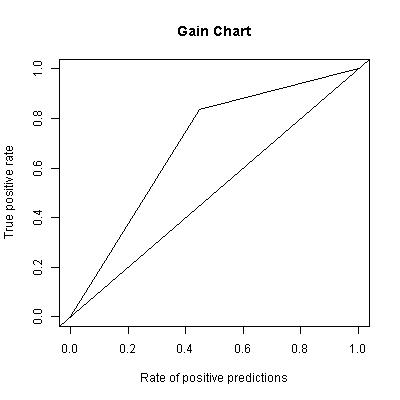
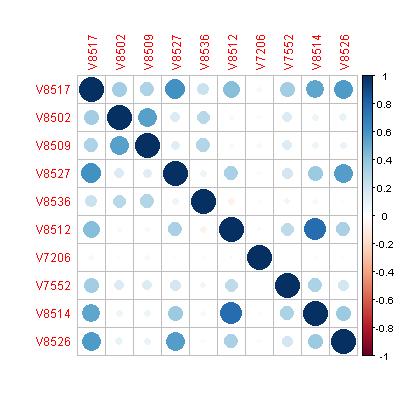
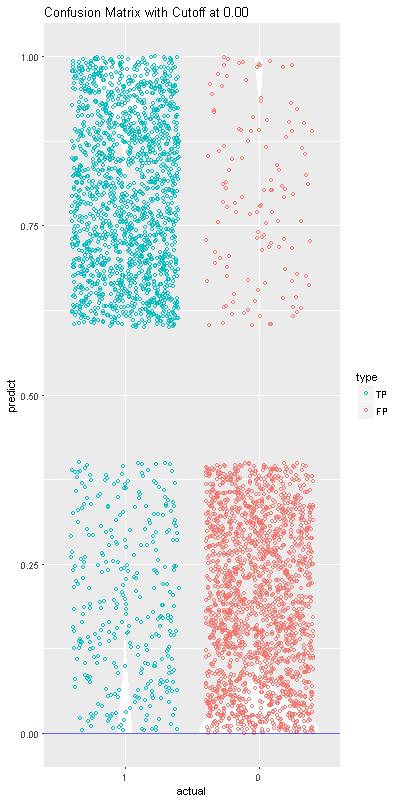
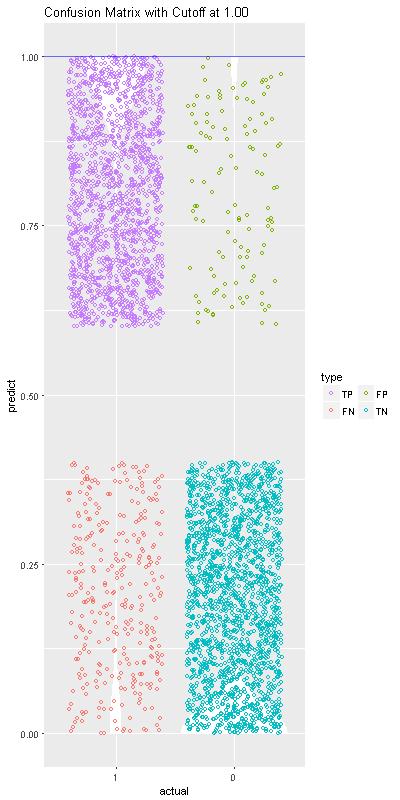
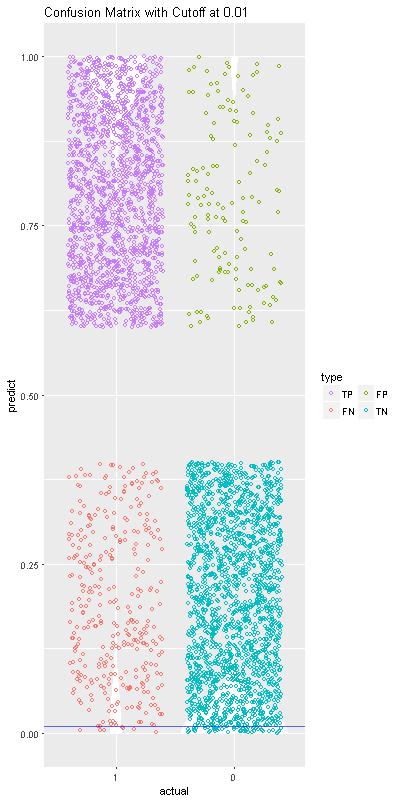
Median :0.00000 Median :0.00000 Median :0.00000 Median :0.000000 Median :1.00 Median :1.0000 Median :0.0000 Median :0.00000

Mean :0.05849 Mean :0.06342 Mean :0.03464 Mean :0.005246 Mean :0.74 Mean :0.5613 Mean :0.2039 Mean :0.03149

3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.000000 3rd Qu.:1.00 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:0.00000

Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.000000 Max. :1.00 Max. :1.0000 Max. :1.0000 Max. :1.00000

**REGRESSION EQUATIONS**

Reference

We found a guide authored by PhD Data Scientist Selva Prabhakaran. <http://r-statistics.co/Logistic-Regression-With-R.html>