Capstone Data Logistic Regression - Predict Aspen

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 14:42:41"
```

```
#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"
#infile="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcover_clean.csv"
#infile="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcoversmall_clean_full.csv"
```

```r
#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 14:43:29"
```

```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)
```

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"]
LodgeAndSpruceAndFirPct<-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.

```r
source("logisticAccuracy.R") # for function calcLogisticModelAccuracy
#save("calcLogisticModelAccuracy", file="logisticAccuracy.Rdata")
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 #################
```

Aspen Logistic Regression

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

Aspen Logistic Regression - All Variables

Create Aspen Logistic Model - All Vars

Create the Aspen logistic model for the Aggregated Soil data using all independent variables.

Aspen 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("Aspen aggregated Logistic Model Calculation started at",curTime))
```

```
## [1] "Aspen aggregated Logistic Model Calculation started at 2018-08-12 14:43:31"
```

```r
Aspen_Agg_LogMod =
  glm(Aspen ~
        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: algorithm did not converge

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

```r
Aspen_Agg_All_LogMod = Aspen_Agg_LogMod
save("Aspen_Agg_All_LogMod", file="Aspen_Agg_All_LogMod.Rdata")

Aspen_Agg_All_aic<-as.integer(Aspen_Agg_LogMod$aic)
Aspen_Agg_All_aic
```

```
## [1] 1737968
```

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

```
## [1] "Aspen aggregated Logistic Model Calculation completed at 2018-08-12 14:46:26"
```

Check the coefficients for the Aspen model using all aggregated data.

```r
summary(Aspen_Agg_LogMod)
```

```
##
## Call:
## glm(formula = Aspen ~ 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
## -8.4904  -0.2036  -0.1122  -0.0362   3.4481
##
## Coefficients:
##                Estimate Std. Error    z value Pr(>|z|)
```

```
## (Intercept)  2.270e+12  9.374e+11      2.421 0.015460 *  
## Elev        -3.458e-03  1.015e-04    -34.069  < 2e-16 ***
## Aspect       1.819e-03  1.967e-04      9.246  < 2e-16 ***
## Slope        1.806e-02  5.255e-03      3.437 0.000588 ***
## H2OHD        9.604e-04  1.103e-04      8.708  < 2e-16 ***
## H2OVD        4.848e-03  3.184e-04     15.223  < 2e-16 ***
## RoadHD      -3.258e-04  1.564e-05    -20.836  < 2e-16 ***
## FirePtHD    -8.782e-05  1.537e-05     -5.712 1.11e-08 ***
## Shade9AM     1.493e-02  5.308e-03      2.813 0.004911 ** 
## Shade12PM   -9.459e-03  4.376e-03     -2.162 0.030645 *  
## Shade3PM     5.816e-03  4.319e-03      1.347 0.178111    
## RWwild      -2.324e+12  9.272e+11     -2.507 0.012190 *  
## NEwild      -2.324e+12  9.272e+11     -2.507 0.012190 *  
## CMwild      -2.324e+12  9.272e+11     -2.507 0.012190 *  
## CPwild      -2.324e+12  9.272e+11     -2.507 0.012190 *  
## ST01         5.425e+10  9.650e+10      0.562 0.574007    
## ST02         5.425e+10  9.650e+10      0.562 0.574007    
## ST03         5.425e+10  9.650e+10      0.562 0.574007    
## ST04         5.425e+10  9.650e+10      0.562 0.574007    
## ST05         5.425e+10  9.650e+10      0.562 0.574007    
## ST06         5.425e+10  9.650e+10      0.562 0.574007    
## ST07        -4.504e+15  9.650e+10 -46670.015  < 2e-16 ***
## ST08        -4.504e+15  9.650e+10 -46670.011  < 2e-16 ***
## ST09         5.425e+10  9.650e+10      0.562 0.574007    
## ST10         5.425e+10  9.650e+10      0.562 0.574007    
## ST11         5.425e+10  9.650e+10      0.562 0.574007    
## ST12        -4.504e+15  9.650e+10 -46670.012  < 2e-16 ***
## ST13         5.425e+10  9.650e+10      0.562 0.574007    
## ST14        -4.504e+15  9.650e+10 -46670.057  < 2e-16 ***
## ST15        -4.504e+15  9.650e+10 -46670.671  < 2e-16 ***
## ST16         5.425e+10  9.650e+10      0.562 0.574007    
## ST17         5.425e+10  9.650e+10      0.562 0.574007    
## ST18         5.425e+10  9.650e+10      0.562 0.574007    
## ST19         5.425e+10  9.650e+10      0.562 0.574007    
## ST20         5.425e+10  9.650e+10      0.562 0.574007    
## ST21        -4.504e+15  9.650e+10 -46670.006  < 2e-16 ***
## ST22         5.425e+10  9.650e+10      0.562 0.574007    
## ST23         5.425e+10  9.650e+10      0.562 0.574007    
## ST24         5.425e+10  9.650e+10      0.562 0.574007    
## ST25         5.425e+10  9.650e+10      0.562 0.574007    
## ST26         5.425e+10  9.650e+10      0.562 0.574007    
## ST27         5.425e+10  9.650e+10      0.562 0.574007    
## ST28         5.425e+10  9.650e+10      0.562 0.574007    
## ST29         5.425e+10  9.650e+10      0.562 0.574007    
## ST30         5.425e+10  9.650e+10      0.562 0.574007    
## ST31         5.425e+10  9.650e+10      0.562 0.574007    
## ST32         5.425e+10  9.650e+10      0.562 0.574007    
## ST33         5.425e+10  9.650e+10      0.562 0.574007    
## ST34         5.425e+10  9.650e+10      0.562 0.574007    
## ST35         5.425e+10  9.650e+10      0.562 0.574007    
## ST36         5.425e+10  9.650e+10      0.562 0.574007    
## ST37         5.425e+10  9.650e+10      0.562 0.574007    
## ST38         5.425e+10  9.650e+10      0.562 0.574007    
## ST39         5.425e+10  9.650e+10      0.562 0.574007    
```

```
## ST40         5.425e+10  9.650e+10      0.562 0.574007
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance:   67859  on 406708  degrees of freedom
## Residual deviance: 1737858  on 406654  degrees of freedom
## AIC: 1737968
##
## Number of Fisher Scoring iterations: 25
```

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.

Aspen 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("Aspen Individual Logistic Model Calculation started at",curTime))
```

```
## [1] "Aspen Individual Logistic Model Calculation started at 2018-08-12 14:46:26"
```

```
Aspen_Ind_LogMod =
  glm(Aspen ~
        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:
          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:
```

```r
            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: algorithm did not converge

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

```r
Aspen_Ind_All_LogMod = Aspen_Ind_LogMod
save("Aspen_Ind_All_LogMod", file="Aspen_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 <-------

Aspen_Ind_All_aic<-as.integer(Aspen_Ind_LogMod$aic)
Aspen_Ind_All_aic
```

```
## [1] 568029
```

```r
summary(Aspen_Ind_LogMod)
```

```
##
## Call:
## glm(formula = Aspen ~ 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
## -8.4904  -0.1968  -0.1009  -0.0296   3.6877
##
## Coefficients: (18 not defined because of singularities)
##                          Estimate Std. Error   z value Pr(>|z|)
## (Intercept)             3.165e+12  1.135e+12     2.789  0.00528 **
## Elev                   -3.067e-03  1.048e-04   -29.274  < 2e-16 ***
## Aspect                  2.258e-03  1.966e-04    11.490  < 2e-16 ***
## Slope                   2.467e-02  5.433e-03     4.540 5.63e-06 ***
## H2OHD                  -9.693e-04  1.108e-04    -8.746  < 2e-16 ***
## H2OVD                   5.269e-03  3.195e-04    16.488  < 2e-16 ***
## RoadHD                 -4.581e-04  1.571e-05   -29.157  < 2e-16 ***
```

```
## FirePtHD              -9.346e-05  1.629e-05    -5.738 9.58e-09 ***
## Shade9AM               2.989e-02  5.564e-03     5.372 7.80e-08 ***
## Shade12PM             -1.183e-02  4.575e-03    -2.585  0.00974 **
## Shade3PM               9.718e-03  4.542e-03     2.140  0.03239 *
## RWwild                -1.608e+12  2.933e+11    -5.482 4.20e-08 ***
## NEwild                -1.608e+12  2.933e+11    -5.482 4.20e-08 ***
## CMwild                -1.608e+12  2.933e+11    -5.482 4.20e-08 ***
## CPwild                -1.608e+12  2.933e+11    -5.482 4.20e-08 ***
## Montane_low            1.314e+13  2.099e+12     6.260 3.85e-10 ***
## Montane                1.470e+13  2.290e+12     6.418 1.38e-10 ***
## SubAlpine             -1.557e+12  1.112e+12    -1.401  0.16126
## Alpine                -1.557e+12  1.112e+12    -1.401  0.16126
## Dry                   -1.557e+12  1.112e+12    -1.401  0.16126
## Non_Dry               -1.470e+13  2.290e+12    -6.418 1.38e-10 ***
## Alluvium              -7.500e-01  1.169e+01    -0.064  0.94883
## Glacial               -1.233e+13  2.747e+12    -4.490 7.13e-06 ***
## Sed_mix               -4.518e+15  2.290e+12 -1972.856  < 2e-16 ***
## Ign_Meta                      NA         NA        NA       NA
## Aquolis_cmplx          4.183e+05  3.889e+07     0.011  0.99142
## Argiborolis_Pachic            NA         NA        NA       NA
## Borohemists_cmplx      2.412e+01  1.022e+04     0.002  0.99812
## Bross                  1.146e+03  7.195e+06     0.000  0.99987
## Bullwark              -1.557e+12  1.112e+12    -1.401  0.16126
## Bullwark_Cmplx        -1.557e+12  1.112e+12    -1.401  0.16126
## Catamount              2.151e-01  3.044e-01     0.707  0.47981
## Catamount_cmplx        1.361e+00  1.796e-01     7.579 3.47e-14 ***
## Cathedral              3.896e-02  3.278e+02     0.000  0.99991
## Como                   3.084e+00  1.108e+01     0.278  0.78086
## Cryaquepts_cmplx       6.503e+00  1.994e+01     0.326  0.74430
## Cryaquepts_Typic      -1.233e+13  2.747e+12    -4.490 7.13e-06 ***
## Cryaquolls            -1.007e+00  1.169e+01    -0.086  0.93132
## Cryaquolls_cmplx       1.245e+00  1.168e+01     0.107  0.91517
## Cryaquolls_Typic      -1.932e+01  1.022e+04    -0.002  0.99849
## Cryaquolls_Typic_cmplx  1.233e+13  2.747e+12     4.490 7.13e-06 ***
## Cryoborolis_cmplx             NA         NA        NA       NA
## Cryorthents           7.792e-04  1.109e+01     0.000  0.99994
## Cryorthents_cmplx      2.984e+00  5.490e+01     0.054  0.95665
## Cryumbrepts                   NA         NA        NA       NA
## Cryumbrepts_cmplx             NA         NA        NA       NA
## Gateview                      NA         NA        NA       NA
## Gothic                 1.254e+00  9.859e+06     0.000  1.00000
## Granile                2.767e+00  1.108e+01     0.250  0.80288
## Haploborolis          -2.289e+01  5.046e+03    -0.005  0.99638
## Legault               -4.505e+15  1.112e+12 -4052.572  < 2e-16 ***
## Legault_cmplx                 NA         NA        NA       NA
## Leighcan               1.309e+00  1.108e+01     0.118  0.90598
## Leighcan_cmplx         2.323e+00  1.109e+01     0.209  0.83408
## Leighcan_warm         -1.925e+01  7.811e+03    -0.002  0.99803
## Moran                         NA         NA        NA       NA
## Ratake                 4.708e-02  3.033e+02     0.000  0.99988
## Ratake_cmplx           7.682e-01  3.033e+02     0.003  0.99798
## Rogert                        NA         NA        NA       NA
## Supervisor_Limber_cmplx       NA         NA        NA       NA
## Troutville            -4.493e+15  2.821e+12 -1592.766  < 2e-16 ***
```

```
## Unspecified                        NA         NA        NA       NA
## Vanet                              NA         NA        NA       NA
## Wetmore                     5.335e-01  3.215e+02     0.002  0.99868
## Bouldery_ext                1.233e+13  2.747e+12     4.490 7.13e-06 ***
## Rock_Land                   1.246e+00  5.099e-02    24.438  < 2e-16 ***
## Rock_Land_cmplx             2.379e+00  3.561e-01     6.683 2.35e-11 ***
## Rock_Outcrop                       NA         NA        NA       NA
## Rock_Outcrop_cmplx          8.167e-01  3.431e-01     2.380  0.01729 *
## 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:  67859  on 406708  degrees of freedom
## Residual deviance: 567917  on 406653  degrees of freedom
## AIC: 568029
##
## Number of Fisher Scoring iterations: 25
```

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

```
## [1] "Aspen Individual Logistic Model Calculation completed at 2018-08-12 14:51:22"
```

```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 Aspen Logistic Model Probabilities - All Aggregated Vars

Aspen Probabilities - All Aggregated Data

Predict the probability of Aspen for aggregated Data - all variables.

```r
# Predict Aspen Agg Data - all variables

  Aspen_Agg_Train_predict= predict(Aspen_Agg_LogMod, type="response")
  Aspen_Agg_Train_Logit= predict(Aspen_Agg_LogMod)
  summary(Aspen_Agg_Train_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.001383 0.006737 0.074669 0.022181 1.000000
```

```r
  str(Aspen_Agg_Train_predict)
```

```
##  Named num [1:406709] 2.81e-02 2.69e-02 2.22e-16 2.53e-02 4.27e-02 ...
##  - attr(*, "names")= chr [1:406709] "1" "2" "3" "4" ...
```

```r
  #plot(table(Aspen_Agg_Train_predict))
  #plot(table(Aspen_Agg_Train_Logit))
  dens<-data.frame(table(Aspen_Agg_Train_predict))
```

```r
# str(dens)

  Aspen_Agg_Test_predict= predict(Aspen_Agg_LogMod, type="response",newdata=forestTest)
  summary(Aspen_Agg_Test_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.001394 0.006740 0.074482 0.022308 1.000000
```

```r
   str(Aspen_Agg_Test_predict)
```

```
##  Named num [1:174303] 0.0732 0.0367 0.0732 0.0826 0.0234 ...
##  - attr(*, "names")= chr [1:174303] "1" "2" "3" "4" ...
```

Aspen Probabilities - All Individuated Data

Predict the probability of Aspen for Individual Data - all variables.

```r
  Aspen_Ind_Train_predict= predict(Aspen_Ind_LogMod, type="response")
  summary(Aspen_Ind_Train_predict)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0007707 0.0055022 0.0758024 0.0208478 1.0000000
```

```r
  Aspen_Ind_Test_predict= predict(Aspen_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
```

```r
  summary(Aspen_Ind_Test_predict)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0007697 0.0054814 0.0748327 0.0207176 1.0000000
```

Aspen Receiver Operating Characteristic (ROC) - All Vars

Aspen Receiver ROC - All Aggregated Data

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

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

    ROCpred_Aspen_Agg = prediction(Aspen_Agg_Train_predict, forestTrain$Aspen)
    summary(ROCpred_Aspen_Agg)
    ROCperf_Aspen_Agg = performance(ROCpred_Aspen_Agg, "tpr", "fpr")
    summary(ROCperf_Aspen_Agg)

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

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

```
  Aspen_Agg_All_ROC_AUC = 84.2
}
```

```
## [1] "ROC graph 1 started at 2018-08-12 14:51:25"
## [1] "Aspen_Agg_All_ROC_AUC= 82.7"
```

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

Aspen 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_Aspen_Ind = prediction(Aspen_Ind_Train_predict, forestTrain$Aspen)
  summary(ROCpred_Aspen_Ind)
  ROCperf_Aspen_Ind = performance(ROCpred_Aspen_Ind, "tpr", "fpr")
  summary(ROCperf_Aspen_Ind)

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

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

```
## [1] "ROCR graph 2 started at 2018-08-12 14:52:16"
## [1] "Aspen_Ind_All_ROC_AUC= 83.1"
```

```
## 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))
```

```
## [1] "ROCR graph 2 completed at 2018-08-12 14:52:58"
```

Calculate Accuracy of Aspen Logisitic Models - All Vars

Calculate Aspen Aggregated Data Logisitic Model Accuracy - All Vars

Find best threshold for Aspen using all aggregated data.

```
result = calcLogisticModelAccuracy (forestTrain$Aspen, Aspen_Agg_Train_predict,
                      0.0, 1, 10, "Aspen", "Other", 1,1)
```

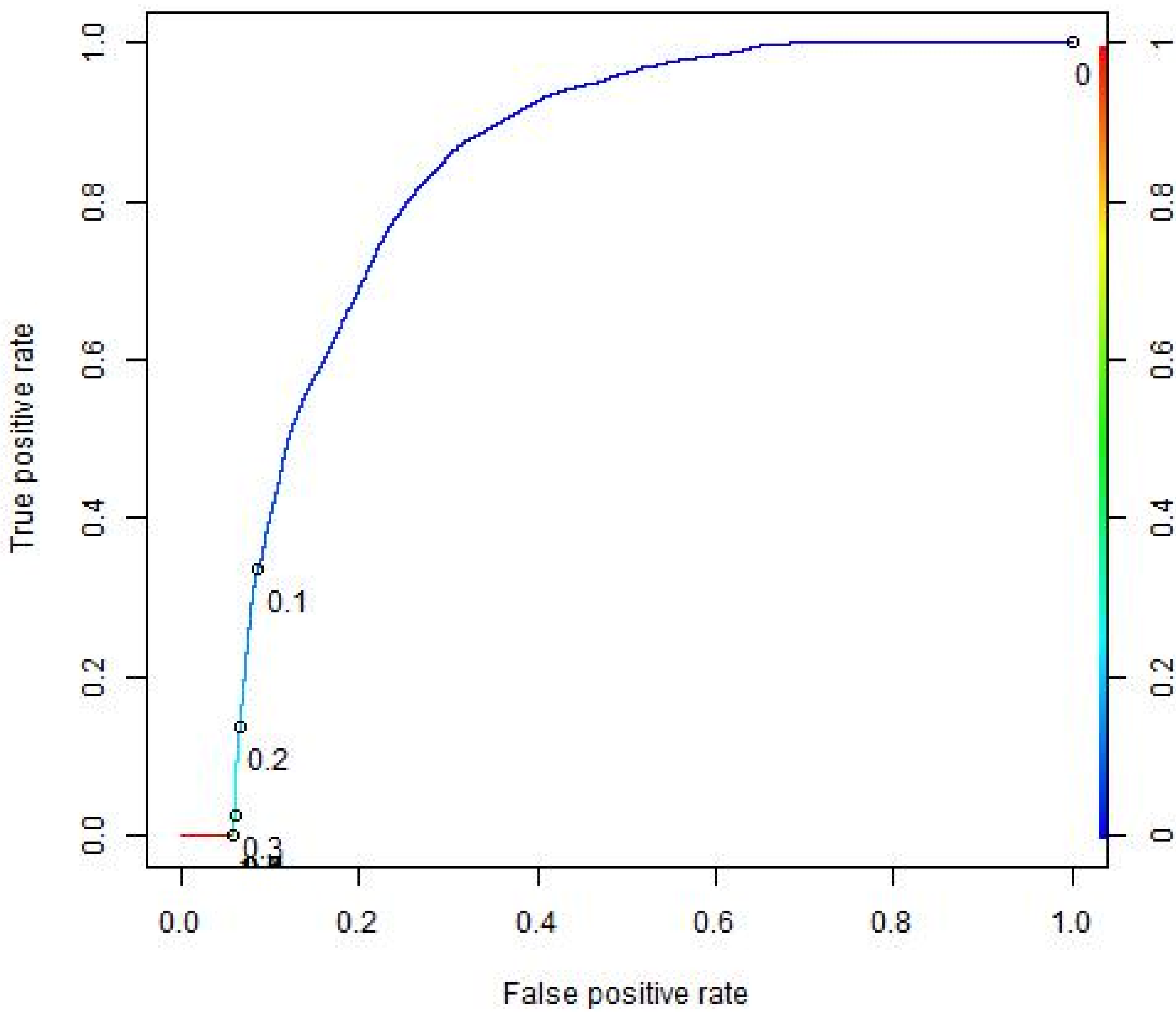

Figure 1: Aspen ROC for All Aggregated Data

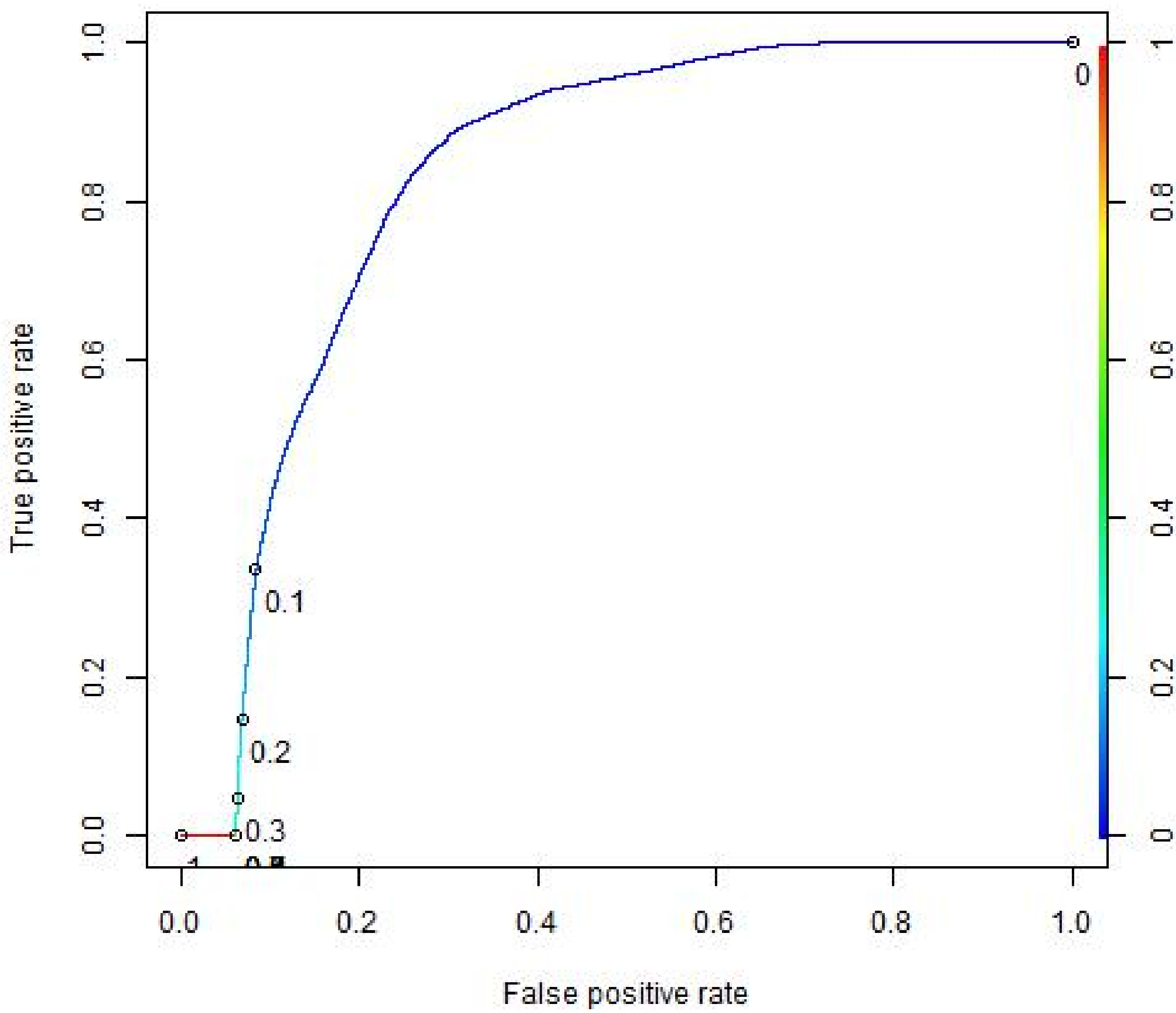

Figure 2: Aspen ROC for All Individuated Data

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
## [1] "Thresh=0, Accuracy=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=90.4%, BaseAcc(Other)=98.3%, Sens=33.7%, Spec=91.3%, Sens^2+Spec^2=0.948"
## [1] "Thresh=0.2, Accuracy=92.1%, BaseAcc(Other)=98.3%, Sens=13.6%, Spec=93.4%, Sens^2+Spec^2=0.892"
## [1] "Thresh=0.3, Accuracy=92.5%, BaseAcc(Other)=98.3%, Sens=2.6%, Spec=94%, Sens^2+Spec^2=0.885"
## [1] "Thresh=0.4, Accuracy=92.5%, BaseAcc(Other)=98.3%, Sens=0.1%, Spec=94.1%, Sens^2+Spec^2=0.886"
## [1] "Thresh=0.5, Accuracy=92.6%, BaseAcc(Other)=98.3%, Sens=0%, Spec=94.1%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.6, Accuracy=92.6%, BaseAcc(Other)=98.3%, Sens=0%, Spec=94.1%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.7, Accuracy=92.6%, BaseAcc(Other)=98.3%, Sens=0%, Spec=94.1%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.8, Accuracy=92.6%, BaseAcc(Other)=98.3%, Sens=0%, Spec=94.1%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.9, Accuracy=92.6%, BaseAcc(Other)=98.3%, Sens=0%, Spec=94.1%, Sens^2+Spec^2=-2"
## [1] "Thresh=1, Accuracy=98.3%, BaseAcc(Other)=98.3%, 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=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.01, Accuracy=60.2%, BaseAcc(Other)=98.3%, Sens=92.7%, Spec=59.7%, Sens^2+Spec^2=1.217"
## [1] "Thresh=0.02, Accuracy=74.2%, BaseAcc(Other)=98.3%, Sens=80.5%, Spec=74.1%, Sens^2+Spec^2=1.198"
## [1] "Thresh=0.03, Accuracy=79.9%, BaseAcc(Other)=98.3%, Sens=68.6%, Spec=80.1%, Sens^2+Spec^2=1.114"
## [1] "Thresh=0.04, Accuracy=83%, BaseAcc(Other)=98.3%, Sens=61.2%, Spec=83.4%, Sens^2+Spec^2=1.071"
## [1] "Thresh=0.05, Accuracy=85.3%, BaseAcc(Other)=98.3%, Sens=56.1%, Spec=85.8%, Sens^2+Spec^2=1.052"
## [1] "Thresh=0.06, Accuracy=87.2%, BaseAcc(Other)=98.3%, Sens=50.6%, Spec=87.8%, Sens^2+Spec^2=1.027"
## [1] "Thresh=0.07, Accuracy=88.4%, BaseAcc(Other)=98.3%, Sens=43.9%, Spec=89.2%, Sens^2+Spec^2=0.989"
## [1] "Thresh=0.08, Accuracy=89.3%, BaseAcc(Other)=98.3%, Sens=39.6%, Spec=90.1%, Sens^2+Spec^2=0.97"
## [1] "Thresh=0.09, Accuracy=89.9%, BaseAcc(Other)=98.3%, Sens=35.8%, Spec=90.8%, Sens^2+Spec^2=0.954"
## [1] "Thresh=0.1, Accuracy=90.4%, BaseAcc(Other)=98.3%, Sens=33.7%, Spec=91.3%, Sens^2+Spec^2=0.948"
## [1] "Thresh=0.11, Accuracy=90.7%, BaseAcc(Other)=98.3%, Sens=31.9%, Spec=91.7%, Sens^2+Spec^2=0.943"
## [1] "Thresh=0.12, Accuracy=91%, BaseAcc(Other)=98.3%, Sens=29.7%, Spec=92%, Sens^2+Spec^2=0.936"
## [1] "Thresh=0.13, Accuracy=91.2%, BaseAcc(Other)=98.3%, Sens=26.8%, Spec=92.3%, Sens^2+Spec^2=0.925"
## [1] "Thresh=0.14, Accuracy=91.4%, BaseAcc(Other)=98.3%, Sens=24.3%, Spec=92.5%, Sens^2+Spec^2=0.916"
## [1] "Thresh=0.15, Accuracy=91.6%, BaseAcc(Other)=98.3%, Sens=22%, Spec=92.7%, Sens^2+Spec^2=0.908"
## [1] "Thresh=0.16, Accuracy=91.7%, BaseAcc(Other)=98.3%, Sens=19.7%, Spec=92.9%, Sens^2+Spec^2=0.902"
## [1] "Thresh=0.17, Accuracy=91.8%, BaseAcc(Other)=98.3%, Sens=17.9%, Spec=93%, Sens^2+Spec^2=0.898"
## [1] "Thresh=0.18, Accuracy=91.9%, BaseAcc(Other)=98.3%, Sens=16.4%, Spec=93.2%, Sens^2+Spec^2=0.896"
## [1] "Thresh=0.19, Accuracy=92%, BaseAcc(Other)=98.3%, Sens=15%, Spec=93.3%, Sens^2+Spec^2=0.894"
## [1] "Best Sensitivity_Specificity threshold= 0.01 inc= 0.01"
## [1] "========================================"
## [1] "start= 0 end= 0.02 inc= 0.001"
## [1] "Thresh=0, Accuracy=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.001, Accuracy=25.5%, BaseAcc(Other)=98.3%, Sens=100%, Spec=24.3%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.002, Accuracy=29.1%, BaseAcc(Other)=98.3%, Sens=100%, Spec=28%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.003, Accuracy=34.1%, BaseAcc(Other)=98.3%, Sens=99.9%, Spec=33%, Sens^2+Spec^2=1.108"
## [1] "Thresh=0.004, Accuracy=39.9%, BaseAcc(Other)=98.3%, Sens=98.6%, Spec=38.9%, Sens^2+Spec^2=1.125
## [1] "Thresh=0.005, Accuracy=45%, BaseAcc(Other)=98.3%, Sens=97.7%, Spec=44.1%, Sens^2+Spec^2=1.15"
## [1] "Thresh=0.006, Accuracy=48.9%, BaseAcc(Other)=98.3%, Sens=96.8%, Spec=48.1%, Sens^2+Spec^2=1.17"
## [1] "Thresh=0.007, Accuracy=52.3%, BaseAcc(Other)=98.3%, Sens=95.7%, Spec=51.6%, Sens^2+Spec^2=1.183
## [1] "Thresh=0.008, Accuracy=55.2%, BaseAcc(Other)=98.3%, Sens=94.7%, Spec=54.5%, Sens^2+Spec^2=1.195
## [1] "Thresh=0.009, Accuracy=57.8%, BaseAcc(Other)=98.3%, Sens=93.9%, Spec=57.2%, Sens^2+Spec^2=1.21"
## [1] "Thresh=0.01, Accuracy=60.2%, BaseAcc(Other)=98.3%, Sens=92.7%, Spec=59.7%, Sens^2+Spec^2=1.217"
## [1] "Thresh=0.011, Accuracy=62.3%, BaseAcc(Other)=98.3%, Sens=91.5%, Spec=61.8%, Sens^2+Spec^2=1.22"
## [1] "Thresh=0.012, Accuracy=64.2%, BaseAcc(Other)=98.3%, Sens=90.2%, Spec=63.8%, Sens^2+Spec^2=1.222
## [1] "Thresh=0.013, Accuracy=66%, BaseAcc(Other)=98.3%, Sens=89.1%, Spec=65.6%, Sens^2+Spec^2=1.225"
## [1] "Thresh=0.014, Accuracy=67.6%, BaseAcc(Other)=98.3%, Sens=88%, Spec=67.2%, Sens^2+Spec^2=1.227"
```

```
## [1] "Thresh=0.015, Accuracy=69%, BaseAcc(Other)=98.3%, Sens=87%, Spec=68.7%, Sens^2+Spec^2=1.23"
## [1] "Thresh=0.016, Accuracy=70.2%, BaseAcc(Other)=98.3%, Sens=85.7%, Spec=70%, Sens^2+Spec^2=1.226"
## [1] "Thresh=0.017, Accuracy=71.3%, BaseAcc(Other)=98.3%, Sens=84.3%, Spec=71.1%, Sens^2+Spec^2=1.217
## [1] "Thresh=0.018, Accuracy=72.4%, BaseAcc(Other)=98.3%, Sens=83%, Spec=72.2%, Sens^2+Spec^2=1.211"
## [1] "Thresh=0.019, Accuracy=73.3%, BaseAcc(Other)=98.3%, Sens=81.7%, Spec=73.2%, Sens^2+Spec^2=1.204
## [1] "========================================"
## [1] "Best Threshold=0.015"
## [1] "Best Sensitivity_Specificity=1.2303663210172"
```

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

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

```
result = calcLogisticModelAccuracy (forestTrain$Aspen, Aspen_Agg_Train_predict,
                       curThresh, curThresh, 1, "Aspen", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.015"
## [1] "predicted performance="
##                 Predicted
## Actual           FALSE=Predict:Other TRUE=Predict:Aspen
##   0=Actual:Other     274910 (TN)         125154 (FP)
##   1=Actual:Aspen     859 (FN)            5786 (TP)
## [1] "Sensitivity= 0.870729872084274 (True positive rate of Aspen = TP/(TP+FN) = 5786 /( 5786 + 859 )
## [1] "Specificity= 0.687165053591425 (True negative rate of Other = TN/(TN+FP) = 274910 /( 274910 + 1
## [1] "Sens^2+Spec^2=1.23"
## [1] "Baseline (Other) Accuracy=0.983661"
## [1] "Logistic Accuracy=0.690164"
```

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

```
result = calcLogisticModelAccuracy (forestTest$Aspen, Aspen_Agg_Test_predict,
                       curThresh, curThresh, 1, "Aspen", "Other", 3,
                       saveFile=saveFileName, desc="Aspen All Aggregate Vars",
                       AIC=Aspen_Agg_All_aic, AUC=Aspen_Agg_All_ROC_AUC, Append=FALSE)
```

```
## [1] "Model Performance for threshold= 0.015"
## [1] "predicted performance="
##                 Predicted
## Actual           FALSE=Predict:Other TRUE=Predict:Aspen
##   0=Actual:Other     117669 (TN)         53786 (FP)
##   1=Actual:Aspen     389 (FN)            2459 (TP)
## [1] "Sensitivity= 0.863412921348315 (True positive rate of Aspen = TP/(TP+FN) = 2459 /( 2459 + 389 )
## [1] "Specificity= 0.686296695926045 (True negative rate of Other = TN/(TN+FP) = 117669 /( 117669 + 5
## [1] "Sens^2+Spec^2=1.216"
## [1] "Baseline (Other) Accuracy=0.98366"
## [1] "Logistic Accuracy=0.68919"
```

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

```r
Aspen_Agg_All_baseline_acc = as.integer(as.numeric(Aspen_Agg_All_baseline_acc)*1000)/10
Aspen_Agg_All_sens = as.integer(as.numeric(Aspen_Agg_All_sens)*1000)/10
Aspen_Agg_All_spec = as.integer(as.numeric(Aspen_Agg_All_spec)*1000)/10
```

Calculate Aspen Individuated Data Logisitic Model Accuracy - All Vars

Find best threshold for Aspen using all individuated data.

```r
result = calcLogisticModelAccuracy (forestTrain$Aspen, Aspen_Ind_Train_predict,
                         0.0, 1, 10, "Aspen", "Other", 1,1)
```

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
## [1] "Thresh=0, Accuracy=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=90.6%, BaseAcc(Other)=98.3%, Sens=33.5%, Spec=91.5%, Sens^2+Spec^2=0.951"
## [1] "Thresh=0.2, Accuracy=91.9%, BaseAcc(Other)=98.3%, Sens=14.8%, Spec=93.2%, Sens^2+Spec^2=0.89"
## [1] "Thresh=0.3, Accuracy=92.2%, BaseAcc(Other)=98.3%, Sens=4.8%, Spec=93.7%, Sens^2+Spec^2=0.88"
## [1] "Thresh=0.4, Accuracy=92.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=93.8%, Sens^2+Spec^2=0.88"
## [1] "Thresh=0.5, Accuracy=92.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=93.8%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.6, Accuracy=92.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=93.8%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.7, Accuracy=92.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=93.8%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.8, Accuracy=92.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=93.8%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.9, Accuracy=92.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=93.8%, Sens^2+Spec^2=-2"
## [1] "Thresh=1, Accuracy=98.3%, BaseAcc(Other)=98.3%, 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=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.01, Accuracy=62.9%, BaseAcc(Other)=98.3%, Sens=92.3%, Spec=62.4%, Sens^2+Spec^2=1.242"
## [1] "Thresh=0.02, Accuracy=75.3%, BaseAcc(Other)=98.3%, Sens=81.2%, Spec=75.2%, Sens^2+Spec^2=1.226"
## [1] "Thresh=0.03, Accuracy=80.9%, BaseAcc(Other)=98.3%, Sens=67.5%, Spec=81.1%, Sens^2+Spec^2=1.115"
## [1] "Thresh=0.04, Accuracy=84.2%, BaseAcc(Other)=98.3%, Sens=58.1%, Spec=84.6%, Sens^2+Spec^2=1.054"
## [1] "Thresh=0.05, Accuracy=86.4%, BaseAcc(Other)=98.3%, Sens=52.4%, Spec=87%, Sens^2+Spec^2=1.032"
## [1] "Thresh=0.06, Accuracy=88%, BaseAcc(Other)=98.3%, Sens=47.2%, Spec=88.7%, Sens^2+Spec^2=1.01"
## [1] "Thresh=0.07, Accuracy=89.1%, BaseAcc(Other)=98.3%, Sens=42.7%, Spec=89.8%, Sens^2+Spec^2=0.99"
## [1] "Thresh=0.08, Accuracy=89.7%, BaseAcc(Other)=98.3%, Sens=38.6%, Spec=90.6%, Sens^2+Spec^2=0.97"
## [1] "Thresh=0.09, Accuracy=90.2%, BaseAcc(Other)=98.3%, Sens=36.2%, Spec=91.1%, Sens^2+Spec^2=0.962"
## [1] "Thresh=0.1, Accuracy=90.6%, BaseAcc(Other)=98.3%, Sens=33.5%, Spec=91.5%, Sens^2+Spec^2=0.951"
## [1] "Thresh=0.11, Accuracy=90.8%, BaseAcc(Other)=98.3%, Sens=30.7%, Spec=91.8%, Sens^2+Spec^2=0.938"
## [1] "Thresh=0.12, Accuracy=91%, BaseAcc(Other)=98.3%, Sens=28.4%, Spec=92.1%, Sens^2+Spec^2=0.929"
## [1] "Thresh=0.13, Accuracy=91.2%, BaseAcc(Other)=98.3%, Sens=26.1%, Spec=92.2%, Sens^2+Spec^2=0.919"
## [1] "Thresh=0.14, Accuracy=91.3%, BaseAcc(Other)=98.3%, Sens=24.3%, Spec=92.4%, Sens^2+Spec^2=0.914"
## [1] "Thresh=0.15, Accuracy=91.4%, BaseAcc(Other)=98.3%, Sens=22.2%, Spec=92.6%, Sens^2+Spec^2=0.907"
## [1] "Thresh=0.16, Accuracy=91.5%, BaseAcc(Other)=98.3%, Sens=20.4%, Spec=92.7%, Sens^2+Spec^2=0.902"
## [1] "Thresh=0.17, Accuracy=91.6%, BaseAcc(Other)=98.3%, Sens=18.9%, Spec=92.8%, Sens^2+Spec^2=0.898"
## [1] "Thresh=0.18, Accuracy=91.7%, BaseAcc(Other)=98.3%, Sens=17.3%, Spec=93%, Sens^2+Spec^2=0.895"
## [1] "Thresh=0.19, Accuracy=91.8%, BaseAcc(Other)=98.3%, Sens=16.1%, Spec=93.1%, Sens^2+Spec^2=0.893"
## [1] "Best Sensitivity_Specificity threshold= 0.01 inc= 0.01"
## [1] "========================================"
## [1] "start= 0 end= 0.02 inc= 0.001"
## [1] "Thresh=0, Accuracy=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.001, Accuracy=28.1%, BaseAcc(Other)=98.3%, Sens=100%, Spec=26.9%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.002, Accuracy=34.6%, BaseAcc(Other)=98.3%, Sens=99.6%, Spec=33.5%, Sens^2+Spec^2=1.105
## [1] "Thresh=0.003, Accuracy=40.4%, BaseAcc(Other)=98.3%, Sens=98.4%, Spec=39.5%, Sens^2+Spec^2=1.125
```

```
## [1] "Thresh=0.004, Accuracy=45.3%, BaseAcc(Other)=98.3%, Sens=97.2%, Spec=44.5%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.005, Accuracy=49.6%, BaseAcc(Other)=98.3%, Sens=96.1%, Spec=48.8%, Sens^2+Spec^2=1.164
## [1] "Thresh=0.006, Accuracy=53.1%, BaseAcc(Other)=98.3%, Sens=95.4%, Spec=52.4%, Sens^2+Spec^2=1.186
## [1] "Thresh=0.007, Accuracy=56.1%, BaseAcc(Other)=98.3%, Sens=94.6%, Spec=55.5%, Sens^2+Spec^2=1.205
## [1] "Thresh=0.008, Accuracy=58.7%, BaseAcc(Other)=98.3%, Sens=94.1%, Spec=58.1%, Sens^2+Spec^2=1.224
## [1] "Thresh=0.009, Accuracy=60.9%, BaseAcc(Other)=98.3%, Sens=93.3%, Spec=60.4%, Sens^2+Spec^2=1.235
## [1] "Thresh=0.01, Accuracy=62.9%, BaseAcc(Other)=98.3%, Sens=92.3%, Spec=62.4%, Sens^2+Spec^2=1.242"
## [1] "Thresh=0.011, Accuracy=64.7%, BaseAcc(Other)=98.3%, Sens=91.3%, Spec=64.3%, Sens^2+Spec^2=1.249
## [1] "Thresh=0.012, Accuracy=66.4%, BaseAcc(Other)=98.3%, Sens=90.6%, Spec=66%, Sens^2+Spec^2=1.258"
## [1] "Thresh=0.013, Accuracy=67.9%, BaseAcc(Other)=98.3%, Sens=89.9%, Spec=67.5%, Sens^2+Spec^2=1.266
## [1] "Thresh=0.014, Accuracy=69.3%, BaseAcc(Other)=98.3%, Sens=89%, Spec=69%, Sens^2+Spec^2=1.269"
## [1] "Thresh=0.015, Accuracy=70.6%, BaseAcc(Other)=98.3%, Sens=87.7%, Spec=70.3%, Sens^2+Spec^2=1.264
## [1] "Thresh=0.016, Accuracy=71.8%, BaseAcc(Other)=98.3%, Sens=86.5%, Spec=71.5%, Sens^2+Spec^2=1.26"
## [1] "Thresh=0.017, Accuracy=72.8%, BaseAcc(Other)=98.3%, Sens=85.1%, Spec=72.6%, Sens^2+Spec^2=1.253
## [1] "Thresh=0.018, Accuracy=73.7%, BaseAcc(Other)=98.3%, Sens=84%, Spec=73.5%, Sens^2+Spec^2=1.247"
## [1] "Thresh=0.019, Accuracy=74.5%, BaseAcc(Other)=98.3%, Sens=82.7%, Spec=74.4%, Sens^2+Spec^2=1.238
## [1] "========================================"
## [1] "Best Threshold=0.014"
## [1] "Best Sensitivity_Specificity=1.26962705249119"
```

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

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

```r
result = calcLogisticModelAccuracy (forestTrain$Aspen, Aspen_Ind_Train_predict,
                        curThresh, curThresh, 1, "Aspen", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.014"
## [1] "predicted performance="
##                  Predicted
## Actual            FALSE=Predict:Other TRUE=Predict:Aspen
##   0=Actual:Other     276151 (TN)         123913 (FP)
##   1=Actual:Aspen     727 (FN)            5918 (TP)
## [1] "Sensitivity= 0.890594431903687 (True positive rate of Aspen = TP/(TP+FN) = 5918 /( 5918 + 727 )
## [1] "Specificity= 0.690267057270837 (True negative rate of Other = TN/(TN+FP) = 276151 /( 276151 + 1
## [1] "Sens^2+Spec^2=1.269"
## [1] "Baseline (Other) Accuracy=0.983661"
## [1] "Logistic Accuracy=0.69354"
```

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

```r
result = calcLogisticModelAccuracy (forestTest$Aspen, Aspen_Ind_Test_predict,
                        curThresh, curThresh, 1, "Aspen", "Other", 3,
                        saveFile=saveFileName, desc="Aspen All Individualized Vars",
                        AIC=Aspen_Ind_All_aic, AUC=Aspen_Ind_All_ROC_AUC)
```

```
## [1] "Model Performance for threshold= 0.014"
## [1] "predicted performance="
##                  Predicted
## Actual            FALSE=Predict:Other TRUE=Predict:Aspen
##   0=Actual:Other     118504 (TN)         52951 (FP)
##   1=Actual:Aspen     328 (FN)            2520 (TP)
## [1] "Sensitivity= 0.884831460674157 (True positive rate of Aspen = TP/(TP+FN) = 2520 /( 2520 + 328 )
## [1] "Specificity= 0.691166778454988 (True negative rate of Other = TN/(TN+FP) = 118504 /( 118504 + 5
## [1] "Sens^2+Spec^2=1.26"
## [1] "Baseline (Other) Accuracy=0.98366"
```

```
## [1] "Logistic Accuracy=0.694331"
```

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

The Aspen 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.

Aspen Logistic Regression - Significant Variables

Create Aspen 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.

Aspen Logistic Model using Significant Aggregated Data

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

```
  Aspen_Agg_LogMod =
    glm(Aspen ~
          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 +  # Amount of shade at 9am
          Shade12PM + # Amount of shade at 12pm
          # Shade3PM +  # Amount of shade at 3pm - removed 1st pass
          # Wilderness areas:
            # RWwild + NEwild + CMwild + CPwild +
          # Aggregated Soil type:
            # ST01 + ST02 + ST03 + ST04 + ST05 + ST06 +  - removed 1st pass
            ST07 + ST08 +
            # ST09 + ST10 + ST11 +   - removed 1st pass
            ST12 +
            # ST13 + - removed 1st pass
            ST14 + ST15 +
            # ST16 + ST17 + ST18 + ST19 + ST20 + - removed 1st pass
            ST21
            # ST22 + ST23 + ST24 + ST25 + ST26 +  - removed 1st pass
            # ST27 + ST28 + ST29 + ST30 +         - removed 1st pass
            # ST31 + ST32 + ST33 + ST34 + ST35 +  - removed 1st pass
            # ST36 + ST37 + ST38 + ST39 + ST40 ,  - removed 1st pass
```

```r
         ,
         data=forestTrain, family=binomial)

  Aspen_Agg_Sig_LogMod = Aspen_Agg_LogMod
  save("Aspen_Agg_Sig_LogMod", file="Aspen_Agg_Sig_LogMod.Rdata")

  Aspen_Agg_Sig_aic<-as.integer(Aspen_Agg_LogMod$aic)
  Aspen_Agg_Sig_aic
```

```
## [1] 60588
```

Check the coefficients of the Aspen model using significant aggregated data.

```r
summary(Aspen_Agg_LogMod)
```

```
##
## Call:
## glm(formula = Aspen ~ Elev + Aspect + Slope + H2OHD + H2OVD +
##     RoadHD + FirePtHD + Shade9AM + Shade12PM + ST07 + ST08 +
##     ST12 + ST14 + ST15 + ST21, family = binomial, data = forestTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8014  -0.2009  -0.1391  -0.0880   3.7105
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.168e+00  2.449e-01 -29.270  < 2e-16 ***
## Elev        -1.007e-03  4.597e-05 -21.895  < 2e-16 ***
## Aspect       1.488e-03  1.833e-04   8.117 4.78e-16 ***
## Slope        1.806e-02  2.160e-03   8.363  < 2e-16 ***
## H2OHD       -1.921e-03  1.074e-04 -17.888  < 2e-16 ***
## H2OVD        5.268e-03  3.142e-04  16.768  < 2e-16 ***
## RoadHD      -4.514e-04  1.323e-05 -34.114  < 2e-16 ***
## FirePtHD     8.510e-06  1.242e-05   0.685    0.493
## Shade9AM     2.458e-02  6.268e-04  39.217  < 2e-16 ***
## Shade12PM    5.378e-03  8.108e-04   6.633 3.29e-11 ***
## ST07        -1.368e+01  1.264e+03  -0.011    0.991
## ST08        -1.391e+01  9.351e+02  -0.015    0.988
## ST12        -1.531e+01  7.166e+01  -0.214    0.831
## ST14        -1.667e+01  5.234e+02  -0.032    0.975
## ST15        -1.670e+01  6.187e+03  -0.003    0.998
## ST21        -1.528e+01  4.331e+02  -0.035    0.972
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 67859  on 406708  degrees of freedom
## Residual deviance: 60556  on 406693  degrees of freedom
## AIC: 60588
##
## Number of Fisher Scoring iterations: 18
```

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

Aspen Logistic Model using Significant Individuated Data

Create a logistic model for the significant individuated variables.

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

```
Aspen_Ind_LogMod =
  glm(Aspen ~
        Elev +     # Elevation in meters of cell
        Aspect +   # Direction in degrees slope faces
        # Slope +    # Slope / steepness of hill in degrees (0 to 90) # rem 3rd pass
        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 +  # Amount of shade at 9am
        Shade12PM + # Amount of shade at 12pm
        Shade3PM +  # Amount of shade at 3pm - removed 1st pass
        # Wilderness areas:
           RWwild + NEwild + CMwild + CPwild +
        # Climate Zone:
          Montane_low +
          Montane +
          # SubAlpine + Alpine +  - removed 1st pass
          # Dry +  - removed 1st pass
          Non_Dry +
        # Geology Zone:
          # Alluvium + - removed 1st pass
          # Glacial + # rem 3rd pass
          Sed_mix +
          # Ign_Meta +  - removed 1st pass
        # Soil Family:
          # Aquolis_cmplx +  - removed 1st pass
          # Argiborolis_Pachic +  - removed 1st pass
          # Borohemists_cmplx + Bross +  - removed 1st pass
          # Bullwark + Bullwark_Cmplx + Catamount +  - removed 1st pass
          Catamount_cmplx +
          # Cathedral + Como + Cryaquepts_cmplx +  - removed 1st pass
          # Cryaquepts_Typic +  # rem 3rd pass
          # Cryaquolls +  - removed 1st pass
          # Cryaquolls_cmplx + Cryaquolls_Typic +  - removed 1st pass
          # Cryaquolls_Typic_cmplx +  # rem 3rd pass
          # Cryoborolis_cmplx +  - removed 1st pass
          # Cryorthents + Cryorthents_cmplx +   - removed 1st pass
          # Cryumbrepts + Cryumbrepts_cmplx + Gateview +  - removed 1st pass
          # Gothic + Granile + Haploborolis +  - removed 1st pass
          Legault +
          # Legault_cmplx +  - removed 1st pass
          # Leighcan + Leighcan_cmplx + Leighcan_warm +  - removed 1st pass
          # Moran + Ratake + Ratake_cmplx + Rogert +  - removed 1st pass
          # Supervisor_Limber_cmplx +  - removed 1st pass
          # Troutville +  # rem 3rd pass
          # Unspecified + Vanet + Wetmore +  - removed 1st pass
        # Soil Rock composition:
         # Bouldery_ext +  # removed 2nd pass
```

```r
            Rock_Land +
            Rock_Land_cmplx +
            # Rock_Outcrop +  - removed 1st pass
            Rock_Outcrop_cmplx
            # Rubbly + Stony + Stony_extreme +  - removed 1st pass
            # Stony_very + Till_Substratum , - removed 1st pass
            ,
          data=forestTrain, family=binomial)
```

```
## Warning: glm.fit: algorithm did not converge

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

```r
Aspen_Ind_Sig_LogMod = Aspen_Ind_LogMod
save("Aspen_Ind_Sig_LogMod", file="Aspen_Ind_Sig_LogMod.Rdata")

Aspen_Ind_Sig_aic<-as.integer(Aspen_Ind_LogMod$aic)
Aspen_Ind_Sig_aic
```

```
## [1] 53086
```

```r
summary(Aspen_Ind_LogMod)
```

```
##
## Call:
## glm(formula = Aspen ~ Elev + Aspect + H2OHD + H2OVD + RoadHD +
##     FirePtHD + Shade9AM + Shade12PM + Shade3PM + RWwild + NEwild +
##     CMwild + CPwild + Montane_low + Montane + Non_Dry + Sed_mix +
##     Catamount_cmplx + Legault + Rock_Land + Rock_Land_cmplx +
##     Rock_Outcrop_cmplx, family = binomial, data = forestTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1363  -0.1733  -0.0948  -0.0441   3.7975
##
## Coefficients: (2 not defined because of singularities)
##                      Estimate Std. Error    z value Pr(>|z|)
## (Intercept)        -1.036e+01  8.037e+00 -1.289e+00 0.197297
## Elev               -3.499e-03  9.012e-05 -3.883e+01  < 2e-16 ***
## Aspect              2.338e-03  1.905e-04  1.227e+01  < 2e-16 ***
## H2OHD              -1.100e-03  1.104e-04 -9.970e+00  < 2e-16 ***
## H2OVD               5.548e-03  3.172e-04  1.749e+01  < 2e-16 ***
## RoadHD             -4.794e-04  1.522e-05 -3.150e+01  < 2e-16 ***
## FirePtHD           -4.613e-05  1.369e-05 -3.369e+00 0.000753 ***
## Shade9AM            1.449e-02  2.598e-03  5.580e+00 2.41e-08 ***
## Shade12PM           4.966e-03  2.588e-03  1.919e+00 0.054979 .
## Shade3PM           -4.990e-03  2.078e-03 -2.402e+00 0.016312 *
## RWwild              1.328e+01  8.026e+00  1.654e+00 0.098040 .
## NEwild             -4.504e+15  4.640e+05 -9.706e+09  < 2e-16 ***
## CMwild              1.354e+01  8.026e+00  1.687e+00 0.091556 .
## CPwild                     NA         NA         NA       NA
## Montane_low        -1.655e+00  1.685e+00 -9.820e-01 0.326211
## Montane            -1.266e+00  1.684e+00 -7.520e-01 0.452223
## Non_Dry             1.324e+00  1.685e+00  7.860e-01 0.431902
## Sed_mix                    NA         NA         NA       NA
## Catamount_cmplx     2.480e-01  6.793e-02  3.651e+00 0.000261 ***
```

```
## Legault            -2.320e+01  2.969e+03 -8.000e-03 0.993765
## Rock_Land           1.229e+00  3.782e-02  3.249e+01  < 2e-16 ***
## Rock_Land_cmplx     2.196e-01  5.993e-02  3.665e+00 0.000247 ***
## Rock_Outcrop_cmplx -5.508e-01  4.660e-02 -1.182e+01  < 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: 67859  on 406708  degrees of freedom
## Residual deviance: 53044  on 406688  degrees of freedom
## AIC: 53086
##
## Number of Fisher Scoring iterations: 25
```

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

Predict Aspen Logistic Model Probabilities - Sig Vars

Aspen Probabilities using Significant Aggregated Data

Predict the probability of Aspen for aggregated Data - significant variables.

```r
# Predict Aspen Agg Data - significant variables

  Aspen_Agg_Train_predict= predict(Aspen_Agg_LogMod, type="response")
  summary(Aspen_Agg_Train_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.004165 0.010051 0.016338 0.020716 0.274654
```

```r
  Aspen_Agg_Test_predict= predict(Aspen_Agg_LogMod, type="response",newdata=forestTest)
  summary(Aspen_Agg_Test_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.004146 0.010021 0.016303 0.020650 0.282219
```

Aspen Probabilities using Significant Individuated Data

Predict the probability of Aspen using significant Individuated Data.

```r
  Aspen_Ind_Train_predict= predict(Aspen_Ind_LogMod, type="response")
  summary(Aspen_Ind_Train_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.001116 0.004828 0.016364 0.016012 0.475643
```

```r
  Aspen_Ind_Test_predict= predict(Aspen_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
```

```r
  summary(Aspen_Ind_Test_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.001121 0.004810 0.016419 0.016078 0.443016
```

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

```
## [1] "ROCR graph 2 completed at 2018-08-12 14:52:58"
```

Aspen Receiver Operating Characteristic (ROC) - Sig Vars

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

```
if (calcROC) {
  ROCpred_Aspen_Agg = prediction(Aspen_Agg_Train_predict, forestTrain$Aspen)
  summary(ROCpred_Aspen_Agg)

  ROCperf_Aspen_Agg = performance(ROCpred_Aspen_Agg, "tpr", "fpr")
  summary(ROCperf_Aspen_Agg)

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

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

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

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

  ROCpred_Aspen_Ind = prediction(Aspen_Ind_Train_predict, forestTrain$Aspen)
  summary(ROCpred_Aspen_Ind)

  ROCperf_Aspen_Ind = performance(ROCpred_Aspen_Ind, "tpr", "fpr")
  summary(ROCperf_Aspen_Ind)

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

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

```
## [1] "ROCR graph 2 started at 2018-08-12 14:58:21"
```

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

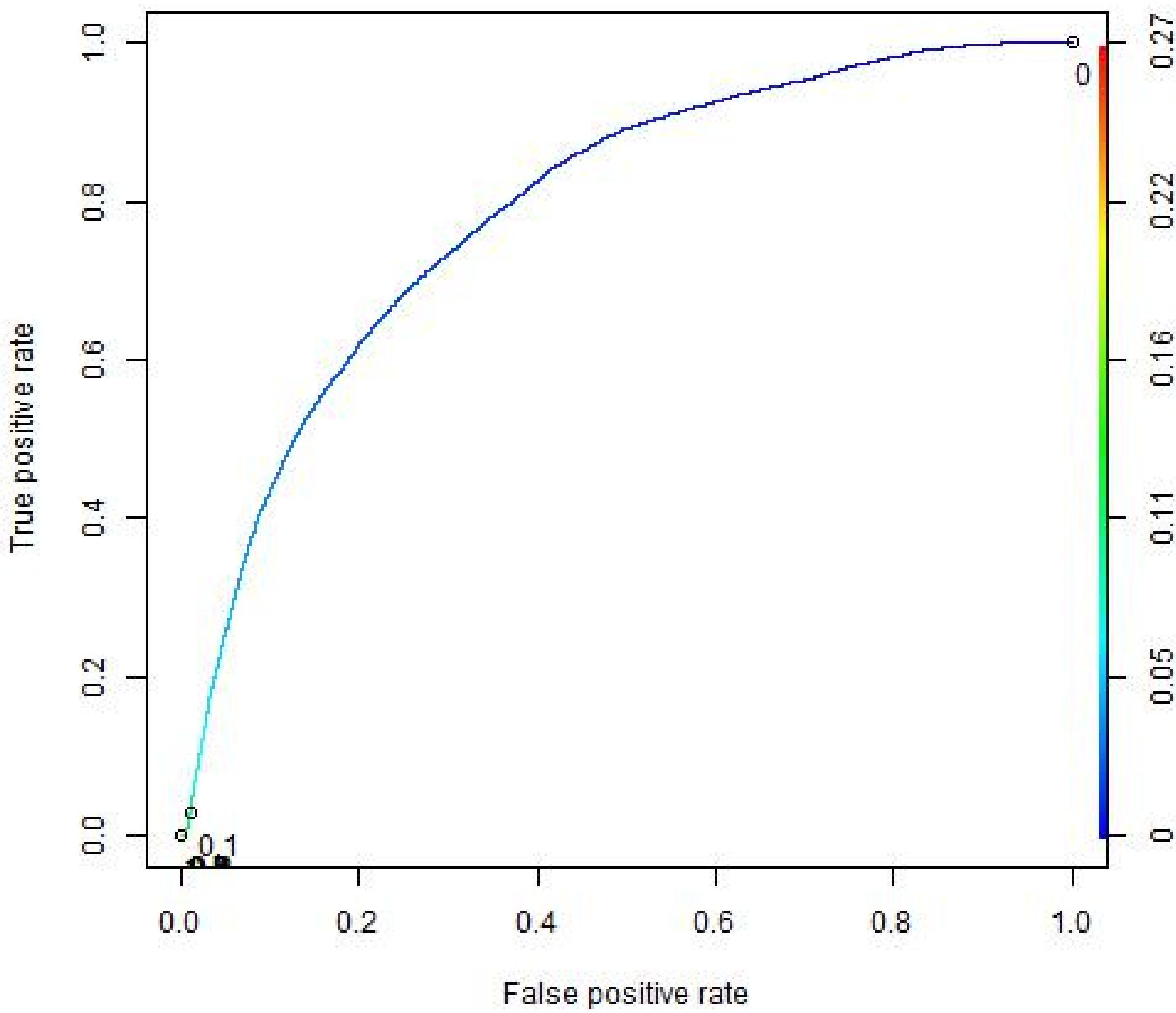

Figure 3: Aspen ROC for Aggregated Significant Data

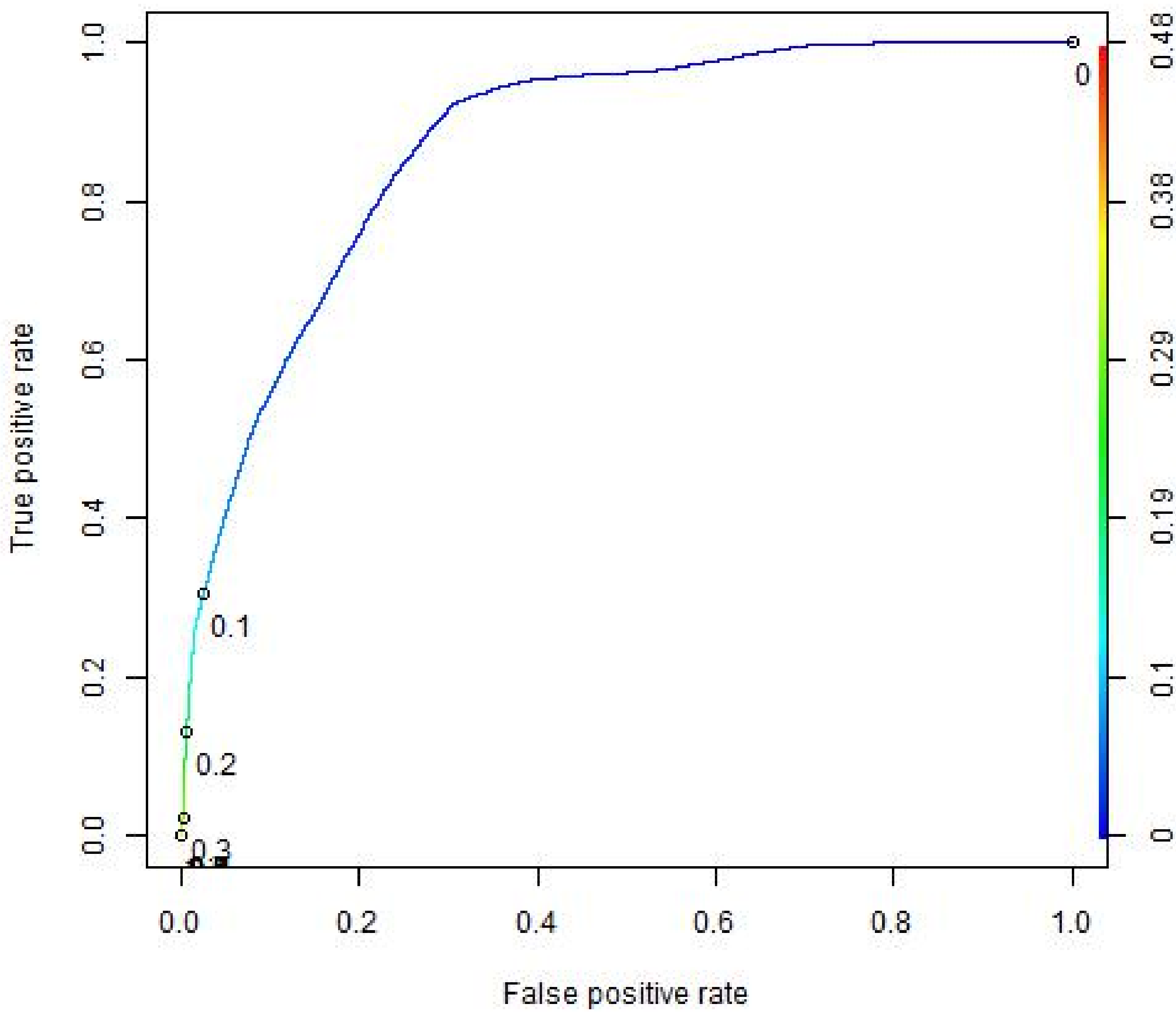

Figure 4: Aspen ROC for Individuated Significant Data

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 Aspen Logisitic Model - Sig Vars

Calculate Aspen Aggregated Data Logisitic Model Accuracy - Significant Vars

Find best Aspen threshold for Aggregated Data using significant variables.

```r
result = calcLogisticModelAccuracy (forestTrain$Aspen, Aspen_Agg_Train_predict,
                                    0.0, 1, 10, "Aspen", "Other", 1,1)
```

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
## [1] "Thresh=0, Accuracy=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=97.4%, BaseAcc(Other)=98.3%, Sens=2.9%, Spec=99%, Sens^2+Spec^2=0.981"
## [1] "Thresh=0.2, Accuracy=98.2%, BaseAcc(Other)=98.3%, Sens=0%, Spec=99.9%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.3, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.4, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.5, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.6, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.7, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.8, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.9, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=1, Accuracy=98.3%, BaseAcc(Other)=98.3%, 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=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.01, Accuracy=51.1%, BaseAcc(Other)=98.3%, Sens=89%, Spec=50.4%, Sens^2+Spec^2=1.048"
## [1] "Thresh=0.02, Accuracy=74.5%, BaseAcc(Other)=98.3%, Sens=68.8%, Spec=74.6%, Sens^2+Spec^2=1.031"
## [1] "Thresh=0.03, Accuracy=85.2%, BaseAcc(Other)=98.3%, Sens=52.8%, Spec=85.8%, Sens^2+Spec^2=1.015"
## [1] "Thresh=0.04, Accuracy=90.7%, BaseAcc(Other)=98.3%, Sens=39.3%, Spec=91.5%, Sens^2+Spec^2=0.993"
## [1] "Thresh=0.05, Accuracy=93.5%, BaseAcc(Other)=98.3%, Sens=27.2%, Spec=94.7%, Sens^2+Spec^2=0.97"
## [1] "Thresh=0.06, Accuracy=95.2%, BaseAcc(Other)=98.3%, Sens=18.6%, Spec=96.5%, Sens^2+Spec^2=0.966"
## [1] "Thresh=0.07, Accuracy=96.2%, BaseAcc(Other)=98.3%, Sens=12.1%, Spec=97.6%, Sens^2+Spec^2=0.968"
## [1] "Thresh=0.08, Accuracy=96.8%, BaseAcc(Other)=98.3%, Sens=8.1%, Spec=98.2%, Sens^2+Spec^2=0.972"
## [1] "Thresh=0.09, Accuracy=97.1%, BaseAcc(Other)=98.3%, Sens=5.4%, Spec=98.7%, Sens^2+Spec^2=0.977"
## [1] "Thresh=0.1, Accuracy=97.4%, BaseAcc(Other)=98.3%, Sens=2.9%, Spec=99%, Sens^2+Spec^2=0.981"
## [1] "Thresh=0.11, Accuracy=97.6%, BaseAcc(Other)=98.3%, Sens=1.5%, Spec=99.2%, Sens^2+Spec^2=0.985"
## [1] "Thresh=0.12, Accuracy=97.7%, BaseAcc(Other)=98.3%, Sens=0.4%, Spec=99.4%, Sens^2+Spec^2=0.988"
## [1] "Thresh=0.13, Accuracy=97.9%, BaseAcc(Other)=98.3%, Sens=0%, Spec=99.5%, Sens^2+Spec^2=0.99"
## [1] "Thresh=0.14, Accuracy=98%, BaseAcc(Other)=98.3%, Sens=0%, Spec=99.6%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.15, Accuracy=98.1%, BaseAcc(Other)=98.3%, Sens=0%, Spec=99.7%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.16, Accuracy=98.1%, BaseAcc(Other)=98.3%, Sens=0%, Spec=99.7%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.17, Accuracy=98.2%, BaseAcc(Other)=98.3%, Sens=0%, Spec=99.8%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.18, Accuracy=98.2%, BaseAcc(Other)=98.3%, Sens=0%, Spec=99.8%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.19, Accuracy=98.2%, BaseAcc(Other)=98.3%, Sens=0%, Spec=99.9%, Sens^2+Spec^2=-2"
## [1] "Best Sensitivity_Specificity threshold= 0.01 inc= 0.01"
## [1] "========================================="
## [1] "start= 0 end= 0.02 inc= 0.001"
## [1] "Thresh=0, Accuracy=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.001, Accuracy=8.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=7.1%, Sens^2+Spec^2=-2"
```

```
## [1] "Thresh=0.002, Accuracy=14%, BaseAcc(Other)=98.3%, Sens=99.6%, Spec=12.6%, Sens^2+Spec^2=1.008"
## [1] "Thresh=0.003, Accuracy=20%, BaseAcc(Other)=98.3%, Sens=98.5%, Spec=18.7%, Sens^2+Spec^2=1.006"
## [1] "Thresh=0.004, Accuracy=25.6%, BaseAcc(Other)=98.3%, Sens=97.1%, Spec=24.4%, Sens^2+Spec^2=1.002
## [1] "Thresh=0.005, Accuracy=30.7%, BaseAcc(Other)=98.3%, Sens=95.3%, Spec=29.7%, Sens^2+Spec^2=0.997
## [1] "Thresh=0.006, Accuracy=35.4%, BaseAcc(Other)=98.3%, Sens=94.2%, Spec=34.4%, Sens^2+Spec^2=1.007
## [1] "Thresh=0.007, Accuracy=39.6%, BaseAcc(Other)=98.3%, Sens=93%, Spec=38.8%, Sens^2+Spec^2=1.015"
## [1] "Thresh=0.008, Accuracy=43.7%, BaseAcc(Other)=98.3%, Sens=91.7%, Spec=42.9%, Sens^2+Spec^2=1.025
## [1] "Thresh=0.009, Accuracy=47.5%, BaseAcc(Other)=98.3%, Sens=90.3%, Spec=46.8%, Sens^2+Spec^2=1.035
## [1] "Thresh=0.01, Accuracy=51.1%, BaseAcc(Other)=98.3%, Sens=89%, Spec=50.4%, Sens^2+Spec^2=1.048"
## [1] "Thresh=0.011, Accuracy=54.3%, BaseAcc(Other)=98.3%, Sens=87.1%, Spec=53.7%, Sens^2+Spec^2=1.048
## [1] "Thresh=0.012, Accuracy=57.3%, BaseAcc(Other)=98.3%, Sens=85.2%, Spec=56.8%, Sens^2+Spec^2=1.049
## [1] "Thresh=0.013, Accuracy=60%, BaseAcc(Other)=98.3%, Sens=82.8%, Spec=59.7%, Sens^2+Spec^2=1.043"
## [1] "Thresh=0.014, Accuracy=62.7%, BaseAcc(Other)=98.3%, Sens=80.3%, Spec=62.4%, Sens^2+Spec^2=1.035
## [1] "Thresh=0.015, Accuracy=65.1%, BaseAcc(Other)=98.3%, Sens=78.3%, Spec=64.8%, Sens^2+Spec^2=1.034
## [1] "Thresh=0.016, Accuracy=67.3%, BaseAcc(Other)=98.3%, Sens=76.1%, Spec=67.1%, Sens^2+Spec^2=1.031
## [1] "Thresh=0.017, Accuracy=69.3%, BaseAcc(Other)=98.3%, Sens=73.9%, Spec=69.2%, Sens^2+Spec^2=1.027
## [1] "Thresh=0.018, Accuracy=71.2%, BaseAcc(Other)=98.3%, Sens=72.3%, Spec=71.2%, Sens^2+Spec^2=1.03"
## [1] "Thresh=0.019, Accuracy=72.9%, BaseAcc(Other)=98.3%, Sens=70.5%, Spec=72.9%, Sens^2+Spec^2=1.03"
## [1] "========================================"
## [1] "Best Threshold=0.012"
## [1] "Best Sensitivity_Specificity=1.04979651699068"
```

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

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

```r
result = calcLogisticModelAccuracy (forestTrain$Aspen, Aspen_Agg_Train_predict,
                        curThresh, curThresh, 1, "Aspen", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.012"
## [1] "predicted performance="
##                  Predicted
## Actual            FALSE=Predict:Other TRUE=Predict:Aspen
##   0=Actual:Other     227551 (TN)         172513 (FP)
##   1=Actual:Aspen     982 (FN)            5663 (TP)
## [1] "Sensitivity= 0.85221971407073 (True positive rate of Aspen = TP/(TP+FN) = 5663 /( 5663 + 982 ))
## [1] "Specificity= 0.568786494160934 (True negative rate of Other = TN/(TN+FP) = 227551 /( 227551 + 1
## [1] "Sens^2+Spec^2=1.049"
## [1] "Baseline (Other) Accuracy=0.983661"
## [1] "Logistic Accuracy=0.573417"
```

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

```r
result = calcLogisticModelAccuracy (forestTest$Aspen, Aspen_Agg_Test_predict,
                        curThresh, curThresh, 1, "Aspen", "Other", 3,
                        saveFile=saveFileName, desc="Aspen Sig Aggregate Vars",
                        AIC=Aspen_Agg_Sig_aic, AUC=Aspen_Agg_Sig_ROC_AUC)
```

```
## [1] "Model Performance for threshold= 0.012"
## [1] "predicted performance="
##                  Predicted
## Actual            FALSE=Predict:Other TRUE=Predict:Aspen
##   0=Actual:Other     97720 (TN)          73735 (FP)
##   1=Actual:Aspen     418 (FN)            2430 (TP)
```

```
## [1] "Sensitivity= 0.853230337078652 (True positive rate of Aspen = TP/(TP+FN) = 2430 /( 2430 + 418 )
## [1] "Specificity= 0.569945466740544 (True negative rate of Other = TN/(TN+FP) = 97720 /( 97720 + 7373
## [1] "Sens^2+Spec^2=1.052"
## [1] "Baseline (Other) Accuracy=0.98366"
## [1] "Logistic Accuracy=0.574574"
```

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

Calculate Aspen Individuated Data Logisitic Model Accuracy - Significant Vars

Find best Aspen threshold for Inividuated Data using significant variables.

```
result = calcLogisticModelAccuracy (forestTrain$Aspen, Aspen_Ind_Train_predict,
                        0.0, 1, 10, "Aspen", "Other", 1,1)
```

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
## [1] "Thresh=0, Accuracy=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=96.4%, BaseAcc(Other)=98.3%, Sens=30.5%, Spec=97.5%, Sens^2+Spec^2=1.045"
## [1] "Thresh=0.2, Accuracy=97.9%, BaseAcc(Other)=98.3%, Sens=13.2%, Spec=99.4%, Sens^2+Spec^2=1.005"
## [1] "Thresh=0.3, Accuracy=98.2%, BaseAcc(Other)=98.3%, Sens=2.3%, Spec=99.8%, Sens^2+Spec^2=0.997"
## [1] "Thresh=0.4, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=99.9%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.5, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.6, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.7, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.8, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.9, Accuracy=98.3%, BaseAcc(Other)=98.3%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Thresh=1, Accuracy=98.3%, BaseAcc(Other)=98.3%, 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=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.01, Accuracy=66.9%, BaseAcc(Other)=98.3%, Sens=93.5%, Spec=66.5%, Sens^2+Spec^2=1.317"
## [1] "Thresh=0.02, Accuracy=79.9%, BaseAcc(Other)=98.3%, Sens=75.8%, Spec=80%, Sens^2+Spec^2=1.215"
## [1] "Thresh=0.03, Accuracy=86%, BaseAcc(Other)=98.3%, Sens=63.5%, Spec=86.4%, Sens^2+Spec^2=1.15"
## [1] "Thresh=0.04, Accuracy=89.5%, BaseAcc(Other)=98.3%, Sens=55.4%, Spec=90%, Sens^2+Spec^2=1.119"
## [1] "Thresh=0.05, Accuracy=91.9%, BaseAcc(Other)=98.3%, Sens=48.9%, Spec=92.6%, Sens^2+Spec^2=1.097"
## [1] "Thresh=0.06, Accuracy=93.5%, BaseAcc(Other)=98.3%, Sens=42.8%, Spec=94.3%, Sens^2+Spec^2=1.074"
## [1] "Thresh=0.07, Accuracy=94.6%, BaseAcc(Other)=98.3%, Sens=38.4%, Spec=95.5%, Sens^2+Spec^2=1.061"
## [1] "Thresh=0.08, Accuracy=95.4%, BaseAcc(Other)=98.3%, Sens=35.1%, Spec=96.4%, Sens^2+Spec^2=1.053"
## [1] "Thresh=0.09, Accuracy=96%, BaseAcc(Other)=98.3%, Sens=32.2%, Spec=97%, Sens^2+Spec^2=1.046"
## [1] "Thresh=0.1, Accuracy=96.4%, BaseAcc(Other)=98.3%, Sens=30.5%, Spec=97.5%, Sens^2+Spec^2=1.045"
## [1] "Thresh=0.11, Accuracy=96.8%, BaseAcc(Other)=98.3%, Sens=28.5%, Spec=97.9%, Sens^2+Spec^2=1.041"
## [1] "Thresh=0.12, Accuracy=97.1%, BaseAcc(Other)=98.3%, Sens=27.2%, Spec=98.2%, Sens^2+Spec^2=1.04"
## [1] "Thresh=0.13, Accuracy=97.3%, BaseAcc(Other)=98.3%, Sens=25.8%, Spec=98.5%, Sens^2+Spec^2=1.037"
## [1] "Thresh=0.14, Accuracy=97.4%, BaseAcc(Other)=98.3%, Sens=24%, Spec=98.7%, Sens^2+Spec^2=1.031"
```

```
## [1] "Thresh=0.15, Accuracy=97.6%, BaseAcc(Other)=98.3%, Sens=21.7%, Spec=98.8%, Sens^2+Spec^2=1.024"
## [1] "Thresh=0.16, Accuracy=97.7%, BaseAcc(Other)=98.3%, Sens=20%, Spec=98.9%, Sens^2+Spec^2=1.02"
## [1] "Thresh=0.17, Accuracy=97.7%, BaseAcc(Other)=98.3%, Sens=18.3%, Spec=99.1%, Sens^2+Spec^2=1.016"
## [1] "Thresh=0.18, Accuracy=97.8%, BaseAcc(Other)=98.3%, Sens=16.7%, Spec=99.2%, Sens^2+Spec^2=1.012"
## [1] "Thresh=0.19, Accuracy=97.9%, BaseAcc(Other)=98.3%, Sens=14.6%, Spec=99.3%, Sens^2+Spec^2=1.007"
## [1] "Best Sensitivity_Specificity threshold= 0.01 inc= 0.01"
## [1] "========================================"
## [1] "start= 0 end= 0.02 inc= 0.001"
## [1] "Thresh=0, Accuracy=1.6%, BaseAcc(Other)=98.3%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.001, Accuracy=25.3%, BaseAcc(Other)=98.3%, Sens=99.9%, Spec=24%, Sens^2+Spec^2=1.057"
## [1] "Thresh=0.002, Accuracy=34.6%, BaseAcc(Other)=98.3%, Sens=99%, Spec=33.5%, Sens^2+Spec^2=1.093"
## [1] "Thresh=0.003, Accuracy=41.9%, BaseAcc(Other)=98.3%, Sens=97.5%, Spec=40.9%, Sens^2+Spec^2=1.119
## [1] "Thresh=0.004, Accuracy=47.5%, BaseAcc(Other)=98.3%, Sens=96.5%, Spec=46.7%, Sens^2+Spec^2=1.15"
## [1] "Thresh=0.005, Accuracy=52.2%, BaseAcc(Other)=98.3%, Sens=96.1%, Spec=51.5%, Sens^2+Spec^2=1.189
## [1] "Thresh=0.006, Accuracy=56.2%, BaseAcc(Other)=98.3%, Sens=95.8%, Spec=55.5%, Sens^2+Spec^2=1.227
## [1] "Thresh=0.007, Accuracy=59.5%, BaseAcc(Other)=98.3%, Sens=95.5%, Spec=58.9%, Sens^2+Spec^2=1.259
## [1] "Thresh=0.008, Accuracy=62.4%, BaseAcc(Other)=98.3%, Sens=95%, Spec=61.8%, Sens^2+Spec^2=1.286"
## [1] "Thresh=0.009, Accuracy=64.8%, BaseAcc(Other)=98.3%, Sens=94.3%, Spec=64.3%, Sens^2+Spec^2=1.304
## [1] "Thresh=0.01, Accuracy=66.9%, BaseAcc(Other)=98.3%, Sens=93.5%, Spec=66.5%, Sens^2+Spec^2=1.317"
## [1] "Thresh=0.011, Accuracy=68.9%, BaseAcc(Other)=98.3%, Sens=92.7%, Spec=68.5%, Sens^2+Spec^2=1.33"
## [1] "Thresh=0.012, Accuracy=70.6%, BaseAcc(Other)=98.3%, Sens=91.1%, Spec=70.3%, Sens^2+Spec^2=1.326
## [1] "Thresh=0.013, Accuracy=72.2%, BaseAcc(Other)=98.3%, Sens=89.1%, Spec=71.9%, Sens^2+Spec^2=1.313
## [1] "Thresh=0.014, Accuracy=73.6%, BaseAcc(Other)=98.3%, Sens=87%, Spec=73.3%, Sens^2+Spec^2=1.296"
## [1] "Thresh=0.015, Accuracy=74.8%, BaseAcc(Other)=98.3%, Sens=85.1%, Spec=74.7%, Sens^2+Spec^2=1.283
## [1] "Thresh=0.016, Accuracy=76%, BaseAcc(Other)=98.3%, Sens=83.4%, Spec=75.9%, Sens^2+Spec^2=1.273"
## [1] "Thresh=0.017, Accuracy=77.1%, BaseAcc(Other)=98.3%, Sens=81.6%, Spec=77%, Sens^2+Spec^2=1.26"
## [1] "Thresh=0.018, Accuracy=78.1%, BaseAcc(Other)=98.3%, Sens=79.6%, Spec=78.1%, Sens^2+Spec^2=1.244
## [1] "Thresh=0.019, Accuracy=79.1%, BaseAcc(Other)=98.3%, Sens=77.9%, Spec=79.1%, Sens^2+Spec^2=1.233
## [1] "========================================"
## [1] "Best Threshold=0.011"
## [1] "Best Sensitivity_Specificity=1.33004745354717"
```

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

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

```r
result = calcLogisticModelAccuracy (forestTrain$Aspen, Aspen_Ind_Train_predict,
                        curThresh, curThresh, 1, "Aspen", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.011"
## [1] "predicted performance="
##                  Predicted
## Actual            FALSE=Predict:Other TRUE=Predict:Aspen
##   0=Actual:Other     274147 (TN)         125917 (FP)
##   1=Actual:Aspen     481 (FN)            6164 (TP)
## [1] "Sensitivity= 0.927614747930775 (True positive rate of Aspen = TP/(TP+FN) = 6164 /( 6164 + 481 )
## [1] "Specificity= 0.685257858742601 (True negative rate of Other = TN/(TN+FP) = 274147 /( 274147 + 1
## [1] "Sens^2+Spec^2=1.33"
## [1] "Baseline (Other) Accuracy=0.983661"
## [1] "Logistic Accuracy=0.689217"
```

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

```
result = calcLogisticModelAccuracy (forestTest$Aspen, Aspen_Ind_Test_predict,
                        curThresh, curThresh, 1, "Aspen", "Other", 3,
                        saveFile=saveFileName, desc="Aspen Sig Individualized Vars",
                        AIC=Aspen_Ind_Sig_aic, AUC=Aspen_Ind_Sig_ROC_AUC)
```

```
## [1] "Model Performance for threshold= 0.011"
## [1] "predicted performance="
##                  Predicted
## Actual            FALSE=Predict:Other TRUE=Predict:Aspen
##   0=Actual:Other     117564 (TN)         53891 (FP)
##   1=Actual:Aspen     193 (FN)            2655 (TP)
## [1] "Sensitivity= 0.932233146067416 (True positive rate of Aspen = TP/(TP+FN) = 2655 /( 2655 + 193 )
## [1] "Specificity= 0.685684290338573 (True negative rate of Other = TN/(TN+FP) = 117564 /( 117564 + 5
## [1] "Sens^2+Spec^2=1.339"
## [1] "Baseline (Other) Accuracy=0.98366"
## [1] "Logistic Accuracy=0.689712"
```

```
list[RC, Aspen_Ind_Sig_model_acc, Aspen_Ind_Sig_baseline_acc,
      TN, FN, FP, TP, Aspen_Ind_Sig_sens, Aspen_Ind_Sig_spec] <- result
  if (RC != "OK") {
    print(paste("Error - terminating:",RC))
    knitr:knit_exit()
  }
  Aspen_Ind_Sig_model_acc = as.integer(as.numeric(Aspen_Ind_Sig_model_acc)*1000)/10
  Aspen_Ind_Sig_baseline_acc = as.integer(as.numeric(Aspen_Ind_Sig_baseline_acc)*1000)/10
  Aspen_Ind_Sig_sens = as.integer(as.numeric(Aspen_Ind_Sig_sens)*1000)/10
  Aspen_Ind_Sig_spec = as.integer(as.numeric(Aspen_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	Threshhold
Aspen Aggregate All Vars	68.9%	86.3%	68.6%	1737968	82.7%	0.015
Aspen Individual All Vars	69.4%	88.4%	69.1%	568029	83.1%	0.014
Aspen Aggregate Sig Vars	57.4%	85.3%	56.9%	60588	79.1%	0.012
Aspen Individual Sig Vars	68.9%	93.2%	68.5%	53086	87.4%	0.011

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 Aspen 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.

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

```
## [1] "Forest Cover Logistic script ended at 2018-08-12 15:01:30"
```