Capstone Data Logistic Regression - Predict Cottonwood and Willow

Tom Thorpe

July 25, 2018

Objective

Use Logistic regression to predict tree coverage.

```
# Include required libraries.

library(gsubfn)
```

```
## Loading required package: proto
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(ggridges) # for easier viewing of sub-group distributions
library(ROCR)
```

```
## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
suppressMessages(library(latticeExtra, warn.conflicts = FALSE, quietly=TRUE))
#library(latticeExtra)

  curTime=Sys.time()
  print(paste("Forest Cover Logistic script started at",curTime))
```

```
## [1] "Forest Cover Logistic script started at 2018-08-12 17:45:31"
```

```
#Point to data. The forestcover_clean_full.csv is the cleaned data to be graphed.

calcROC <- 1
saveFileName="ForestCoverLogisticStats.csv"

infile="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcover_clean_full.csv"
```

```r
#infile="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcover_clean.csv"
#infile="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcoversmall_clean_full.csv"
#infile="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcoversmall_clean.csv"
out2file="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcover_graph.csv"
#out1file="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcoversmall_clean_full.csv"
#out2file="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcoversmall_clean.csv"

alphaVal<-0.05 # large data
#alphaVal<-0.1  # small data
```

```r
forestcover <- read.csv(infile,header=TRUE,sep=",") %>% tbl_df()
  curTime=Sys.time()
  print(paste("Forest Cover data load completed at",curTime))
```

```
## [1] "Forest Cover data load completed at 2018-08-12 17:46:11"
```

```r
forestcover$SoilType<-as.factor(forestcover$SoilType)
forestcover$ClimateZone<-as.factor(forestcover$ClimateZone)
forestcover$GeoZone<-as.factor(forestcover$GeoZone)

# glimpse(forestcover)

# table(forestcover$Sed_mix)
#knitr::knit_exit()

# Coverage binary outcome Vars:
# Aspen
# Cottonwood_Willow
# DouglasFir
# Krummholz
# LodgepolePine
# PonderosaPine
# Spruce_Fir
```

A table showing the number of occurrences for each tree type is shown below.

```r
covCount<-data.frame(table(forestcover$CovName))
totCount<-nrow(forestcover)
covCount <- mutate(covCount,Percent = as.integer(covCount$Freq*1000/totCount)/10)
LodgePct<-covCount$Percent[covCount$Var1=="Lodgepole"]
SpruceAndFirPct<-covCount$Percent[covCount$Var1=="Spruce&Fir"]
LodgeAndSpruceAndFir<-LodgePct+SpruceAndFirPct
#```
#```{r echo=TRUE}
covCount
```

```
##             Var1   Freq Percent
## 1          Aspen   9493     1.6
## 2 Cotton&Willow   2747     0.4
## 3     DouglasFir  17367     2.9
## 4      Krummholz  20510     3.5
## 5      Lodgepole 283301    48.7
## 6      Ponderosa  35754     6.1
## 7     Spruce&Fir 211840    36.4
```

Lodge pole Pine represents 48.7 percent of the sample. So always guessing “Lodge pole” would provide success rate of 48.7 percent and can be used as a baseline for comparing our predictions. Spruce & Fir represent the next largest number of trees. The two together represent 85.1 percent.

Logistic Model Accuracy Function

A function to help determine threshold for best accuracy and testing is shown below.

```
source("logisticAccuracy.R")
bestThreshIndex=11
```

Create Training and Testing Sets

Split data into training and testing data for logistic regression. The split is based on cover type so that the different coverage types will be split proportionately for all cover types in the training and test sets.

```
library(caTools)
set.seed(127)
split = sample.split(forestcover$CovType, 0.70) # we want 65% in the training set
forestTrain = subset(forestcover, split == TRUE)
forestTest  = subset(forestcover, split == FALSE)
```

Check training set coverage percentages and compare with test set to ensure there is a representative amount of data in each set for each coverage type.

View Training Set Coverage Percentages

Check training set coverage percentages.

```
covCount<-data.frame(table(forestTrain$CovName))
totCount<-nrow(forestTrain)
covCount <- mutate(covCount,Percent = as.integer(covCount$Freq*1000/totCount)/10)
covCount
```

```
##            Var1   Freq Percent
## 1         Aspen   6645     1.6
## 2 Cotton&Willow   1923     0.4
## 3    DouglasFir  12157     2.9
## 4     Krummholz  14357     3.5
## 5     Lodgepole 198311    48.7
## 6     Ponderosa  25028     6.1
## 7    Spruce&Fir 148288    36.4
```

View Test Set Coverage Percentages

Check test set coverage percentages.

```
covCount<-data.frame(table(forestTest$CovName))
totCount<-nrow(forestTest)
covCount <- mutate(covCount,Percent = as.integer(covCount$Freq*1000/totCount)/10)
covCount
```

```
##            Var1  Freq Percent
## 1         Aspen  2848     1.6
```

```
## 2 Cotton&Willow   824     0.4
## 3    DouglasFir  5210     2.9
## 4     Krummholz  6153     3.5
## 5     Lodgepole 84990    48.7
## 6     Ponderosa 10726     6.1
## 7    Spruce&Fir 63552    36.4
```

```
# knitr::knit_exit() # exit early

#glimpse(forestTrain)
#glimpse(forestTest)
#summary(forestTrain)
#summary(forestTest)
#table(forestTrain$Sed_mix)
#table(forestTrain$GeoName)
#table(forestTrain$Spruce_Fir)
#table(forestTest$Spruce_Fir)

# the above all work without error.

#table(forestTest$Rock_Land)
# Get the following error with above code:
#  Error in table(SpfFir_test$Rock_Land) : object 'SpfFir_test' not found
#    Calls: <Anonymous> ... withCallingHandlers -> withVisible -> eval -> eval -> table


#table(forestTrain$Rock_Land)
#table(forestTest$Rock_Land)
#table(forestTrain$Rubbly)
#table(forestTest$Rubbly)

#table(forestTrain$Sed_mix)
#table(forestTrain$Gateview)
#table(forestTrain$Rubbly)
#table(forestTest$Sed_mix)
#table(forestTest$Gateview)
#table(forestTest$Rubbly)

############# Start Start Start Start Start Start Start Start #################
```

Cottonwood and Willow Logistic Regression

Logistic regression models are created and compared for the Cottonwood and Willow coverage type. The outcome is based on the binary 'Cottonwood_Willow' variable.

Cottonwood and Willow Logistic Regression - All Variables

Create Cottonwood and Willow Logistic Model - All Vars

Create the Cottonwood and Willow logistic model for the Aggregated Soil data using all independent variables.

Cottonwood and Willow All Aggregated Soil Types

The original project used aggregated Soil Types. Compute a logistic regression model using the aggregated soil types to see how the dis-aggregated / individuated variables compare.

```
# You can remove the levels of the factor variables using the option exclude:
#   lm(dependent ~ factor(independent1, exclude=c('b','d')) + independent2)
#   This way the factors b, d will not be included in the regression.

curTime=Sys.time()
print(paste("Cottonwood_Willow aggregated Logistic Model Calculation started at",curTime))
```

```
## [1] "Cottonwood_Willow aggregated Logistic Model Calculation started at 2018-08-12 17:46:14"
```

```
CotWil_Agg_LogMod =
  glm(Cottonwood_Willow ~
        Elev +     # Elevation in meters of data cell
        Aspect +   # Direction in degrees slope faces
        Slope +    # Slope / steepness of hill in degrees (0 to 90)
        H2OHD +    # Horizontal distance in meters to nearest water
        H2OVD +    # Vertical distance in meters to nearest water
        RoadHD +   # Horizontal distance in meters to nearest road
        FirePtHD + # Horizontal distance in meters to nearest fire point
        Shade9AM + Shade12PM + Shade3PM + # Amount of shade at 9am, 12pm and 3pm
        # Wilderness areas:
          RWwild + NEwild + CMwild + CPwild +
        # Aggregated Soil type:
          ST01 + ST02 + ST03 + ST04 + ST05 + ST06 + ST07 + ST08 + ST09 + ST10 +
          ST11 + ST12 + ST13 + ST14 + ST15 + ST16 + ST17 + ST18 + ST19 + ST20 +
          ST21 + ST22 + ST23 + ST24 + ST25 + ST26 + ST27 + ST28 + ST29 + ST30 +
          ST31 + ST32 + ST33 + ST34 + ST35 + ST36 + ST37 + ST38 + ST39 + ST40 ,
        data=forestTrain, family=binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
CotWil_Agg_All_LogMod = CotWil_Agg_LogMod
save("CotWil_Agg_All_LogMod", file="CotWil_Agg_All_LogMod.Rdata")

CotWil_Agg_All_aic<-as.integer(CotWil_Agg_LogMod$aic)
CotWil_Agg_All_aic
```

```
## [1] 7932
```

```
curTime=Sys.time()
print(paste("Cottonwood_Willow aggregated Logistic Model Calculation completed at",curTime))
```

```
## [1] "Cottonwood_Willow aggregated Logistic Model Calculation completed at 2018-08-12 17:48:55"
```

Check the coefficients for the Cottonwood and Willow model using all aggregated data.

```
summary(CotWil_Agg_LogMod)
```

```
##
## Call:
## glm(formula = Cottonwood_Willow ~ Elev + Aspect + Slope + H2OHD +
##     H2OVD + RoadHD + FirePtHD + Shade9AM + Shade12PM + Shade3PM +
##     RWwild + NEwild + CMwild + CPwild + ST01 + ST02 + ST03 +
##     ST04 + ST05 + ST06 + ST07 + ST08 + ST09 + ST10 + ST11 + ST12 +
##     ST13 + ST14 + ST15 + ST16 + ST17 + ST18 + ST19 + ST20 + ST21 +
```

```
##     ST22 + ST23 + ST24 + ST25 + ST26 + ST27 + ST28 + ST29 + ST30 +
##     ST31 + ST32 + ST33 + ST34 + ST35 + ST36 + ST37 + ST38 + ST39 +
##     ST40, family = binomial, data = forestTrain)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.168   0.000   0.000   0.000   3.873
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.680e+07  7.364e+10  -0.001  0.99949
## Elev        -6.755e-03  3.074e-04 -21.973  < 2e-16 ***
## Aspect      -6.421e-04  4.394e-04  -1.462  0.14385
## Slope       -5.403e-02  8.589e-03  -6.290 3.17e-10 ***
## H2OHD       -6.242e-03  5.652e-04 -11.044  < 2e-16 ***
## H2OVD        1.237e-02  1.234e-03  10.024  < 2e-16 ***
## RoadHD       1.779e-03  8.288e-05  21.460  < 2e-16 ***
## FirePtHD     1.507e-03  8.463e-05  17.810  < 2e-16 ***
## Shade9AM     1.349e-02  7.825e-03   1.724  0.08479 .
## Shade12PM    3.566e-02  6.549e-03   5.445 5.18e-08 ***
## Shade3PM    -1.778e-02  6.216e-03  -2.860  0.00423 **
## RWwild      -2.927e+01  4.877e+02  -0.060  0.95215
## NEwild      -2.250e+01  9.072e+02  -0.025  0.98021
## CMwild      -2.443e+01  6.238e+02  -0.039  0.96876
## CPwild              NA         NA      NA       NA
## ST01         4.680e+07  7.364e+10   0.001  0.99949
## ST02         4.680e+07  7.364e+10   0.001  0.99949
## ST03         4.680e+07  7.364e+10   0.001  0.99949
## ST04         4.680e+07  7.364e+10   0.001  0.99949
## ST05         4.680e+07  7.364e+10   0.001  0.99949
## ST06         4.680e+07  7.364e+10   0.001  0.99949
## ST07         4.680e+07  7.364e+10   0.001  0.99949
## ST08         4.680e+07  7.364e+10   0.001  0.99949
## ST09         4.680e+07  7.364e+10   0.001  0.99949
## ST10         4.680e+07  7.364e+10   0.001  0.99949
## ST11         4.680e+07  7.364e+10   0.001  0.99949
## ST12         4.680e+07  7.364e+10   0.001  0.99949
## ST13         4.680e+07  7.364e+10   0.001  0.99949
## ST14         4.680e+07  7.364e+10   0.001  0.99949
## ST15         4.680e+07  7.364e+10   0.001  0.99949
## ST16         4.680e+07  7.364e+10   0.001  0.99949
## ST17         4.680e+07  7.364e+10   0.001  0.99949
## ST18         4.680e+07  7.364e+10   0.001  0.99949
## ST19         4.680e+07  7.364e+10   0.001  0.99949
## ST20         4.680e+07  7.364e+10   0.001  0.99949
## ST21         4.680e+07  7.364e+10   0.001  0.99949
## ST22         4.680e+07  7.364e+10   0.001  0.99949
## ST23         4.680e+07  7.364e+10   0.001  0.99949
## ST24         4.680e+07  7.364e+10   0.001  0.99949
## ST25         4.680e+07  7.364e+10   0.001  0.99949
## ST26         4.680e+07  7.364e+10   0.001  0.99949
## ST27         4.680e+07  7.364e+10   0.001  0.99949
## ST28         4.680e+07  7.364e+10   0.001  0.99949
## ST29         4.680e+07  7.364e+10   0.001  0.99949
```

```
## ST30         4.680e+07  7.364e+10   0.001  0.99949
## ST31         4.680e+07  7.364e+10   0.001  0.99949
## ST32         4.680e+07  7.364e+10   0.001  0.99949
## ST33         4.680e+07  7.364e+10   0.001  0.99949
## ST34         4.680e+07  7.364e+10   0.001  0.99949
## ST35         4.680e+07  7.364e+10   0.001  0.99949
## ST36         4.680e+07  7.364e+10   0.001  0.99949
## ST37         4.680e+07  7.364e+10   0.001  0.99949
## ST38         4.680e+07  7.364e+10   0.001  0.99949
## ST39         4.680e+07  7.364e+10   0.001  0.99949
## ST40         4.680e+07  7.364e+10   0.001  0.99949
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 24429.2  on 406708  degrees of freedom
## Residual deviance:  7824.2  on 406655  degrees of freedom
## AIC: 7932.2
##
## Number of Fisher Scoring iterations: 23
```

WOW! The intercept is huge and listed as not significant. Wilderness area and several soil types are not significant and can be removed in the next iteration.

Cottonwood and Willow All Individuated Soil Types

Create a logistic model using the Individuated variables that were derived from the Soil Types. The Soil Type was the intersection of climate zone, geology zone, soil families, and rock content. These variables are used instead of the Soil types.

```
curTime=Sys.time()
print(paste("Cottonwood_Willow Individual Logistic Model Calculation started at",curTime))
```

```
## [1] "Cottonwood_Willow Individual Logistic Model Calculation started at 2018-08-12 17:48:55"
```

```
CotWil_Ind_LogMod =
  glm(Cottonwood_Willow ~
        Elev +     # Elevation in meters of cell
        Aspect +   # Direction in degrees slope faces
        Slope +    # Slope / steepness of hill in degrees (0 to 90)
        H2OHD +    # Horizontal distance in meters to nearest water
        H2OVD +    # Vertical distance in meters to nearest water
        RoadHD +   # Horizontal distance in meters to nearest road
        FirePtHD + # Horizontal distance in meters to nearest fire point
        Shade9AM + Shade12PM + Shade3PM + # Amount of shade at 9am, 12pm and 3pm
        # Wilderness areas:
          RWwild + NEwild + CMwild + CPwild +
        # Climate Zone:
        # ClimateName +
          Montane_low + Montane + SubAlpine + Alpine + Dry + Non_Dry +
        # Geology Zone:
        # GeoName +
          Alluvium + Glacial + Sed_mix + Ign_Meta +
        # Soil Family:
```

```r
            Aquolis_cmplx + Argiborolis_Pachic + Borohemists_cmplx + Bross +
            Bullwark + Bullwark_Cmplx + Catamount + Catamount_cmplx +
            Cathedral + Como + Cryaquepts_cmplx + Cryaquepts_Typic + Cryaquolls +
            Cryaquolls_cmplx + Cryaquolls_Typic + Cryaquolls_Typic_cmplx +
            Cryoborolis_cmplx + Cryorthents + Cryorthents_cmplx + Cryumbrepts +
            Cryumbrepts_cmplx + Gateview + Gothic + Granile + Haploborolis +
            Legault + Legault_cmplx + Leighcan + Leighcan_cmplx + Leighcan_warm +
            Moran + Ratake + Ratake_cmplx + Rogert + Supervisor_Limber_cmplx +
            Troutville + Unspecified + Vanet + Wetmore +
          # Soil Rock composition:
            Bouldery_ext + Rock_Land + Rock_Land_cmplx + Rock_Outcrop +
            Rock_Outcrop_cmplx + Rubbly + Stony + Stony_extreme + Stony_very +
            Till_Substratum ,
          data=forestTrain, family=binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
CotWil_Ind_All_LogMod = CotWil_Ind_LogMod
save("CotWil_Ind_All_LogMod", file="CotWil_Ind_All_LogMod.Rdata")

#table(forestTrain$GeoName)
#table(forestTrain$Sed_mix)
#table(forestTrain$Gateview)
# above: Error in table(SpfFir_test$Gateview) : object 'SpfFir_train' not found <-------

CotWil_Ind_All_aic<-as.integer(CotWil_Ind_LogMod$aic)
CotWil_Ind_All_aic
```

```
## [1] 7938
```

```r
summary(CotWil_Ind_LogMod)
```

```
##
## Call:
## glm(formula = Cottonwood_Willow ~ Elev + Aspect + Slope + H2OHD +
##     H2OVD + RoadHD + FirePtHD + Shade9AM + Shade12PM + Shade3PM +
##     RWwild + NEwild + CMwild + CPwild + Montane_low + Montane +
##     SubAlpine + Alpine + Dry + Non_Dry + Alluvium + Glacial +
##     Sed_mix + Ign_Meta + Aquolis_cmplx + Argiborolis_Pachic +
##     Borohemists_cmplx + Bross + Bullwark + Bullwark_Cmplx + Catamount +
##     Catamount_cmplx + Cathedral + Como + Cryaquepts_cmplx + Cryaquepts_Typic +
##     Cryaquolls + Cryaquolls_cmplx + Cryaquolls_Typic + Cryaquolls_Typic_cmplx +
##     Cryoborolis_cmplx + Cryorthents + Cryorthents_cmplx + Cryumbrepts +
##     Cryumbrepts_cmplx + Gateview + Gothic + Granile + Haploborolis +
##     Legault + Legault_cmplx + Leighcan + Leighcan_cmplx + Leighcan_warm +
##     Moran + Ratake + Ratake_cmplx + Rogert + Supervisor_Limber_cmplx +
##     Troutville + Unspecified + Vanet + Wetmore + Bouldery_ext +
##     Rock_Land + Rock_Land_cmplx + Rock_Outcrop + Rock_Outcrop_cmplx +
##     Rubbly + Stony + Stony_extreme + Stony_very + Till_Substratum,
##     family = binomial, data = forestTrain)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.168   0.000   0.000   0.000   3.873
##
```

```
## Coefficients: (17 not defined because of singularities)
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -8.666e+08  4.422e+11  -0.002   0.9984
## Elev                    -6.754e-03  3.858e-04 -17.507  < 2e-16 ***
## Aspect                  -6.413e-04  4.427e-04  -1.449   0.1475
## Slope                   -5.406e-02  8.699e-03  -6.214 5.16e-10 ***
## H2OHD                   -6.245e-03  5.875e-04 -10.630  < 2e-16 ***
## H2OVD                    1.237e-02  1.238e-03   9.994  < 2e-16 ***
## RoadHD                   1.779e-03  8.367e-05  21.258  < 2e-16 ***
## FirePtHD                 1.507e-03  8.566e-05  17.597  < 2e-16 ***
## Shade9AM                 1.352e-02  7.910e-03   1.709   0.0875 .
## Shade12PM                3.565e-02  6.605e-03   5.397 6.78e-08 ***
## Shade3PM                -1.778e-02  6.227e-03  -2.855   0.0043 **
## RWwild                  -2.626e+01  1.084e+02  -0.242   0.8086
## NEwild                  -1.946e+01  2.011e+02  -0.097   0.9229
## CMwild                  -2.141e+01  1.385e+02  -0.155   0.8771
## CPwild                          NA         NA      NA       NA
## Montane_low             -4.648e+09  5.256e+11  -0.009   0.9929
## Montane                 -6.288e+10  2.381e+12  -0.026   0.9789
## SubAlpine               -1.392e+08  1.151e+12   0.000   0.9999
## Alpine                  -1.392e+08  1.151e+12   0.000   0.9999
## Dry                     -1.795e+11  7.437e+12  -0.024   0.9807
## Non_Dry                  5.515e+09  7.686e+11   0.007   0.9943
## Alluvium                 5.837e+10  2.161e+12   0.027   0.9784
## Glacial                  7.012e-01  2.390e+02   0.003   0.9977
## Sed_mix                  2.432e+11  9.265e+12   0.026   0.9791
## Ign_Meta                        NA         NA      NA       NA
## Aquolis_cmplx            1.793e+11  7.135e+12   0.025   0.9799
## Argiborolis_Pachic              NA         NA      NA       NA
## Borohemists_cmplx        1.006e+09  1.333e+12   0.001   0.9994
## Bross                    1.006e+09  1.333e+12   0.001   0.9994
## Bullwark                 5.823e+10  2.252e+12   0.026   0.9794
## Bullwark_Cmplx           5.823e+10  2.252e+12   0.026   0.9794
## Catamount               -2.691e+00  1.509e+03  -0.002   0.9986
## Catamount_cmplx          2.614e+00  2.740e+02   0.010   0.9924
## Cathedral               -8.291e-02  2.164e-01  -0.383   0.7016
## Como                     1.006e+09  1.333e+12   0.001   0.9994
## Cryaquepts_cmplx         8.058e-01  1.126e+03   0.001   0.9994
## Cryaquepts_Typic        -5.737e+10  2.327e+12  -0.025   0.9803
## Cryaquolls               1.739e+00  8.707e+02   0.002   0.9984
## Cryaquolls_cmplx         2.563e+00  8.707e+02   0.003   0.9977
## Cryaquolls_Typic        -5.837e+10  2.161e+12  -0.027   0.9784
## Cryaquolls_Typic_cmplx  -1.169e-01  1.669e+02  -0.001   0.9994
## Cryoborolis_cmplx               NA         NA      NA       NA
## Cryorthents             1.006e+09  1.333e+12   0.001   0.9994
## Cryorthents_cmplx        6.293e+00  2.178e+03   0.003   0.9977
## Cryumbrepts                     NA         NA      NA       NA
## Cryumbrepts_cmplx               NA         NA      NA       NA
## Gateview                        NA         NA      NA       NA
## Gothic                   7.109e-01  4.083e+03   0.000   0.9999
## Granile                  1.006e+09  1.333e+12   0.001   0.9994
## Haploborolis             1.622e+00  2.006e-01   8.083 6.31e-16 ***
## Legault                  5.823e+10  2.252e+12   0.026   0.9794
## Legault_cmplx                   NA         NA      NA       NA
```

```
## Leighcan                   1.006e+09  1.333e+12   0.001   0.9994
## Leighcan_cmplx             4.637e+00  1.548e+03   0.003   0.9976
## Leighcan_warm              1.006e+09  1.333e+12   0.001   0.9994
## Moran                             NA         NA      NA       NA
## Ratake                     9.809e-01  2.159e-01   4.544 5.52e-06 ***
## Ratake_cmplx               3.798e+00  1.517e+03   0.003   0.9980
## Rogert                     5.837e+10  2.161e+12   0.027   0.9784
## Supervisor_Limber_cmplx           NA         NA      NA       NA
## Troutville                 5.823e+10  2.252e+12   0.026   0.9794
## Unspecified                1.793e+11  7.135e+12   0.025   0.9799
## Vanet                             NA         NA      NA       NA
## Wetmore                    1.583e-01  2.030e-01   0.780   0.4355
## Bouldery_ext                      NA         NA      NA       NA
## Rock_Land                 -7.346e-01  1.500e+02  -0.005   0.9961
## Rock_Land_cmplx            4.403e+00  1.517e+03   0.003   0.9977
## Rock_Outcrop               1.006e+09  1.333e+12   0.001   0.9994
## Rock_Outcrop_cmplx         4.076e+00  1.517e+03   0.003   0.9979
## Rubbly                            NA         NA      NA       NA
## Stony                             NA         NA      NA       NA
## Stony_extreme                     NA         NA      NA       NA
## Stony_very                        NA         NA      NA       NA
## Till_Substratum                   NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 24429.2  on 406708  degrees of freedom
## Residual deviance:  7824.2  on 406652  degrees of freedom
## AIC: 7938.2
##
## Number of Fisher Scoring iterations: 20
```

```r
  curTime=Sys.time()
  print(paste("Cottonwood_Willow Individual Logistic Model Calculation completed at",curTime))
```

```
## [1] "Cottonwood_Willow Individual Logistic Model Calculation completed at 2018-08-12 17:52:59"
```

```r
  #table(forestTest$Rock_Land)
  # Get the following error with above code:
  #  Error in table(SpfFir_test$Rock_Land) : object 'SpfFir_test' not found
  #    Calls: <Anonymous> ... withCallingHandlers -> withVisible -> eval -> eval -> table
```

Predict Cottonwood and Willow Logistic Model Probabilities - All Aggregated Vars

Cottonwood and Willow Probabilities - All Aggregated Data

Predict the probability of Cottonwood and Willow for aggregated Data - all variables.

```r
# Predict Cottonwood and Willow Agg Data - all variables

  CotWil_Agg_Train_predict= predict(CotWil_Agg_LogMod, type="response")
  CotWil_Agg_Train_Logit= predict(CotWil_Agg_LogMod)
  summary(CotWil_Agg_Train_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
```

```
## 0.000000 0.000000 0.000000 0.004728 0.000000 0.975556
```

```
str(CotWil_Agg_Train_predict)
```

```
##  Named num [1:406709] 6.45e-12 6.78e-12 1.39e-10 9.22e-12 1.82e-12 ...
##  - attr(*, "names")= chr [1:406709] "1" "2" "3" "4" ...
```

```
  #plot(table(CotWil_Agg_Train_predict))
  #plot(table(CotWil_Agg_Train_Logit))
  dens<-data.frame(table(CotWil_Agg_Train_predict))
# str(dens)

  CotWil_Agg_Test_predict= predict(CotWil_Agg_LogMod, type="response",newdata=forestTest)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
summary(CotWil_Agg_Test_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.000000 0.000000 0.004675 0.000000 0.977850
```

```
 str(CotWil_Agg_Test_predict)
```

```
##  Named num [1:174303] 3.76e-10 1.03e-11 9.22e-11 9.75e-11 3.69e-11 ...
##  - attr(*, "names")= chr [1:174303] "1" "2" "3" "4" ...
```

Cottonwood and Willow Probabilities - All Individuated Data

Predict the probability of Cottonwood and Willow for Individual Data - all variables.

```
CotWil_Ind_Train_predict= predict(CotWil_Ind_LogMod, type="response")
summary(CotWil_Ind_Train_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.000000 0.000000 0.004728 0.000000 0.975572
```

```
CotWil_Ind_Test_predict= predict(CotWil_Ind_LogMod, type="response",newdata=forestTest)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
summary(CotWil_Ind_Test_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.000000 0.000000 0.004675 0.000000 0.977858
```

Cottonwood and Willow Receiver Operating Characteristic (ROC) - All Vars

Cottonwood and Willow Receiver ROC - All Aggregated Data

Next, look at the True Positive and False Positive rates based on threshold value for the aggregated data.

```
if (calcROC) {
  curTime=Sys.time()
  print(paste("ROC graph 1 started at",curTime))

  ROCpred_CotWil_Agg = prediction(CotWil_Agg_Train_predict, forestTrain$Cottonwood_Willow)
  summary(ROCpred_CotWil_Agg)
```

```
  ROCperf_CotWil_Agg = performance(ROCpred_CotWil_Agg, "tpr", "fpr")
  summary(ROCperf_CotWil_Agg)

  CotWil_Agg_All_ROC_AUC = as.numeric(performance(ROCpred_CotWil_Agg, "auc")@y.values)
  CotWil_Agg_All_ROC_AUC=as.integer(as.numeric(CotWil_Agg_All_ROC_AUC)*1000)/10
  print(paste("CotWil_Agg_All_ROC_AUC=",CotWil_Agg_All_ROC_AUC))

  jpeg(filename="Fig-ROCR_perf_CotWil_Agg.jpg")
  plot(ROCperf_CotWil_Agg, colorize=TRUE, print.cutoffs.at=seq(0,1,0.1), text.adj=c(-0.2,1.7))
  dev.off()
} else {
  CotWil_Agg_All_ROC_AUC = 84.2
}
```

```
## [1] "ROC graph 1 started at 2018-08-12 17:53:04"
## [1] "CotWil_Agg_All_ROC_AUC= 99.5"
```

```
## pdf
##   2
```

Cottonwood and Willow Receiver ROC - All Individuated Data

The Response Operating Curve for the individuated data is shown below.

```
if (calcROC) {
  curTime=Sys.time()
  print(paste("ROCR graph 2 started at",curTime))

  ROCpred_CotWil_Ind = prediction(CotWil_Ind_Train_predict, forestTrain$Cottonwood_Willow)
  summary(ROCpred_CotWil_Ind)
  ROCperf_CotWil_Ind = performance(ROCpred_CotWil_Ind, "tpr", "fpr")
  summary(ROCperf_CotWil_Ind)

  CotWil_Ind_All_ROC_AUC = as.numeric(performance(ROCpred_CotWil_Ind, "auc")@y.values)
  CotWil_Ind_All_ROC_AUC=as.integer(as.numeric(CotWil_Ind_All_ROC_AUC)*1000)/10
  print(paste("CotWil_Ind_All_ROC_AUC=",CotWil_Ind_All_ROC_AUC))

  jpeg(filename="Fig-ROCR_perf_CotWil_Ind.jpg")
  plot(ROCperf_CotWil_Ind, colorize=TRUE, print.cutoffs.at=seq(0,1,0.1), text.adj=c(-0.2,1.7))
  dev.off()
} else {
  CotWil_Ind_All_ROC_AUC = 84.2
}
```

```
## [1] "ROCR graph 2 started at 2018-08-12 17:55:29"
## [1] "CotWil_Ind_All_ROC_AUC= 99.5"
```

```
## pdf
##   2
```

The threshold graphs are essentially identical. This is making me think that there is not much difference between the two models. The AIC score for the Soil Type model is AIC: 351676 and for the individuated variables is: AIC: 351839. The Soil type model AIC score is 0.046% better than the individuated model.

```
curTime=Sys.time()
print(paste("ROCR graph 2 completed at",curTime))
```

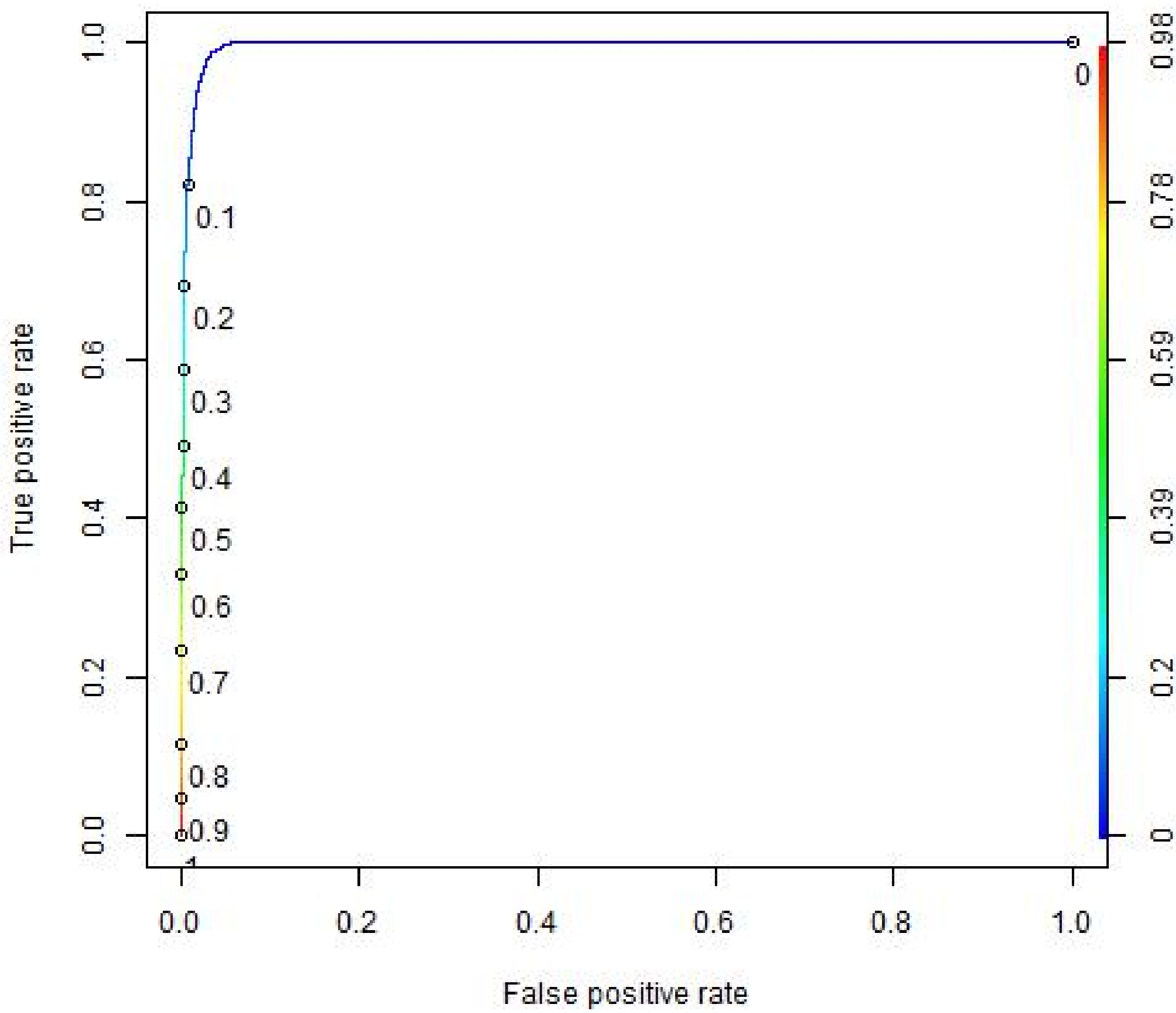

Figure 1: Cottonwood and Willow ROC for All Aggregated Data

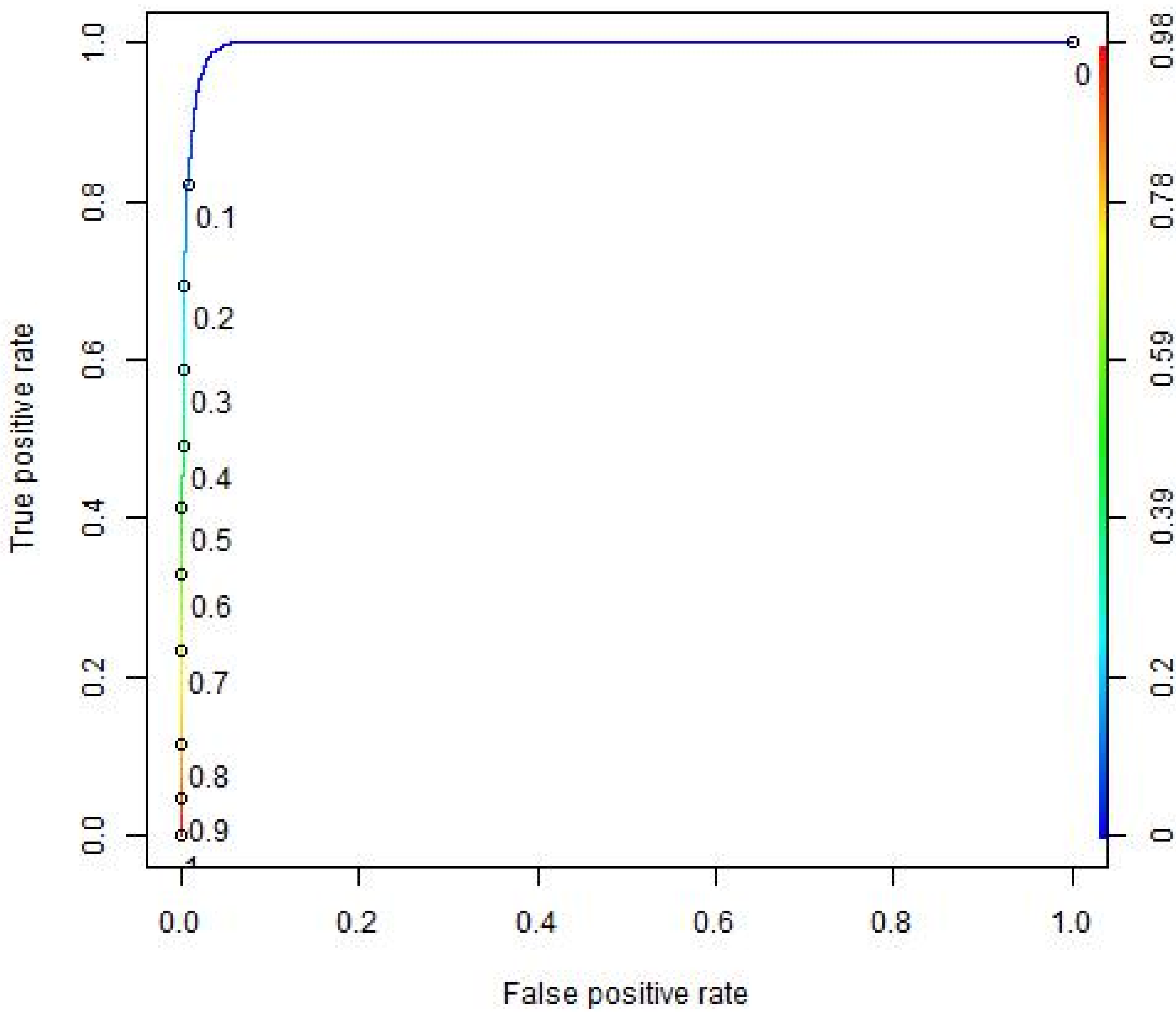

Figure 2: Cottonwood and Willow ROC for All Individuated Data

```
## [1] "ROCR graph 2 completed at 2018-08-12 17:58:17"
```

Calculate Accuracy of Cottonwood and Willow Logisitic Models - All Vars

Calculate Cottonwood and Willow Aggregated Data Logisitic Model Accuracy - All Vars

Find best threshold for Cottonwood and Willow using all aggregated data.

```r
result = calcLogisticModelAccuracy (forestTrain$Cottonwood_Willow, CotWil_Agg_Train_predict,
                       0.0, 1, 10, "Cotton_Wil", "Other", 1,1)
```

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=82.2%, Spec=99.3%, Sens^2+Spec^2=1.662"
## [1] "Thresh=0.2, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=69.4%, Spec=99.6%, Sens^2+Spec^2=1.475"
## [1] "Thresh=0.3, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=58.8%, Spec=99.7%, Sens^2+Spec^2=1.342"
## [1] "Thresh=0.4, Accuracy=99.6%, BaseAcc(Other)=99.5%, Sens=49.1%, Spec=99.8%, Sens^2+Spec^2=1.238"
## [1] "Thresh=0.5, Accuracy=99.6%, BaseAcc(Other)=99.5%, Sens=41.2%, Spec=99.9%, Sens^2+Spec^2=1.168"
## [1] "Thresh=0.6, Accuracy=99.6%, BaseAcc(Other)=99.5%, Sens=33%, Spec=99.9%, Sens^2+Spec^2=1.108"
## [1] "Thresh=0.7, Accuracy=99.6%, BaseAcc(Other)=99.5%, Sens=23.4%, Spec=99.9%, Sens^2+Spec^2=1.054"
## [1] "Thresh=0.8, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=11.5%, Spec=99.9%, Sens^2+Spec^2=1.013"
## [1] "Thresh=0.9, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=4.5%, Spec=99.9%, Sens^2+Spec^2=1.002"
## [1] "Thresh=1, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Best Sensitivity_Specificity threshold= 0.1 inc= 0.1"
## [1] "========================================"
## [1] "start= 0 end= 0.2 inc= 0.01"
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.01, Accuracy=97.1%, BaseAcc(Other)=99.5%, Sens=97.8%, Spec=97.1%, Sens^2+Spec^2=1.9"
## [1] "Thresh=0.02, Accuracy=97.8%, BaseAcc(Other)=99.5%, Sens=95.4%, Spec=97.8%, Sens^2+Spec^2=1.868"
## [1] "Thresh=0.03, Accuracy=98.2%, BaseAcc(Other)=99.5%, Sens=93.1%, Spec=98.3%, Sens^2+Spec^2=1.835"
## [1] "Thresh=0.04, Accuracy=98.5%, BaseAcc(Other)=99.5%, Sens=91.1%, Spec=98.5%, Sens^2+Spec^2=1.802"
## [1] "Thresh=0.05, Accuracy=98.7%, BaseAcc(Other)=99.5%, Sens=88.8%, Spec=98.7%, Sens^2+Spec^2=1.764"
## [1] "Thresh=0.06, Accuracy=98.8%, BaseAcc(Other)=99.5%, Sens=86.8%, Spec=98.9%, Sens^2+Spec^2=1.733"
## [1] "Thresh=0.07, Accuracy=98.9%, BaseAcc(Other)=99.5%, Sens=85.4%, Spec=99%, Sens^2+Spec^2=1.711"
## [1] "Thresh=0.08, Accuracy=99%, BaseAcc(Other)=99.5%, Sens=84%, Spec=99.1%, Sens^2+Spec^2=1.689"
## [1] "Thresh=0.09, Accuracy=99.1%, BaseAcc(Other)=99.5%, Sens=83%, Spec=99.2%, Sens^2+Spec^2=1.675"
## [1] "Thresh=0.1, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=82.2%, Spec=99.3%, Sens^2+Spec^2=1.662"
## [1] "Thresh=0.11, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=80.7%, Spec=99.3%, Sens^2+Spec^2=1.639"
## [1] "Thresh=0.12, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=79.4%, Spec=99.4%, Sens^2+Spec^2=1.619"
## [1] "Thresh=0.13, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=77.9%, Spec=99.4%, Sens^2+Spec^2=1.596"
## [1] "Thresh=0.14, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=76.7%, Spec=99.5%, Sens^2+Spec^2=1.579"
## [1] "Thresh=0.15, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=75.5%, Spec=99.5%, Sens^2+Spec^2=1.561"
## [1] "Thresh=0.16, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=74.7%, Spec=99.5%, Sens^2+Spec^2=1.55"
## [1] "Thresh=0.17, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=73.1%, Spec=99.5%, Sens^2+Spec^2=1.527"
## [1] "Thresh=0.18, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=71.7%, Spec=99.6%, Sens^2+Spec^2=1.507"
## [1] "Thresh=0.19, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=70.6%, Spec=99.6%, Sens^2+Spec^2=1.492"
## [1] "Best Sensitivity_Specificity threshold= 0.01 inc= 0.01"
## [1] "========================================"
## [1] "start= 0 end= 0.02 inc= 0.001"
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.001, Accuracy=94.7%, BaseAcc(Other)=99.5%, Sens=99.9%, Spec=94.7%, Sens^2+Spec^2=1.897
## [1] "Thresh=0.002, Accuracy=95.2%, BaseAcc(Other)=99.5%, Sens=99.5%, Spec=95.2%, Sens^2+Spec^2=1.899
## [1] "Thresh=0.003, Accuracy=95.6%, BaseAcc(Other)=99.5%, Sens=99.3%, Spec=95.6%, Sens^2+Spec^2=1.901
## [1] "Thresh=0.004, Accuracy=96%, BaseAcc(Other)=99.5%, Sens=99.2%, Spec=96%, Sens^2+Spec^2=1.906"
```

```
## [1] "Thresh=0.005, Accuracy=96.2%, BaseAcc(Other)=99.5%, Sens=99%, Spec=96.2%, Sens^2+Spec^2=1.907"
## [1] "Thresh=0.006, Accuracy=96.5%, BaseAcc(Other)=99.5%, Sens=98.8%, Spec=96.5%, Sens^2+Spec^2=1.908
## [1] "Thresh=0.007, Accuracy=96.6%, BaseAcc(Other)=99.5%, Sens=98.5%, Spec=96.6%, Sens^2+Spec^2=1.905
## [1] "Thresh=0.008, Accuracy=96.8%, BaseAcc(Other)=99.5%, Sens=98.1%, Spec=96.8%, Sens^2+Spec^2=1.9"
## [1] "Thresh=0.009, Accuracy=96.9%, BaseAcc(Other)=99.5%, Sens=97.9%, Spec=96.9%, Sens^2+Spec^2=1.9"
## [1] "Thresh=0.01, Accuracy=97.1%, BaseAcc(Other)=99.5%, Sens=97.8%, Spec=97.1%, Sens^2+Spec^2=1.9"
## [1] "Thresh=0.011, Accuracy=97.2%, BaseAcc(Other)=99.5%, Sens=97.6%, Spec=97.2%, Sens^2+Spec^2=1.897
## [1] "Thresh=0.012, Accuracy=97.3%, BaseAcc(Other)=99.5%, Sens=97.2%, Spec=97.3%, Sens^2+Spec^2=1.893
## [1] "Thresh=0.013, Accuracy=97.4%, BaseAcc(Other)=99.5%, Sens=96.9%, Spec=97.4%, Sens^2+Spec^2=1.888
## [1] "Thresh=0.014, Accuracy=97.4%, BaseAcc(Other)=99.5%, Sens=96.7%, Spec=97.4%, Sens^2+Spec^2=1.886
## [1] "Thresh=0.015, Accuracy=97.5%, BaseAcc(Other)=99.5%, Sens=96.4%, Spec=97.5%, Sens^2+Spec^2=1.881
## [1] "Thresh=0.016, Accuracy=97.6%, BaseAcc(Other)=99.5%, Sens=96.2%, Spec=97.6%, Sens^2+Spec^2=1.878
## [1] "Thresh=0.017, Accuracy=97.7%, BaseAcc(Other)=99.5%, Sens=95.9%, Spec=97.7%, Sens^2+Spec^2=1.876
## [1] "Thresh=0.018, Accuracy=97.7%, BaseAcc(Other)=99.5%, Sens=95.7%, Spec=97.7%, Sens^2+Spec^2=1.872
## [1] "Thresh=0.019, Accuracy=97.8%, BaseAcc(Other)=99.5%, Sens=95.5%, Spec=97.8%, Sens^2+Spec^2=1.869
## [1] "========================================"
## [1] "Best Threshold=0.006"
## [1] "Best Sensitivity_Specificity=1.90852508067791"
```

```r
curThresh = as.numeric(result[bestThreshIndex])
CotWil_Agg_All_threshold = curThresh
```

The accuracy for the best threshold on the training set for Cottonwood and Willow using all aggregated data is shown below.

```r
result = calcLogisticModelAccuracy (forestTrain$Cottonwood_Willow, CotWil_Agg_Train_predict,
                       curThresh, curThresh, 1, "Cotton_Wil", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.006"
## [1] "predicted performance="
##                      Predicted
## Actual                FALSE=Predict:Other TRUE=Predict:Cotton_Wil
##   0=Actual:Other          390629 (TN)         14157 (FP)
##   1=Actual:Cotton_Wil     22 (FN)             1901 (TP)
## [1] "Sensitivity= 0.988559542381695 (True positive rate of Cotton_Wil = TP/(TP+FN) = 1901 /( 1901 + 2
## [1] "Specificity= 0.965025964336711 (True negative rate of Other = TN/(TN+FP) = 390629 /( 390629 + 14
## [1] "Sens^2+Spec^2=1.908"
## [1] "Baseline (Other) Accuracy=0.995271"
## [1] "Logistic Accuracy=0.965137"
```

The accuracy for the best threshold on the testing set for Cottonwood and Willow using all aggregated data is shown below.

```r
result = calcLogisticModelAccuracy (forestTest$Cottonwood_Willow, CotWil_Agg_Test_predict,
                       curThresh, curThresh, 1, "Cotton_Wil", "Other", 3,
                       saveFile=saveFileName, desc="Cottonwd/Willow All Aggregate Vars",
                       AIC=CotWil_Agg_All_aic, AUC=CotWil_Agg_All_ROC_AUC)
```

```
## [1] "Model Performance for threshold= 0.006"
## [1] "predicted performance="
##                      Predicted
## Actual                FALSE=Predict:Other TRUE=Predict:Cotton_Wil
##   0=Actual:Other          167374 (TN)         6105 (FP)
##   1=Actual:Cotton_Wil     17 (FN)             807 (TP)
## [1] "Sensitivity= 0.979368932038835 (True positive rate of Cotton_Wil = TP/(TP+FN) = 807 /( 807 + 17
## [1] "Specificity= 0.964808420615752 (True negative rate of Other = TN/(TN+FP) = 167374 /( 167374 + 61
## [1] "Sens^2+Spec^2=1.89"
```

```
## [1] "Baseline (Other) Accuracy=0.995272"
## [1] "Logistic Accuracy=0.964877"
```

```
# retVal = c(modelPerformance, sensitivity,specificity) # TN, FN, FP, TP, sens, spec
# c(funcStat,accuracy,baseline,retVal)
list[RC, CotWil_Agg_All_model_acc, CotWil_Agg_All_baseline_acc,
    TN, FN, FP, TP, CotWil_Agg_All_sens, CotWil_Agg_All_spec] <- result
if (RC != "OK") {
  print(paste("Error - terminating:",RC))
  knitr:knit_exit()
}
CotWil_Agg_All_model_acc = as.integer(as.numeric(CotWil_Agg_All_model_acc)*1000)/10
CotWil_Agg_All_baseline_acc = as.integer(as.numeric(CotWil_Agg_All_baseline_acc)*1000)/10
CotWil_Agg_All_sens = as.integer(as.numeric(CotWil_Agg_All_sens)*1000)/10
CotWil_Agg_All_spec = as.integer(as.numeric(CotWil_Agg_All_spec)*1000)/10
```

Calculate Cottonwood and Willow Individuated Data Logisitic Model Accuracy - All Vars

Find best threshold for Cottonwood and Willow using all individuated data.

```
result = calcLogisticModelAccuracy (forestTrain$Cottonwood_Willow, CotWil_Ind_Train_predict,
                        0.0, 1, 10, "Cotton_Wil", "Other", 1,1)
```

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=82.2%, Spec=99.3%, Sens^2+Spec^2=1.662"
## [1] "Thresh=0.2, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=69.4%, Spec=99.6%, Sens^2+Spec^2=1.475"
## [1] "Thresh=0.3, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=58.8%, Spec=99.7%, Sens^2+Spec^2=1.342"
## [1] "Thresh=0.4, Accuracy=99.6%, BaseAcc(Other)=99.5%, Sens=49%, Spec=99.8%, Sens^2+Spec^2=1.237"
## [1] "Thresh=0.5, Accuracy=99.6%, BaseAcc(Other)=99.5%, Sens=41.2%, Spec=99.9%, Sens^2+Spec^2=1.168"
## [1] "Thresh=0.6, Accuracy=99.6%, BaseAcc(Other)=99.5%, Sens=33%, Spec=99.9%, Sens^2+Spec^2=1.108"
## [1] "Thresh=0.7, Accuracy=99.6%, BaseAcc(Other)=99.5%, Sens=23.4%, Spec=99.9%, Sens^2+Spec^2=1.054"
## [1] "Thresh=0.8, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=11.5%, Spec=99.9%, Sens^2+Spec^2=1.013"
## [1] "Thresh=0.9, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=4.5%, Spec=99.9%, Sens^2+Spec^2=1.002"
## [1] "Thresh=1, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Best Sensitivity_Specificity threshold= 0.1 inc= 0.1"
## [1] "========================================"
## [1] "start= 0 end= 0.2 inc= 0.01"
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.01, Accuracy=97.1%, BaseAcc(Other)=99.5%, Sens=97.8%, Spec=97.1%, Sens^2+Spec^2=1.9"
## [1] "Thresh=0.02, Accuracy=97.8%, BaseAcc(Other)=99.5%, Sens=95.4%, Spec=97.8%, Sens^2+Spec^2=1.868"
## [1] "Thresh=0.03, Accuracy=98.2%, BaseAcc(Other)=99.5%, Sens=93.1%, Spec=98.3%, Sens^2+Spec^2=1.834"
## [1] "Thresh=0.04, Accuracy=98.5%, BaseAcc(Other)=99.5%, Sens=91.1%, Spec=98.5%, Sens^2+Spec^2=1.802"
## [1] "Thresh=0.05, Accuracy=98.7%, BaseAcc(Other)=99.5%, Sens=88.7%, Spec=98.7%, Sens^2+Spec^2=1.763"
## [1] "Thresh=0.06, Accuracy=98.8%, BaseAcc(Other)=99.5%, Sens=86.8%, Spec=98.9%, Sens^2+Spec^2=1.733"
## [1] "Thresh=0.07, Accuracy=98.9%, BaseAcc(Other)=99.5%, Sens=85.4%, Spec=99%, Sens^2+Spec^2=1.711"
## [1] "Thresh=0.08, Accuracy=99%, BaseAcc(Other)=99.5%, Sens=84%, Spec=99.1%, Sens^2+Spec^2=1.689"
## [1] "Thresh=0.09, Accuracy=99.1%, BaseAcc(Other)=99.5%, Sens=83%, Spec=99.2%, Sens^2+Spec^2=1.675"
## [1] "Thresh=0.1, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=82.2%, Spec=99.3%, Sens^2+Spec^2=1.662"
## [1] "Thresh=0.11, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=80.7%, Spec=99.3%, Sens^2+Spec^2=1.639"
## [1] "Thresh=0.12, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=79.4%, Spec=99.4%, Sens^2+Spec^2=1.618"
## [1] "Thresh=0.13, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=77.9%, Spec=99.4%, Sens^2+Spec^2=1.596"
## [1] "Thresh=0.14, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=76.7%, Spec=99.5%, Sens^2+Spec^2=1.579"
## [1] "Thresh=0.15, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=75.5%, Spec=99.5%, Sens^2+Spec^2=1.561"
```

```
## [1] "Thresh=0.16, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=74.7%, Spec=99.5%, Sens^2+Spec^2=1.55"
## [1] "Thresh=0.17, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=73.1%, Spec=99.5%, Sens^2+Spec^2=1.527"
## [1] "Thresh=0.18, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=71.7%, Spec=99.6%, Sens^2+Spec^2=1.507"
## [1] "Thresh=0.19, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=70.6%, Spec=99.6%, Sens^2+Spec^2=1.492"
## [1] "Best Sensitivity_Specificity threshold= 0.01 inc= 0.01"
## [1] "========================================"
## [1] "start= 0 end= 0.02 inc= 0.001"
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.001, Accuracy=94.7%, BaseAcc(Other)=99.5%, Sens=99.9%, Spec=94.7%, Sens^2+Spec^2=1.897
## [1] "Thresh=0.002, Accuracy=95.2%, BaseAcc(Other)=99.5%, Sens=99.5%, Spec=95.2%, Sens^2+Spec^2=1.899
## [1] "Thresh=0.003, Accuracy=95.6%, BaseAcc(Other)=99.5%, Sens=99.3%, Spec=95.6%, Sens^2+Spec^2=1.901
## [1] "Thresh=0.004, Accuracy=96%, BaseAcc(Other)=99.5%, Sens=99.2%, Spec=96%, Sens^2+Spec^2=1.906"
## [1] "Thresh=0.005, Accuracy=96.2%, BaseAcc(Other)=99.5%, Sens=99%, Spec=96.2%, Sens^2+Spec^2=1.907"
## [1] "Thresh=0.006, Accuracy=96.5%, BaseAcc(Other)=99.5%, Sens=98.8%, Spec=96.5%, Sens^2+Spec^2=1.908
## [1] "Thresh=0.007, Accuracy=96.6%, BaseAcc(Other)=99.5%, Sens=98.4%, Spec=96.6%, Sens^2+Spec^2=1.904
## [1] "Thresh=0.008, Accuracy=96.8%, BaseAcc(Other)=99.5%, Sens=98%, Spec=96.8%, Sens^2+Spec^2=1.899"
## [1] "Thresh=0.009, Accuracy=96.9%, BaseAcc(Other)=99.5%, Sens=97.9%, Spec=96.9%, Sens^2+Spec^2=1.9"
## [1] "Thresh=0.01, Accuracy=97.1%, BaseAcc(Other)=99.5%, Sens=97.8%, Spec=97.1%, Sens^2+Spec^2=1.9"
## [1] "Thresh=0.011, Accuracy=97.2%, BaseAcc(Other)=99.5%, Sens=97.6%, Spec=97.2%, Sens^2+Spec^2=1.897
## [1] "Thresh=0.012, Accuracy=97.3%, BaseAcc(Other)=99.5%, Sens=97.2%, Spec=97.3%, Sens^2+Spec^2=1.893
## [1] "Thresh=0.013, Accuracy=97.4%, BaseAcc(Other)=99.5%, Sens=96.9%, Spec=97.4%, Sens^2+Spec^2=1.888
## [1] "Thresh=0.014, Accuracy=97.4%, BaseAcc(Other)=99.5%, Sens=96.7%, Spec=97.4%, Sens^2+Spec^2=1.886
## [1] "Thresh=0.015, Accuracy=97.5%, BaseAcc(Other)=99.5%, Sens=96.4%, Spec=97.5%, Sens^2+Spec^2=1.881
## [1] "Thresh=0.016, Accuracy=97.6%, BaseAcc(Other)=99.5%, Sens=96.1%, Spec=97.6%, Sens^2+Spec^2=1.877
## [1] "Thresh=0.017, Accuracy=97.7%, BaseAcc(Other)=99.5%, Sens=95.9%, Spec=97.7%, Sens^2+Spec^2=1.876
## [1] "Thresh=0.018, Accuracy=97.7%, BaseAcc(Other)=99.5%, Sens=95.7%, Spec=97.7%, Sens^2+Spec^2=1.872
## [1] "Thresh=0.019, Accuracy=97.8%, BaseAcc(Other)=99.5%, Sens=95.5%, Spec=97.8%, Sens^2+Spec^2=1.869
## [1] "========================================"
## [1] "Best Threshold=0.006"
## [1] "Best Sensitivity_Specificity=1.9085536893761"
```

```r
curThresh = as.numeric(result[bestThreshIndex])
CotWil_Ind_All_threshold = curThresh
```

The accuracy for the best threshold on the training set for Cottonwood and Willow using all individuated data is shown below.

```r
result = calcLogisticModelAccuracy (forestTrain$Cottonwood_Willow, CotWil_Ind_Train_predict,
                        curThresh, curThresh, 1, "Cotton_Wil", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.006"
## [1] "predicted performance="
##                       Predicted
## Actual                 FALSE=Predict:Other TRUE=Predict:Cotton_Wil
##   0=Actual:Other           390635 (TN)         14151 (FP)
##   1=Actual:Cotton_Wil      22 (FN)             1901 (TP)
## [1] "Sensitivity= 0.988559542381695 (True positive rate of Cotton_Wil = TP/(TP+FN) = 1901 /( 1901 + 2
## [1] "Specificity= 0.96504078698374 (True negative rate of Other = TN/(TN+FP) = 390635 /( 390635 + 141
## [1] "Sens^2+Spec^2=1.908"
## [1] "Baseline (Other) Accuracy=0.995271"
## [1] "Logistic Accuracy=0.965151"
```

The accuracy for the best threshold on the testing set for Cottonwood and Willow using all individuated data is shown below.

```
result = calcLogisticModelAccuracy (forestTest$Cottonwood_Willow, CotWil_Ind_Test_predict,
                        curThresh, curThresh, 1, "Cotton_Wil", "Other", 3,
                        saveFile=saveFileName, desc="Cottonwd/Willow All Individualized Vars",
                        AIC=CotWil_Ind_All_aic, AUC=CotWil_Ind_All_ROC_AUC)
```

```
## [1] "Model Performance for threshold= 0.006"
## [1] "predicted performance="
##                      Predicted
## Actual                FALSE=Predict:Other TRUE=Predict:Cotton_Wil
##   0=Actual:Other          167380 (TN)         6099 (FP)
##   1=Actual:Cotton_Wil     17 (FN)             807 (TP)
## [1] "Sensitivity= 0.979368932038835 (True positive rate of Cotton_Wil = TP/(TP+FN) = 807 /( 807 + 17
## [1] "Specificity= 0.964843006934557 (True negative rate of Other = TN/(TN+FP) = 167380 /( 167380 + 60
## [1] "Sens^2+Spec^2=1.89"
## [1] "Baseline (Other) Accuracy=0.995272"
## [1] "Logistic Accuracy=0.964911"
```

```
list[RC, CotWil_Ind_All_model_acc, CotWil_Ind_All_baseline_acc,
      TN, FN, FP, TP, CotWil_Ind_All_sens, CotWil_Ind_All_spec] <- result
  if (RC != "OK") {
    print(paste("Error - terminating:",RC))
    knitr:knit_exit()
  }
  CotWil_Ind_All_model_acc = as.integer(as.numeric(CotWil_Ind_All_model_acc)*1000)/10
  CotWil_Ind_All_baseline_acc = as.integer(as.numeric(CotWil_Ind_All_baseline_acc)*1000)/10
  CotWil_Ind_All_sens = as.integer(as.numeric(CotWil_Ind_All_sens)*1000)/10
  CotWil_Ind_All_spec = as.integer(as.numeric(CotWil_Ind_All_spec)*1000)/10
```

The Cottonwood and Willow aggregated model accuracy on the test data is 77.15% compared to 77.12% for the individuated data model, essentially identical. Both are ~ 14% better than the baseline model.

Cottonwood and Willow Logistic Regression - Significant Variables

Create Cottonwood and Willow Logistic Model - Sig Vars

Now create the logistic model for the Aggregated Soil data using just the significant variables and compare to the previous models.

Cottonwood and Willow Logistic Model using Significant Aggregated Data

Variables that have been removed are commented out in the code below.

```
  CotWil_Agg_LogMod =
    glm(Cottonwood_Willow ~
          Elev +     # Elevation in meters of cell
          Aspect +   # Direction in degrees slope faces
          Slope +    # Slope / steepness of hill in degrees (0 to 90)
          H2OHD +    # Horizontal distance in meters to nearest water
          H2OVD +    # Vertical distance in meters to nearest water
          RoadHD +   # Horizontal distance in meters to nearest road
          FirePtHD + # Horizontal distance in meters to nearest fire point
          Shade9AM +
          # Shade12PM + Shade3PM + # Amount of shade at 9am, 12pm and 3pm - removed 2nd pass
          # Wilderness areas:
```

```r
            # RWwild + NEwild + CMwild + CPwild +
          # Aggregated Soil type:
            # ST01 + ST02 + ST03 +
            ST04 +
            # ST05 + ST06 + ST07 +
            # ST08 + ST09 + # removed 2nd pass
            ST10 + ST11 +
            # ST12 +  # removed 2nd pass
            # ST13 + ST14 + ST15 +
            ST16 + ST17
            # ST18 + ST19 + ST20 + # removed 2nd pass
            # ST21 + ST22 + ST23 + ST24 + ST25 +  # removed 2nd pass
            # ST26 + ST27 + ST28 + ST29 + ST30 + # removed 2nd pass
            # ST31 + ST32 + ST33 +  # removed 2nd pass
            # ST34 + ST35 +
            # ST36 +  # removed 2nd pass
            # ST37 +
            # ST38 + ST39 + # removed 2nd pass
            # ST40
          ,
          data=forestTrain, family=binomial)

CotWil_Agg_Sig_LogMod = CotWil_Agg_LogMod
save("CotWil_Agg_Sig_LogMod", file="CotWil_Agg_Sig_LogMod.Rdata")

CotWil_Agg_Sig_aic<-as.integer(CotWil_Agg_LogMod$aic)
CotWil_Agg_Sig_aic
```

```
## [1] 10876
```

Check the coefficients of the Cottonwood and Willow model using significant aggregated data.

```r
summary(CotWil_Agg_LogMod)
```

```
##
## Call:
## glm(formula = Cottonwood_Willow ~ Elev + Aspect + Slope + H2OHD +
##     H2OVD + RoadHD + FirePtHD + Shade9AM + ST04 + ST10 + ST11 +
##     ST16 + ST17, family = binomial, data = forestTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3396  -0.0294  -0.0104  -0.0033   4.3176
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  9.970e+00  5.131e-01  19.430  < 2e-16 ***
## Elev        -1.008e-02  1.986e-04 -50.784  < 2e-16 ***
## Aspect       3.589e-03  3.655e-04   9.818  < 2e-16 ***
## Slope       -6.116e-02  3.787e-03 -16.149  < 2e-16 ***
## H2OHD       -8.036e-03  5.077e-04 -15.828  < 2e-16 ***
## H2OVD        1.861e-02  1.097e-03  16.968  < 2e-16 ***
## RoadHD       7.496e-04  4.438e-05  16.890  < 2e-16 ***
## FirePtHD    -7.445e-05  3.875e-05  -1.921 0.054685 .
## Shade9AM     4.688e-02  1.532e-03  30.591  < 2e-16 ***
```

```
## ST04          2.376e-01  1.044e-01   2.277 0.022815 *  
## ST10         -3.648e-01  9.916e-02  -3.679 0.000234 ***
## ST11         -4.568e-01  2.504e-01  -1.824 0.068130 .  
## ST16          9.395e-01  1.769e-01   5.311 1.09e-07 ***
## ST17          1.744e+00  1.027e-01  16.982  < 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: 24429  on 406708  degrees of freedom
## Residual deviance: 10848  on 406695  degrees of freedom
## AIC: 10876
## 
## Number of Fisher Scoring iterations: 11
```

The intercept looks much more reasonable. Some soil types that were significant previously are no longer significant.

Cottonwood and Willow Logistic Model using Significant Individuated Data

Create a logistic model for the significant individuated variables.

Again, the non-significant variables have been commented out.

```
CotWil_Ind_LogMod =
  glm(Cottonwood_Willow ~
        Elev +     # Elevation in meters of cell
        Aspect +   # Direction in degrees slope faces
        Slope +    # Slope / steepness of hill in degrees (0 to 90)
        H2OHD +    # Horizontal distance in meters to nearest water
        H2OVD +    # Vertical distance in meters to nearest water
        RoadHD +   # Horizontal distance in meters to nearest road
        FirePtHD + # Horizontal distance in meters to nearest fire point
        Shade9AM +
        # Shade12PM + Shade3PM + # Amount of shade at 9am, 12pm and 3pm    # removed 2nd pass
        # Wilderness areas:
          # RWwild + NEwild + CMwild + CPwild +
        # Climate Zone:
        # ClimateName +
          # Montane_low + Montane +
          # SubAlpine + Alpine +      # removed 2nd pass
          # Dry + Non_Dry +
        # Geology Zone:
        # GeoName +
          # Alluvium + Glacial +     # removed 2nd pass
          # Sed_mix + Ign_Meta +
        # Soil Family:
          # Aquolis_cmplx +     # removed 2nd pass
          # Argiborolis_Pachic +
          # Borohemists_cmplx + Bross +     # removed 2nd pass
          # Bullwark + Bullwark_Cmplx + Catamount + Catamount_cmplx +     # removed 2nd pass
          # Cathedral + Como +
          # Cryaquepts_cmplx + Cryaquepts_Typic + Cryaquolls +     # removed 2nd pass
          # Cryaquolls_cmplx + Cryaquolls_Typic + Cryaquolls_Typic_cmplx +     # removed 2nd pass
```

```
          # Cryoborolis_cmplx +
          # Cryorthents +     # removed 2nd pass
          # Cryorthents_cmplx + Cryumbrepts + Cryumbrepts_cmplx + Gateview +
          # Gothic + Granile + Haploborolis +
          # Legault +     # removed 2nd pass
          # Legault_cmplx +
          # Leighcan + Leighcan_cmplx + Leighcan_warm +     # removed 2nd pass
          # Moran + Ratake + Ratake_cmplx + Rogert + Supervisor_Limber_cmplx +
          # Troutville + Unspecified + Vanet + Wetmore +
        # Soil Rock composition:
          # Bouldery_ext +
          # Rock_Land +    # removed 2nd pass
          # Rock_Land_cmplx + Rock_Outcrop +
          Rock_Outcrop_cmplx ,
          # Rubbly + Stony + Stony_extreme + Stony_very + Till_Substratum ,
        data=forestTrain, family=binomial)

CotWil_Ind_Sig_LogMod = CotWil_Ind_LogMod
save("CotWil_Ind_Sig_LogMod", file="CotWil_Ind_Sig_LogMod.Rdata")

CotWil_Ind_Sig_aic<-as.integer(CotWil_Ind_LogMod$aic)
CotWil_Ind_Sig_aic
```

```
## [1] 11206
```

```
summary(CotWil_Ind_LogMod)
```

```
##
## Call:
## glm(formula = Cottonwood_Willow ~ Elev + Aspect + Slope + H2OHD +
##     H2OVD + RoadHD + FirePtHD + Shade9AM + Rock_Outcrop_cmplx,
##     family = binomial, data = forestTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2365  -0.0298  -0.0097  -0.0028   4.4298
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.085e+01  4.915e-01  22.079   <2e-16 ***
## Elev               -1.038e-02  1.923e-04 -54.002   <2e-16 ***
## Aspect              4.123e-03  3.498e-04  11.787   <2e-16 ***
## Slope              -7.595e-02  3.793e-03 -20.021   <2e-16 ***
## H2OHD              -9.542e-03  5.144e-04 -18.549   <2e-16 ***
## H2OVD               1.962e-02  1.113e-03  17.631   <2e-16 ***
## RoadHD              7.782e-04  4.269e-05  18.228   <2e-16 ***
## FirePtHD           -3.780e-05  3.675e-05  -1.028    0.304
## Shade9AM            4.753e-02  1.372e-03  34.640   <2e-16 ***
## Rock_Outcrop_cmplx  6.708e-02  7.152e-02   0.938    0.348
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 24429  on 406708  degrees of freedom
## Residual deviance: 11187  on 406699  degrees of freedom
## AIC: 11207
##
## Number of Fisher Scoring iterations: 11
```

Again the intercept looks much better. Also a few variables have become non-significant.

Predict Cottonwood and Willow Logistic Model Probabilities - Sig Vars

Cottonwood and Willow Probabilities using Significant Aggregated Data

Predict the probability of Cottonwood and Willow for aggregated Data - significant variables.

```
# Predict Cottonwood and Willow Agg Data - significant variables

  CotWil_Agg_Train_predict= predict(CotWil_Agg_LogMod, type="response")
  summary(CotWil_Agg_Train_predict)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0000059 0.0000560 0.0047282 0.0004521 0.9352251
```

```
  CotWil_Agg_Test_predict= predict(CotWil_Agg_LogMod, type="response",newdata=forestTest)
  summary(CotWil_Agg_Test_predict)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0000056 0.0000559 0.0045486 0.0004453 0.9274409
```

Cottonwood and Willow Probabilities using Significant Individuated Data

Predict the probability of Cottonwood_Willow using significant Individuated Data.

```
  CotWil_Ind_Train_predict= predict(CotWil_Ind_LogMod, type="response")
  summary(CotWil_Ind_Train_predict)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0000043 0.0000492 0.0047282 0.0004666 0.9179898
```

```
  CotWil_Ind_Test_predict= predict(CotWil_Ind_LogMod, type="response",newdata=forestTest)
  summary(CotWil_Ind_Test_predict)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0000041 0.0000492 0.0045691 0.0004617 0.8525628
```

```
  print(paste("ROCR graph 2 completed at",curTime))
```

```
## [1] "ROCR graph 2 completed at 2018-08-12 17:58:17"
```

Cottonwood and Willow Receiver Operating Characteristic (ROC) - Sig Vars

Look at the True Positive and False Positive rates based on threshold value.

```
  if (calcROC) {
    ROCpred_CotWil_Agg = prediction(CotWil_Agg_Train_predict, forestTrain$Cottonwood_Willow)
    summary(ROCpred_CotWil_Agg)

    ROCperf_CotWil_Agg = performance(ROCpred_CotWil_Agg, "tpr", "fpr")
    summary(ROCperf_CotWil_Agg)
```

```
    CotWil_Agg_Sig_ROC_AUC = as.numeric(performance(ROCpred_CotWil_Agg, "auc")@y.values)
    CotWil_Agg_Sig_ROC_AUC=as.integer(as.numeric(CotWil_Agg_Sig_ROC_AUC)*1000)/10
    CotWil_Agg_Sig_ROC_AUC

    jpeg(filename="Fig-ROCR_perf_CotWil_Agg_Sig.jpg")
    plot(ROCperf_CotWil_Agg, colorize=TRUE, print.cutoffs.at=seq(0,1,0.1), text.adj=c(-0.2,1.7))
    dev.off()
  } else {
    CotWil_Agg_Sig_ROC_AUC = 83.7
  }
```

```
## pdf
##   2
```

```
  if (calcROC) {
    curTime=Sys.time()
    print(paste("ROCR graph 2 started at",curTime))

    ROCpred_CotWil_Ind = prediction(CotWil_Ind_Train_predict, forestTrain$Cottonwood_Willow)
    summary(ROCpred_CotWil_Ind)

    ROCperf_CotWil_Ind = performance(ROCpred_CotWil_Ind, "tpr", "fpr")
    summary(ROCperf_CotWil_Ind)

    CotWil_Ind_Sig_ROC_AUC = as.numeric(performance(ROCpred_CotWil_Ind, "auc")@y.values)
    CotWil_Ind_Sig_ROC_AUC=as.integer(as.numeric(CotWil_Ind_Sig_ROC_AUC)*1000)/10
    CotWil_Ind_Sig_ROC_AUC

    jpeg(filename="Fig-ROC_perf_CotWil_Ind_Sig.jpg")
    plot(ROCperf_CotWil_Ind, colorize=TRUE, print.cutoffs.at=seq(0,1,0.1), text.adj=c(-0.2,1.7))
    dev.off()
  } else {
    CotWil_Ind_Sig_ROC_AUC = 83.8
  }
```

```
## [1] "ROCR graph 2 started at 2018-08-12 18:04:18"
```

```
## pdf
##   2
```

The threshold graphs are essentially identical. This is making me think that there is not much difference between the two models. The AIC score for the Soil Type model is AIC: 351676 and for the individuated variables is: AIC: 351839. The Soil type model AIC score is 0.046% better than the individuated model.

Calculate Accuracy of Cottonwood and Willow Logisitic Model - Sig Vars

Calculate Cottonwood and Willow Aggregated Data Logisitic Model Accuracy - Significant Vars

Find best Cottonwood and Willow threshold for Aggregated Data using significant variables.

```
result = calcLogisticModelAccuracy (forestTrain$Cottonwood_Willow, CotWil_Agg_Train_predict,
                        0.0, 1, 10, "Cotton_Wil", "Other", 1,1)
```

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
```

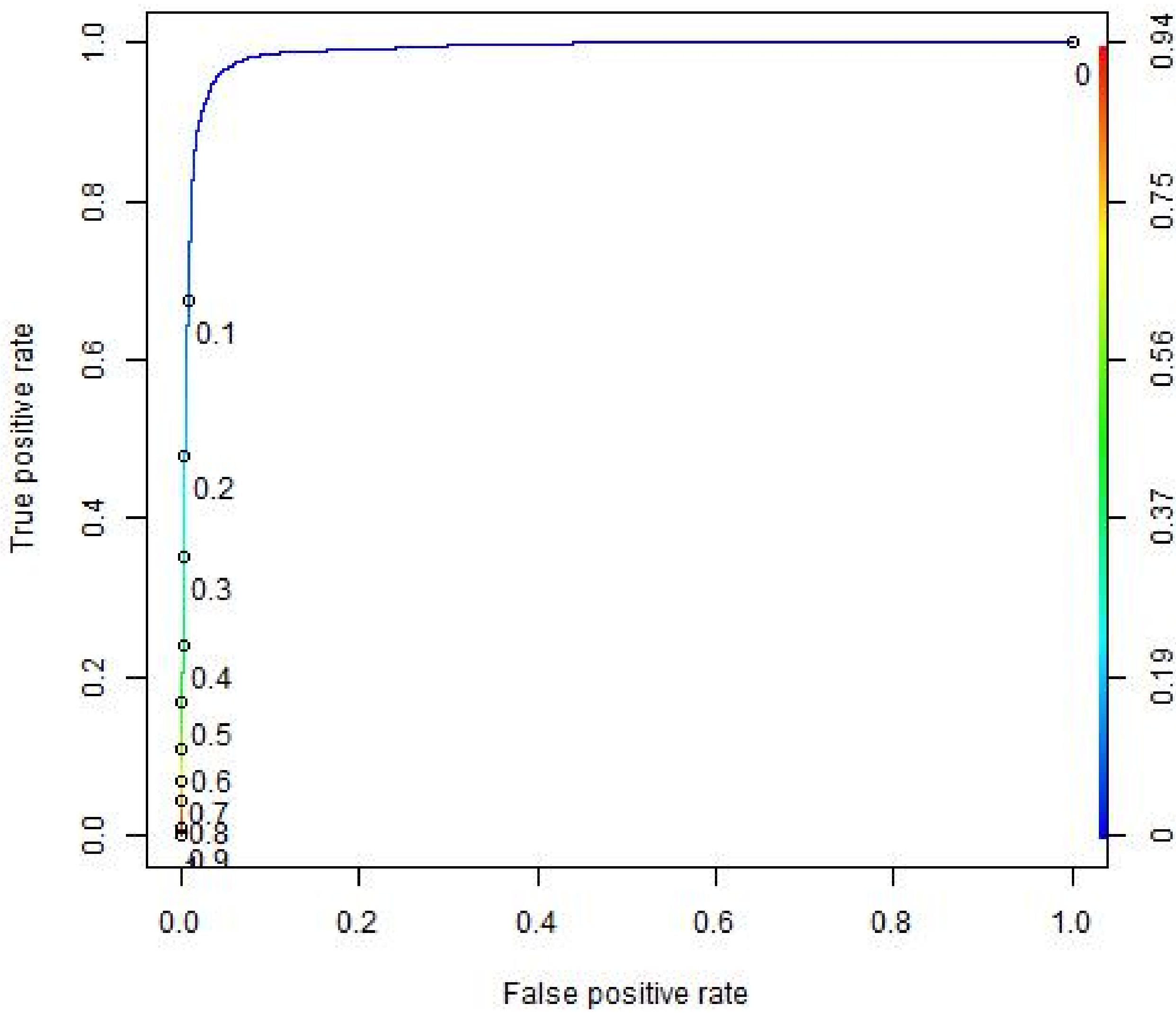

Figure 3: Cottonwood and Willow ROC for Aggregated Significant Data

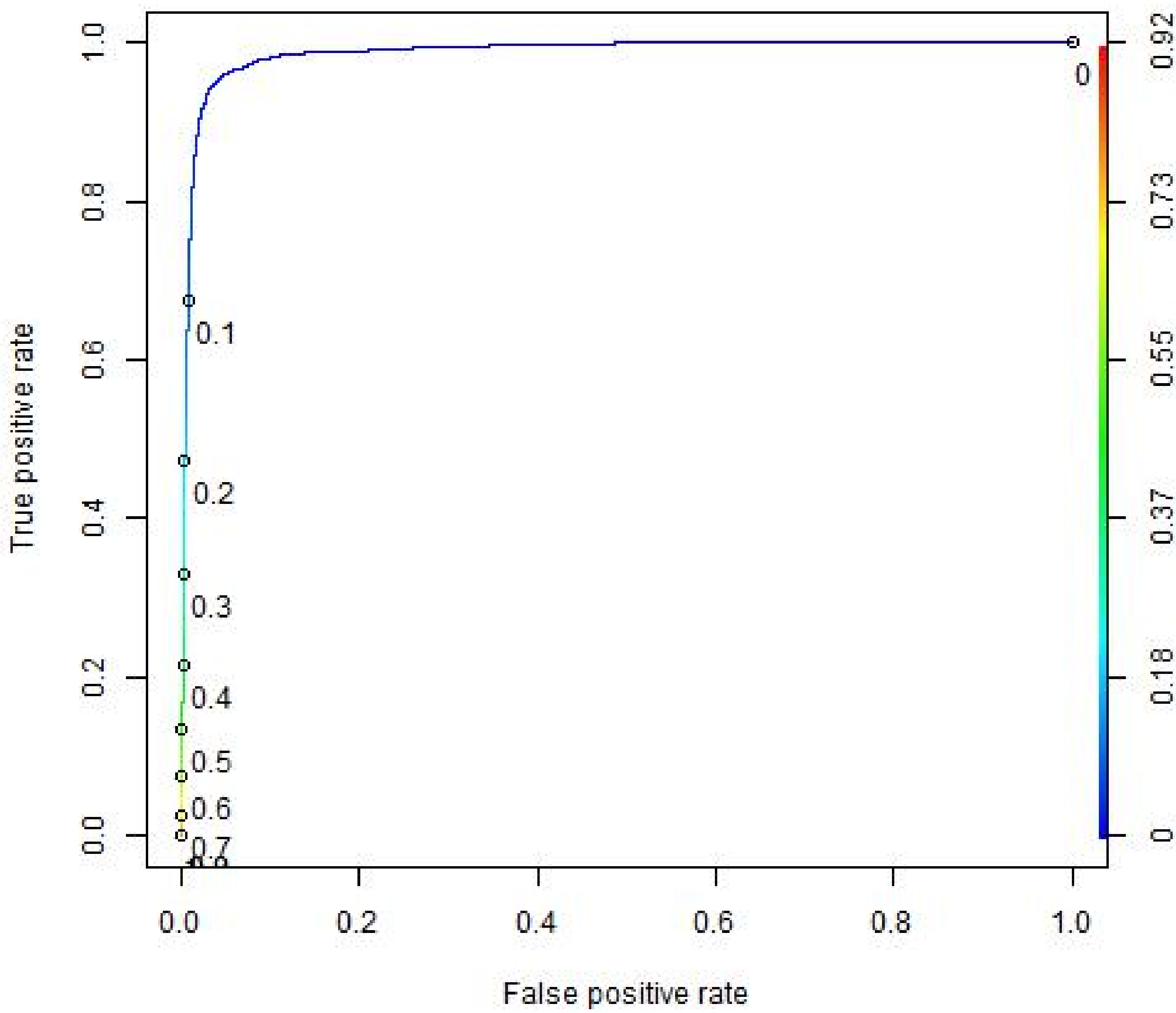

Figure 4: Cottonwood and Willow ROC for Individuated Significant Data

```
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=99%, BaseAcc(Other)=99.5%, Sens=67.5%, Spec=99.2%, Sens^2+Spec^2=1.441"
## [1] "Thresh=0.2, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=47.8%, Spec=99.5%, Sens^2+Spec^2=1.22"
## [1] "Thresh=0.3, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=35.2%, Spec=99.7%, Sens^2+Spec^2=1.119"
## [1] "Thresh=0.4, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=23.8%, Spec=99.8%, Sens^2+Spec^2=1.053"
## [1] "Thresh=0.5, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=16.6%, Spec=99.8%, Sens^2+Spec^2=1.025"
## [1] "Thresh=0.6, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=10.8%, Spec=99.9%, Sens^2+Spec^2=1.01"
## [1] "Thresh=0.7, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=6.9%, Spec=99.9%, Sens^2+Spec^2=1.004"
## [1] "Thresh=0.8, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=4.2%, Spec=99.9%, Sens^2+Spec^2=1.001"
## [1] "Thresh=0.9, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=1%, Spec=99.9%, Sens^2+Spec^2=1"
## [1] "Thresh=1, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Best Sensitivity_Specificity threshold= 0.1 inc= 0.1"
## [1] "========================================"
## [1] "start= 0 end= 0.2 inc= 0.01"
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.01, Accuracy=96.2%, BaseAcc(Other)=99.5%, Sens=95.1%, Spec=96.2%, Sens^2+Spec^2=1.832"
## [1] "Thresh=0.02, Accuracy=97.5%, BaseAcc(Other)=99.5%, Sens=91.6%, Spec=97.6%, Sens^2+Spec^2=1.793"
## [1] "Thresh=0.03, Accuracy=98.1%, BaseAcc(Other)=99.5%, Sens=89.2%, Spec=98.1%, Sens^2+Spec^2=1.76"
## [1] "Thresh=0.04, Accuracy=98.4%, BaseAcc(Other)=99.5%, Sens=86%, Spec=98.5%, Sens^2+Spec^2=1.71"
## [1] "Thresh=0.05, Accuracy=98.6%, BaseAcc(Other)=99.5%, Sens=82.8%, Spec=98.7%, Sens^2+Spec^2=1.662"
## [1] "Thresh=0.06, Accuracy=98.8%, BaseAcc(Other)=99.5%, Sens=79.1%, Spec=98.9%, Sens^2+Spec^2=1.604"
## [1] "Thresh=0.07, Accuracy=98.9%, BaseAcc(Other)=99.5%, Sens=75.7%, Spec=99%, Sens^2+Spec^2=1.554"
## [1] "Thresh=0.08, Accuracy=98.9%, BaseAcc(Other)=99.5%, Sens=72.7%, Spec=99.1%, Sens^2+Spec^2=1.511"
## [1] "Thresh=0.09, Accuracy=99%, BaseAcc(Other)=99.5%, Sens=70.1%, Spec=99.1%, Sens^2+Spec^2=1.475"
## [1] "Thresh=0.1, Accuracy=99%, BaseAcc(Other)=99.5%, Sens=67.5%, Spec=99.2%, Sens^2+Spec^2=1.441"
## [1] "Thresh=0.11, Accuracy=99.1%, BaseAcc(Other)=99.5%, Sens=65.1%, Spec=99.2%, Sens^2+Spec^2=1.409"
## [1] "Thresh=0.12, Accuracy=99.1%, BaseAcc(Other)=99.5%, Sens=62.1%, Spec=99.3%, Sens^2+Spec^2=1.373"
## [1] "Thresh=0.13, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=60.6%, Spec=99.3%, Sens^2+Spec^2=1.355"
## [1] "Thresh=0.14, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=58.8%, Spec=99.4%, Sens^2+Spec^2=1.334"
## [1] "Thresh=0.15, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=56.7%, Spec=99.4%, Sens^2+Spec^2=1.311"
## [1] "Thresh=0.16, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=54.6%, Spec=99.4%, Sens^2+Spec^2=1.288"
## [1] "Thresh=0.17, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=52.7%, Spec=99.5%, Sens^2+Spec^2=1.268"
## [1] "Thresh=0.18, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=51%, Spec=99.5%, Sens^2+Spec^2=1.251"
## [1] "Thresh=0.19, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=48.9%, Spec=99.5%, Sens^2+Spec^2=1.23"
## [1] "Best Sensitivity_Specificity threshold= 0.01 inc= 0.01"
## [1] "========================================"
## [1] "start= 0 end= 0.02 inc= 0.001"
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.001, Accuracy=83.2%, BaseAcc(Other)=99.5%, Sens=99%, Spec=83.1%, Sens^2+Spec^2=1.673"
## [1] "Thresh=0.002, Accuracy=88.7%, BaseAcc(Other)=99.5%, Sens=98.8%, Spec=88.6%, Sens^2+Spec^2=1.762
## [1] "Thresh=0.003, Accuracy=91.2%, BaseAcc(Other)=99.5%, Sens=98.3%, Spec=91.2%, Sens^2+Spec^2=1.8"
## [1] "Thresh=0.004, Accuracy=92.8%, BaseAcc(Other)=99.5%, Sens=98%, Spec=92.8%, Sens^2+Spec^2=1.823"
## [1] "Thresh=0.005, Accuracy=93.8%, BaseAcc(Other)=99.5%, Sens=97.5%, Spec=93.8%, Sens^2+Spec^2=1.831
## [1] "Thresh=0.006, Accuracy=94.6%, BaseAcc(Other)=99.5%, Sens=97%, Spec=94.6%, Sens^2+Spec^2=1.837"
## [1] "Thresh=0.007, Accuracy=95.1%, BaseAcc(Other)=99.5%, Sens=96.6%, Spec=95.1%, Sens^2+Spec^2=1.839
## [1] "Thresh=0.008, Accuracy=95.6%, BaseAcc(Other)=99.5%, Sens=96.2%, Spec=95.6%, Sens^2+Spec^2=1.84"
## [1] "Thresh=0.009, Accuracy=95.9%, BaseAcc(Other)=99.5%, Sens=95.6%, Spec=95.9%, Sens^2+Spec^2=1.835
## [1] "Thresh=0.01, Accuracy=96.2%, BaseAcc(Other)=99.5%, Sens=95.1%, Spec=96.2%, Sens^2+Spec^2=1.832"
## [1] "Thresh=0.011, Accuracy=96.4%, BaseAcc(Other)=99.5%, Sens=95%, Spec=96.4%, Sens^2+Spec^2=1.833"
## [1] "Thresh=0.012, Accuracy=96.6%, BaseAcc(Other)=99.5%, Sens=94.5%, Spec=96.6%, Sens^2+Spec^2=1.829
## [1] "Thresh=0.013, Accuracy=96.8%, BaseAcc(Other)=99.5%, Sens=93.6%, Spec=96.8%, Sens^2+Spec^2=1.815
## [1] "Thresh=0.014, Accuracy=96.9%, BaseAcc(Other)=99.5%, Sens=93.3%, Spec=97%, Sens^2+Spec^2=1.813"
## [1] "Thresh=0.015, Accuracy=97.1%, BaseAcc(Other)=99.5%, Sens=92.9%, Spec=97.1%, Sens^2+Spec^2=1.807
## [1] "Thresh=0.016, Accuracy=97.2%, BaseAcc(Other)=99.5%, Sens=92.6%, Spec=97.2%, Sens^2+Spec^2=1.804
```

```
## [1] "Thresh=0.017, Accuracy=97.3%, BaseAcc(Other)=99.5%, Sens=92.3%, Spec=97.3%, Sens^2+Spec^2=1.8"
## [1] "Thresh=0.018, Accuracy=97.4%, BaseAcc(Other)=99.5%, Sens=92.1%, Spec=97.4%, Sens^2+Spec^2=1.799
## [1] "Thresh=0.019, Accuracy=97.5%, BaseAcc(Other)=99.5%, Sens=92%, Spec=97.5%, Sens^2+Spec^2=1.798"
## [1] "========================================"
## [1] "Best Threshold=0.008"
## [1] "Best Sensitivity_Specificity=1.84093948848621"
```

```
curThresh = as.numeric(result[bestThreshIndex])
CotWil_Agg_Sig_threshold = curThresh
```

The accuracy for the best threshold on the training set for Cottonwood and Willow using significant aggregated data is shown below.

```
result = calcLogisticModelAccuracy (forestTrain$Cottonwood_Willow, CotWil_Agg_Train_predict,
                       curThresh, curThresh, 1, "Cotton_Wil", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.008"
## [1] "predicted performance="
##                       Predicted
## Actual                 FALSE=Predict:Other TRUE=Predict:Cotton_Wil
##   0=Actual:Other           387078 (TN)         17708 (FP)
##   1=Actual:Cotton_Wil      72 (FN)             1851 (TP)
## [1] "Sensitivity= 0.962558502340094 (True positive rate of Cotton_Wil = TP/(TP+FN) = 1851 /( 1851 + 
## [1] "Specificity= 0.956253427737125 (True negative rate of Other = TN/(TN+FP) = 387078 /( 387078 + 1
## [1] "Sens^2+Spec^2=1.84"
## [1] "Baseline (Other) Accuracy=0.995271"
## [1] "Logistic Accuracy=0.956283"
```

The accuracy for the best threshold on the testing set for Cottonwood and Willow using significant aggregated data is shown below.

```
result = calcLogisticModelAccuracy (forestTest$Cottonwood_Willow, CotWil_Agg_Test_predict,
                       curThresh, curThresh, 1, "Cotton_Wil", "Other", 3,
                       saveFile=saveFileName, desc="Cottonwd/Willow Sig Aggregate Vars",
                       AIC=CotWil_Agg_Sig_aic, AUC=CotWil_Agg_Sig_ROC_AUC)
```

```
## [1] "Model Performance for threshold= 0.008"
## [1] "predicted performance="
##                       Predicted
## Actual                 FALSE=Predict:Other TRUE=Predict:Cotton_Wil
##   0=Actual:Other           165917 (TN)         7562 (FP)
##   1=Actual:Cotton_Wil      48 (FN)             776 (TP)
## [1] "Sensitivity= 0.941747572815534 (True positive rate of Cotton_Wil = TP/(TP+FN) = 776 /( 776 + 48
## [1] "Specificity= 0.956409709532566 (True negative rate of Other = TN/(TN+FP) = 165917 /( 165917 + 7
## [1] "Sens^2+Spec^2=1.801"
## [1] "Baseline (Other) Accuracy=0.995272"
## [1] "Logistic Accuracy=0.95634"
```

```
list[RC, CotWil_Agg_Sig_model_acc, CotWil_Agg_Sig_baseline_acc,
      TN, FN, FP, TP, CotWil_Agg_Sig_sens, CotWil_Agg_Sig_spec] <- result
  if (RC != "OK") {
    print(paste("Error - terminating:",RC))
    knitr:knit_exit()
  }
  CotWil_Agg_Sig_model_acc = as.integer(as.numeric(CotWil_Agg_Sig_model_acc)*1000)/10
  CotWil_Agg_Sig_baseline_acc = as.integer(as.numeric(CotWil_Agg_Sig_baseline_acc)*1000)/10
  CotWil_Agg_Sig_sens = as.integer(as.numeric(CotWil_Agg_Sig_sens)*1000)/10
```

```
CotWil_Agg_Sig_spec = as.integer(as.numeric(CotWil_Agg_Sig_spec)*1000)/10
```

Calculate Cottonwood and Willow Individuated Data Logisitic Model Accuracy - Significant Vars

Find best Cottonwood and Willow threshold for Inividuated Data using significant variables.

```
result = calcLogisticModelAccuracy (forestTrain$Cottonwood_Willow, CotWil_Ind_Train_predict,
                        0.0, 1, 10, "Cotton_Wil", "Other", 1,1)
```

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=99.1%, BaseAcc(Other)=99.5%, Sens=67.4%, Spec=99.2%, Sens^2+Spec^2=1.44"
## [1] "Thresh=0.2, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=47.3%, Spec=99.5%, Sens^2+Spec^2=1.216"
## [1] "Thresh=0.3, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=32.8%, Spec=99.7%, Sens^2+Spec^2=1.102"
## [1] "Thresh=0.4, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=21.3%, Spec=99.8%, Sens^2+Spec^2=1.042"
## [1] "Thresh=0.5, Accuracy=99.4%, BaseAcc(Other)=99.5%, Sens=13.5%, Spec=99.8%, Sens^2+Spec^2=1.016"
## [1] "Thresh=0.6, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=7.4%, Spec=99.9%, Sens^2+Spec^2=1.004"
## [1] "Thresh=0.7, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=2.6%, Spec=99.9%, Sens^2+Spec^2=1"
## [1] "Thresh=0.8, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=0%, Spec=99.9%, Sens^2+Spec^2=0.999"
## [1] "Thresh=0.9, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=0%, Spec=99.9%, Sens^2+Spec^2=-2"
## [1] "Thresh=1, Accuracy=99.5%, BaseAcc(Other)=99.5%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Best Sensitivity_Specificity threshold= 0.1 inc= 0.1"
## [1] "========================================"
## [1] "start= 0 end= 0.2 inc= 0.01"
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.01, Accuracy=95.9%, BaseAcc(Other)=99.5%, Sens=95.1%, Spec=95.9%, Sens^2+Spec^2=1.826"
## [1] "Thresh=0.02, Accuracy=97.5%, BaseAcc(Other)=99.5%, Sens=92%, Spec=97.5%, Sens^2+Spec^2=1.799"
## [1] "Thresh=0.03, Accuracy=98.1%, BaseAcc(Other)=99.5%, Sens=88.6%, Spec=98.1%, Sens^2+Spec^2=1.749"
## [1] "Thresh=0.04, Accuracy=98.4%, BaseAcc(Other)=99.5%, Sens=85.7%, Spec=98.5%, Sens^2+Spec^2=1.705"
## [1] "Thresh=0.05, Accuracy=98.6%, BaseAcc(Other)=99.5%, Sens=82.2%, Spec=98.7%, Sens^2+Spec^2=1.65"
## [1] "Thresh=0.06, Accuracy=98.8%, BaseAcc(Other)=99.5%, Sens=78.8%, Spec=98.9%, Sens^2+Spec^2=1.6"
## [1] "Thresh=0.07, Accuracy=98.9%, BaseAcc(Other)=99.5%, Sens=75.4%, Spec=99%, Sens^2+Spec^2=1.549"
## [1] "Thresh=0.08, Accuracy=98.9%, BaseAcc(Other)=99.5%, Sens=72.5%, Spec=99.1%, Sens^2+Spec^2=1.509"
## [1] "Thresh=0.09, Accuracy=99%, BaseAcc(Other)=99.5%, Sens=70.3%, Spec=99.1%, Sens^2+Spec^2=1.478"
## [1] "Thresh=0.1, Accuracy=99.1%, BaseAcc(Other)=99.5%, Sens=67.4%, Spec=99.2%, Sens^2+Spec^2=1.44"
## [1] "Thresh=0.11, Accuracy=99.1%, BaseAcc(Other)=99.5%, Sens=64.4%, Spec=99.3%, Sens^2+Spec^2=1.401"
## [1] "Thresh=0.12, Accuracy=99.1%, BaseAcc(Other)=99.5%, Sens=61.8%, Spec=99.3%, Sens^2+Spec^2=1.369"
## [1] "Thresh=0.13, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=59.3%, Spec=99.3%, Sens^2+Spec^2=1.34"
## [1] "Thresh=0.14, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=57.4%, Spec=99.4%, Sens^2+Spec^2=1.318"
## [1] "Thresh=0.15, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=55%, Spec=99.4%, Sens^2+Spec^2=1.291"
## [1] "Thresh=0.16, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=53.4%, Spec=99.4%, Sens^2+Spec^2=1.275"
## [1] "Thresh=0.17, Accuracy=99.2%, BaseAcc(Other)=99.5%, Sens=52%, Spec=99.5%, Sens^2+Spec^2=1.261"
## [1] "Thresh=0.18, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=50.5%, Spec=99.5%, Sens^2+Spec^2=1.246"
## [1] "Thresh=0.19, Accuracy=99.3%, BaseAcc(Other)=99.5%, Sens=49%, Spec=99.5%, Sens^2+Spec^2=1.232"
## [1] "Best Sensitivity_Specificity threshold= 0.01 inc= 0.01"
## [1] "========================================"
## [1] "start= 0 end= 0.02 inc= 0.001"
## [1] "Thresh=0, Accuracy=0.4%, BaseAcc(Other)=99.5%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.001, Accuracy=82.5%, BaseAcc(Other)=99.5%, Sens=98.9%, Spec=82.4%, Sens^2+Spec^2=1.657
## [1] "Thresh=0.002, Accuracy=87.8%, BaseAcc(Other)=99.5%, Sens=98.4%, Spec=87.8%, Sens^2+Spec^2=1.741
## [1] "Thresh=0.003, Accuracy=90.5%, BaseAcc(Other)=99.5%, Sens=97.9%, Spec=90.5%, Sens^2+Spec^2=1.778
## [1] "Thresh=0.004, Accuracy=92.1%, BaseAcc(Other)=99.5%, Sens=97.3%, Spec=92.1%, Sens^2+Spec^2=1.796
```

```
## [1] "Thresh=0.005, Accuracy=93.3%, BaseAcc(Other)=99.5%, Sens=96.7%, Spec=93.2%, Sens^2+Spec^2=1.806
## [1] "Thresh=0.006, Accuracy=94.1%, BaseAcc(Other)=99.5%, Sens=96.5%, Spec=94.1%, Sens^2+Spec^2=1.818
## [1] "Thresh=0.007, Accuracy=94.7%, BaseAcc(Other)=99.5%, Sens=96.2%, Spec=94.7%, Sens^2+Spec^2=1.824
## [1] "Thresh=0.008, Accuracy=95.2%, BaseAcc(Other)=99.5%, Sens=95.9%, Spec=95.2%, Sens^2+Spec^2=1.828
## [1] "Thresh=0.009, Accuracy=95.6%, BaseAcc(Other)=99.5%, Sens=95.5%, Spec=95.6%, Sens^2+Spec^2=1.826
## [1] "Thresh=0.01, Accuracy=95.9%, BaseAcc(Other)=99.5%, Sens=95.1%, Spec=95.9%, Sens^2+Spec^2=1.826"
## [1] "Thresh=0.011, Accuracy=96.2%, BaseAcc(Other)=99.5%, Sens=94.7%, Spec=96.2%, Sens^2+Spec^2=1.824
## [1] "Thresh=0.012, Accuracy=96.4%, BaseAcc(Other)=99.5%, Sens=94.5%, Spec=96.4%, Sens^2+Spec^2=1.824
## [1] "Thresh=0.013, Accuracy=96.6%, BaseAcc(Other)=99.5%, Sens=94.3%, Spec=96.6%, Sens^2+Spec^2=1.823
## [1] "Thresh=0.014, Accuracy=96.8%, BaseAcc(Other)=99.5%, Sens=94.1%, Spec=96.8%, Sens^2+Spec^2=1.824
## [1] "Thresh=0.015, Accuracy=96.9%, BaseAcc(Other)=99.5%, Sens=93.8%, Spec=96.9%, Sens^2+Spec^2=1.821
## [1] "Thresh=0.016, Accuracy=97%, BaseAcc(Other)=99.5%, Sens=93.4%, Spec=97.1%, Sens^2+Spec^2=1.817"
## [1] "Thresh=0.017, Accuracy=97.2%, BaseAcc(Other)=99.5%, Sens=93%, Spec=97.2%, Sens^2+Spec^2=1.811"
## [1] "Thresh=0.018, Accuracy=97.3%, BaseAcc(Other)=99.5%, Sens=92.6%, Spec=97.3%, Sens^2+Spec^2=1.805
## [1] "Thresh=0.019, Accuracy=97.4%, BaseAcc(Other)=99.5%, Sens=92.4%, Spec=97.4%, Sens^2+Spec^2=1.803
## [1] "========================================"
## [1] "Best Threshold=0.008"
## [1] "Best Sensitivity_Specificity=1.82841561231725"
```

```
curThresh = as.numeric(result[bestThreshIndex])
CotWil_Ind_Sig_threshold = curThresh
```

The accuracy for the best threshold on the training set for Cottonwood and Willow using significant individuated data is shown below.

```
result = calcLogisticModelAccuracy (forestTrain$Cottonwood_Willow, CotWil_Ind_Train_predict,
                       curThresh, curThresh, 1, "Cotton_Wil", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.008"
## [1] "predicted performance="
##                      Predicted
## Actual                FALSE=Predict:Other TRUE=Predict:Cotton_Wil
##   0=Actual:Other          385482 (TN)         19304 (FP)
##   1=Actual:Cotton_Wil     77 (FN)             1846 (TP)
## [1] "Sensitivity= 0.959958398335933 (True positive rate of Cotton_Wil = TP/(TP+FN) = 1846 /( 1846 +
## [1] "Specificity= 0.952310603627596 (True negative rate of Other = TN/(TN+FP) = 385482 /( 385482 + 1
## [1] "Sens^2+Spec^2=1.828"
## [1] "Baseline (Other) Accuracy=0.995271"
## [1] "Logistic Accuracy=0.952346"
```

The accuracy for the best threshold on the testing set for Cottonwood and Willow using significant individuated data is shown below.

```
result = calcLogisticModelAccuracy (forestTest$Cottonwood_Willow, CotWil_Ind_Test_predict,
                       curThresh, curThresh, 1, "Cotton_Wil", "Other", 3,
                       saveFile=saveFileName, desc="Cottonwd/Willow Sig Individualized Vars",
                       AIC=CotWil_Ind_Sig_aic, AUC=CotWil_Ind_Sig_ROC_AUC)
```

```
## [1] "Model Performance for threshold= 0.008"
## [1] "predicted performance="
##                      Predicted
## Actual                FALSE=Predict:Other TRUE=Predict:Cotton_Wil
##   0=Actual:Other          165181 (TN)         8298 (FP)
##   1=Actual:Cotton_Wil     54 (FN)             770 (TP)
## [1] "Sensitivity= 0.934466019417476 (True positive rate of Cotton_Wil = TP/(TP+FN) = 770 /( 770 + 54
## [1] "Specificity= 0.952167121092466 (True negative rate of Other = TN/(TN+FP) = 165181 /( 165181 + 8
## [1] "Sens^2+Spec^2=1.779"
```

```
## [1] "Baseline (Other) Accuracy=0.995272"
## [1] "Logistic Accuracy=0.952083"
```

```
list[RC, CotWil_Ind_Sig_model_acc, CotWil_Ind_Sig_baseline_acc,
      TN, FN, FP, TP, CotWil_Ind_Sig_sens, CotWil_Ind_Sig_spec] <- result
  if (RC != "OK") {
    print(paste("Error - terminating:",RC))
    knitr:knit_exit()
  }
  CotWil_Ind_Sig_model_acc = as.integer(as.numeric(CotWil_Ind_Sig_model_acc)*1000)/10
  CotWil_Ind_Sig_baseline_acc = as.integer(as.numeric(CotWil_Ind_Sig_baseline_acc)*1000)/10
  CotWil_Ind_Sig_sens = as.integer(as.numeric(CotWil_Ind_Sig_sens)*1000)/10
  CotWil_Ind_Sig_spec = as.integer(as.numeric(CotWil_Ind_Sig_spec)*1000)/10

############# End End End End End End End End End End End End ##################
```

The accuracy of the models is shown below:

Logistic Model	Accuracy	Sens	Spec	AIC	AUC	Threshold
Cottonwood/Willow Aggregate All Vars	96.4%	97.9%	96.4%	7932	99.5%	0.006
Cottonwood/Willow Individual All Vars	96.4%	97.9%	96.4%	7938	99.5%	0.006
Cottonwood/Willow Aggregate Sig Vars	95.6%	94.1%	95.6%	10876	98.8%	0.008
Cottonwood/Willow Individual Sig Vars	95.2%	93.4%	95.2%	11206	98.7%	0.008

There is a slight degradation in the accuracy with insignificant variables eliminated, but not by much.

Conclusion

It is beginning to look like there is no advantage to dis-aggregating the Soil Type variables into their component parts. I was hoping there would be some improvement by allowing the individual variables to be "more finely" tuned. There is probably a mathematical explanation that proves there is no advantage of breaking out aggregated variables. I have to think about that more.

The logistic regression results for Spruce and Fir are 7% better than the original paper this project was modeled after. These tests need to be done for the remaining 6 forest cover types to see how regression does overall.

```
  curTime=Sys.time()
  print(paste("Forest Cover Logistic script ended at",curTime))
```

```
## [1] "Forest Cover Logistic script ended at 2018-08-12 18:08:23"
```