Capstone Data Logistic Regression - Predict Lodgepole Pine

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 19:01:57"
```

```
#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"
#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
```

```
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 19:02:39"
```

```
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.

```
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") # for calcLogisticModelAccuracy function
bestThreshIndex=11
#save("calcLogisticModelAccuracy", file="logisticAccuracy.Rdata")
```

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$LodgepolePine)
#table(forestTest$LodgepolePine)

# 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 ##################
```

Lodgepole Pine Logistic Regression

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

Lodgepole Pine Logistic Regression - All Variables

Create Lodgepole Pine Logistic Model - All Vars

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

Lodgepole Pine 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("LodgepolePine aggregated Logistic Model Calculation started at",curTime))
```

```
## [1] "LodgepolePine aggregated Logistic Model Calculation started at 2018-08-12 19:02:45"
```

```
  Lodge_Agg_LogMod =
    glm(LodgepolePine ~
          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)

  # save model for later use
  Lodge_Agg_All_LogMod = Lodge_Agg_LogMod
  save("Lodge_Agg_All_LogMod", file="Lodge_Agg_All_LogMod.Rdata")

  Lodge_Agg_All_aic<-as.integer(Lodge_Agg_LogMod$aic)
  Lodge_Agg_All_aic
```

```
## [1] 418219
```

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

```
## [1] "LodgepolePine aggregated Logistic Model Calculation completed at 2018-08-12 19:04:35"
```

Check the coefficients for the Lodgepole Pine model using all aggregated data.

```
summary(Lodge_Agg_LogMod)
```

```
##
## Call:
## glm(formula = LodgepolePine ~ 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
## -2.9282  -0.8783  -0.0957   0.8612   3.6264
##
## Coefficients: (2 not defined because of singularities)
##               Estimate Std. Error  z value Pr(>|z|)
## (Intercept)  4.216e+00  2.777e-01   15.182  < 2e-16 ***
## Elev        -5.501e-03  3.050e-05 -180.359  < 2e-16 ***
## Aspect      -3.117e-04  4.641e-05   -6.715 1.88e-11 ***
## Slope        1.541e-02  1.424e-03   10.824  < 2e-16 ***
## H2OHD        1.536e-03  2.459e-05   62.482  < 2e-16 ***
## H2OVD        4.869e-04  8.996e-05    5.412 6.22e-08 ***
## RoadHD       8.098e-05  3.110e-06   26.037  < 2e-16 ***
## FirePtHD     1.346e-05  3.427e-06    3.929 8.54e-05 ***
## Shade9AM     2.064e-04  1.553e-03    0.133   0.8943
## Shade12PM    2.523e-02  1.280e-03   19.710  < 2e-16 ***
## Shade3PM    -2.880e-03  1.274e-03   -2.260   0.0238 *
## RWwild       4.706e+00  4.000e-02  117.647  < 2e-16 ***
## NEwild       5.203e+00  4.222e-02  123.241  < 2e-16 ***
## CMwild       4.556e+00  3.575e-02  127.427  < 2e-16 ***
## CPwild              NA         NA       NA       NA
## ST01        -1.369e+01  2.891e+01   -0.474   0.6358
## ST02        -2.626e+00  8.598e-02  -30.543  < 2e-16 ***
## ST03        -1.133e+00  8.752e-02  -12.949  < 2e-16 ***
## ST04        -1.535e+00  7.719e-02  -19.888  < 2e-16 ***
## ST05        -1.355e+01  4.002e+01   -0.339   0.7349
## ST06         1.071e+00  9.032e-02   11.853  < 2e-16 ***
## ST07         1.621e+01  1.710e+02    0.095   0.9245
## ST08         1.644e+00  2.078e-01    7.913 2.52e-15 ***
## ST09         1.271e+00  1.240e-01   10.245  < 2e-16 ***
## ST10         7.310e-01  7.428e-02    9.841  < 2e-16 ***
## ST11         1.461e+00  7.579e-02   19.271  < 2e-16 ***
## ST12         2.752e+00  7.469e-02   36.841  < 2e-16 ***
## ST13         2.099e+00  7.390e-02   28.402  < 2e-16 ***
## ST14        -1.566e+01  5.665e+01   -0.276   0.7822
## ST15        -1.338e+01  8.341e+02   -0.016   0.9872
## ST16         1.204e+00  8.780e-02   13.713  < 2e-16 ***
## ST17        -6.423e-01  8.717e-02   -7.368 1.73e-13 ***
## ST18         1.224e+00  1.156e-01   10.585  < 2e-16 ***
## ST19         1.282e+00  8.027e-02   15.976  < 2e-16 ***
## ST20         1.210e+00  7.507e-02   16.123  < 2e-16 ***
## ST21        -1.574e+00  2.807e-01   -5.607 2.06e-08 ***
## ST22         9.140e-01  7.087e-02   12.897  < 2e-16 ***
## ST23         1.243e+00  7.015e-02   17.716  < 2e-16 ***
## ST24         1.854e+00  7.159e-02   25.895  < 2e-16 ***
## ST25         3.098e+00  1.485e-01   20.861  < 2e-16 ***
## ST26         2.525e+00  9.701e-02   26.025  < 2e-16 ***
## ST27         1.505e+00  1.026e-01   14.660  < 2e-16 ***
## ST28         3.350e+00  1.926e-01   17.389  < 2e-16 ***
## ST29         2.000e+00  7.011e-02   28.525  < 2e-16 ***
```

```
## ST30         1.931e+00  7.173e-02   26.921  < 2e-16 ***
## ST31         1.939e+00  7.148e-02   27.121  < 2e-16 ***
## ST32         2.265e+00  7.039e-02   32.177  < 2e-16 ***
## ST33         2.000e+00  7.068e-02   28.297  < 2e-16 ***
## ST34         3.360e+00  1.214e-01   27.677  < 2e-16 ***
## ST35        -1.878e+00  3.853e-01   -4.874 1.09e-06 ***
## ST36         1.729e+00  2.335e-01    7.406 1.30e-13 ***
## ST37        -1.158e+01  9.901e+01   -0.117   0.9069
## ST38        -3.934e-02  8.212e-02   -0.479   0.6319
## ST39        -5.306e-01  9.331e-02   -5.687 1.29e-08 ***
## ST40                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: 563568  on 406708  degrees of freedom
## Residual deviance: 418114  on 406656  degrees of freedom
## AIC: 418220
##
## Number of Fisher Scoring iterations: 14
```

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.

Lodgepole Pine 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("LodgepolePine Individual Logistic Model Calculation started at",curTime))
```

```
## [1] "LodgepolePine Individual Logistic Model Calculation started at 2018-08-12 19:04:35"
```

```
Lodge_Ind_LogMod =
  glm(LodgepolePine ~
        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:
            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
```

```
  # save model for later use
  Lodge_Ind_All_LogMod = Lodge_Ind_LogMod
  save("Lodge_Ind_All_LogMod", file="Lodge_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 <-------

  Lodge_Ind_All_aic<-as.integer(Lodge_Ind_LogMod$aic)
  Lodge_Ind_All_aic
```

```
## [1] 418474
```

```
  summary(Lodge_Ind_LogMod)
```

```
##
## Call:
## glm(formula = LodgepolePine ~ 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
```

```
## -2.9325  -0.8765  -0.0946   0.8602   3.6419
##
## Coefficients: (21 not defined because of singularities)
##                            Estimate Std. Error  z value Pr(>|z|)
## (Intercept)              7.252e+00  3.174e-01   22.846  < 2e-16 ***
## Elev                    -5.530e-03  3.062e-05 -180.588  < 2e-16 ***
## Aspect                  -3.153e-04  4.644e-05   -6.789 1.13e-11 ***
## Slope                    1.553e-02  1.425e-03   10.902  < 2e-16 ***
## H2OHD                    1.544e-03  2.460e-05   62.767  < 2e-16 ***
## H2OVD                    4.626e-04  8.975e-05    5.154 2.55e-07 ***
## RoadHD                   7.959e-05  3.116e-06   25.546  < 2e-16 ***
## FirePtHD                 1.418e-05  3.431e-06    4.131 3.61e-05 ***
## Shade9AM                -4.269e-04  1.551e-03   -0.275 0.783104
## Shade12PM                2.609e-02  1.280e-03   20.385  < 2e-16 ***
## Shade3PM                -3.477e-03  1.273e-03   -2.731 0.006307 **
## RWwild                   4.725e+00  3.984e-02  118.596  < 2e-16 ***
## NEwild                   5.240e+00  4.199e-02  124.776  < 2e-16 ***
## CMwild                   4.567e+00  3.555e-02  128.462  < 2e-16 ***
## CPwild                          NA         NA       NA       NA
## Montane_low              2.463e+12  4.698e+11    5.242 1.59e-07 ***
## Montane                  2.463e+12  4.698e+11    5.242 1.59e-07 ***
## SubAlpine               -1.001e+00  1.386e-01   -7.218 5.27e-13 ***
## Alpine                  -4.366e+00  1.850e-01  -23.605  < 2e-16 ***
## Dry                     -2.463e+12  4.698e+11   -5.242 1.59e-07 ***
## Non_Dry                 -2.463e+12  4.698e+11   -5.242 1.59e-07 ***
## Alluvium                -3.324e+00  1.669e-01  -19.915  < 2e-16 ***
## Glacial                 -9.682e-01  2.577e-02  -37.568  < 2e-16 ***
## Sed_mix                         NA         NA       NA       NA
## Ign_Meta                        NA         NA       NA       NA
## Aquolis_cmplx            2.463e+12  4.698e+11    5.242 1.59e-07 ***
## Argiborolis_Pachic              NA         NA       NA       NA
## Borohemists_cmplx        2.787e+00  2.934e-01    9.500  < 2e-16 ***
## Bross                    3.231e+00  2.922e-01   11.055  < 2e-16 ***
## Bullwark                -1.622e+00  1.058e-01  -15.334  < 2e-16 ***
## Bullwark_Cmplx          -1.535e-01  1.379e-01   -1.113 0.265524
## Catamount                9.256e-02  8.583e-02    1.078 0.280830
## Catamount_cmplx          6.311e-02  2.433e-02    2.593 0.009503 **
## Cathedral               -1.636e-01  1.139e+04    0.000 0.999989
## Como                     5.584e-02  1.039e-01    0.537 0.590994
## Cryaquepts_cmplx        -4.540e-01  4.258e-01   -1.066 0.286311
## Cryaquepts_Typic         2.259e+00  2.241e-01   10.081  < 2e-16 ***
## Cryaquolls               3.293e+00  1.493e-01   22.056  < 2e-16 ***
## Cryaquolls_cmplx         1.449e+00  1.309e-01   11.071  < 2e-16 ***
## Cryaquolls_Typic        -1.255e-01  3.209e-01   -0.391 0.695669
## Cryaquolls_Typic_cmplx   3.292e-01  2.000e-02   16.463  < 2e-16 ***
## Cryoborolis_cmplx               NA         NA       NA       NA
## Cryorthents              6.627e-01  1.617e-01    4.100 4.14e-05 ***
## Cryorthents_cmplx       -2.128e+01  2.585e+04   -0.001 0.999343
## Cryumbrepts                     NA         NA       NA       NA
## Cryumbrepts_cmplx               NA         NA       NA       NA
## Gateview                        NA         NA       NA       NA
## Gothic                   2.569e+01  4.448e+04    0.001 0.999539
## Granile                  5.190e-01  1.225e-01    4.237 2.27e-05 ***
## Haploborolis             2.329e+01  9.203e+03    0.003 0.997981
```

```
## Legault                 -2.430e-01  1.407e-01   -1.728 0.084043 .
## Legault_cmplx                   NA         NA       NA       NA
## Leighcan                -6.315e-02  1.010e-01   -0.625 0.531940
## Leighcan_cmplx           2.328e-01  1.314e-01    1.771 0.076487 .
## Leighcan_warm           -1.219e-02  1.332e-01   -0.092 0.927070
## Moran                           NA         NA       NA       NA
## Ratake                   2.289e+01  9.203e+03    0.002 0.998015
## Ratake_cmplx             2.183e+01  9.203e+03    0.002 0.998107
## Rogert                          NA         NA       NA       NA
## Supervisor_Limber_cmplx         NA         NA       NA       NA
## Troutville                      NA         NA       NA       NA
## Unspecified              2.463e+12  4.698e+11    5.242 1.59e-07 ***
## Vanet                           NA         NA       NA       NA
## Wetmore                  2.551e+01  9.203e+03    0.003 0.997789
## Bouldery_ext                    NA         NA       NA       NA
## Rock_Land               -7.086e-02  1.827e-02   -3.879 0.000105 ***
## Rock_Land_cmplx          7.600e-01  9.157e-02    8.299  < 2e-16 ***
## Rock_Outcrop                    NA         NA       NA       NA
## Rock_Outcrop_cmplx       2.921e-02  8.560e-02    0.341 0.732946
## 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: 563568  on 406708  degrees of freedom
## Residual deviance: 418369  on 406656  degrees of freedom
## AIC: 418475
##
## Number of Fisher Scoring iterations: 25
```

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

```
## [1] "LodgepolePine Individual Logistic Model Calculation completed at 2018-08-12 19:09:48"
```

```
  #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 Lodgepole Pine Logistic Model Probabilities - All Aggregated Vars

Lodgepole Pine Probabilities - All Aggregated Data

Predict the probability of Lodgepole Pine for aggregated Data - all variables.

```
# Predict Lodgepole Pine Agg Data - all variables

  Lodge_Agg_Train_predict= predict(Lodge_Agg_LogMod, type="response")
  Lodge_Agg_Train_Logit= predict(Lodge_Agg_LogMod)
```

```
  summary(Lodge_Agg_Train_predict)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.2595  0.5047  0.4876  0.7231  1.0000
```

```
  str(Lodge_Agg_Train_predict)
```

```
##  Named num [1:406709] 0.935 0.935 0.942 0.926 0.95 ...
##  - attr(*, "names")= chr [1:406709] "1" "2" "3" "4" ...
```

```
  #plot(table(Lodge_Agg_Train_predict))
  #plot(table(Lodge_Agg_Train_Logit))
  dens<-data.frame(table(Lodge_Agg_Train_predict))
# str(dens)

  Lodge_Agg_Test_predict= predict(Lodge_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(Lodge_Agg_Test_predict)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.2602  0.5038  0.4874  0.7231  1.0000
```

```
   str(Lodge_Agg_Test_predict)
```

```
##  Named num [1:174303] 0.903 0.919 0.883 0.908 0.883 ...
##  - attr(*, "names")= chr [1:174303] "1" "2" "3" "4" ...
```

Lodgepole Pine Probabilities - All Individuated Data

Predict the probability of Lodgepole Pine for Individual Data - all variables.

```
  Lodge_Ind_Train_predict= predict(Lodge_Ind_LogMod, type="response")
  summary(Lodge_Ind_Train_predict)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.2581  0.5045  0.4876  0.7239  1.0000
```

```
  Lodge_Ind_Test_predict= predict(Lodge_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(Lodge_Ind_Test_predict)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.2585  0.5034  0.4874  0.7236  1.0000
```

Lodgepole Pine Receiver Operating Characteristic (ROC) - All Vars

Lodgepole Pine 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_Lodge_Agg = prediction(Lodge_Agg_Train_predict, forestTrain$LodgepolePine)
    summary(ROCpred_Lodge_Agg)
    ROCperf_Lodge_Agg = performance(ROCpred_Lodge_Agg, "tpr", "fpr")
    summary(ROCperf_Lodge_Agg)

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

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

```
## [1] "ROC graph 1 started at 2018-08-12 19:09:53"
## [1] "Lodge_Agg_All_ROC_AUC= 82.6"
```

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

Lodgepole Pine 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_Lodge_Ind = prediction(Lodge_Ind_Train_predict, forestTrain$LodgepolePine)
    summary(ROCpred_Lodge_Ind)
    ROCperf_Lodge_Ind = performance(ROCpred_Lodge_Ind, "tpr", "fpr")
    summary(ROCperf_Lodge_Ind)

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

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

```
## [1] "ROCR graph 2 started at 2018-08-12 19:12:54"
## [1] "Lodge_Ind_All_ROC_AUC= 82.6"
```

```
## 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.

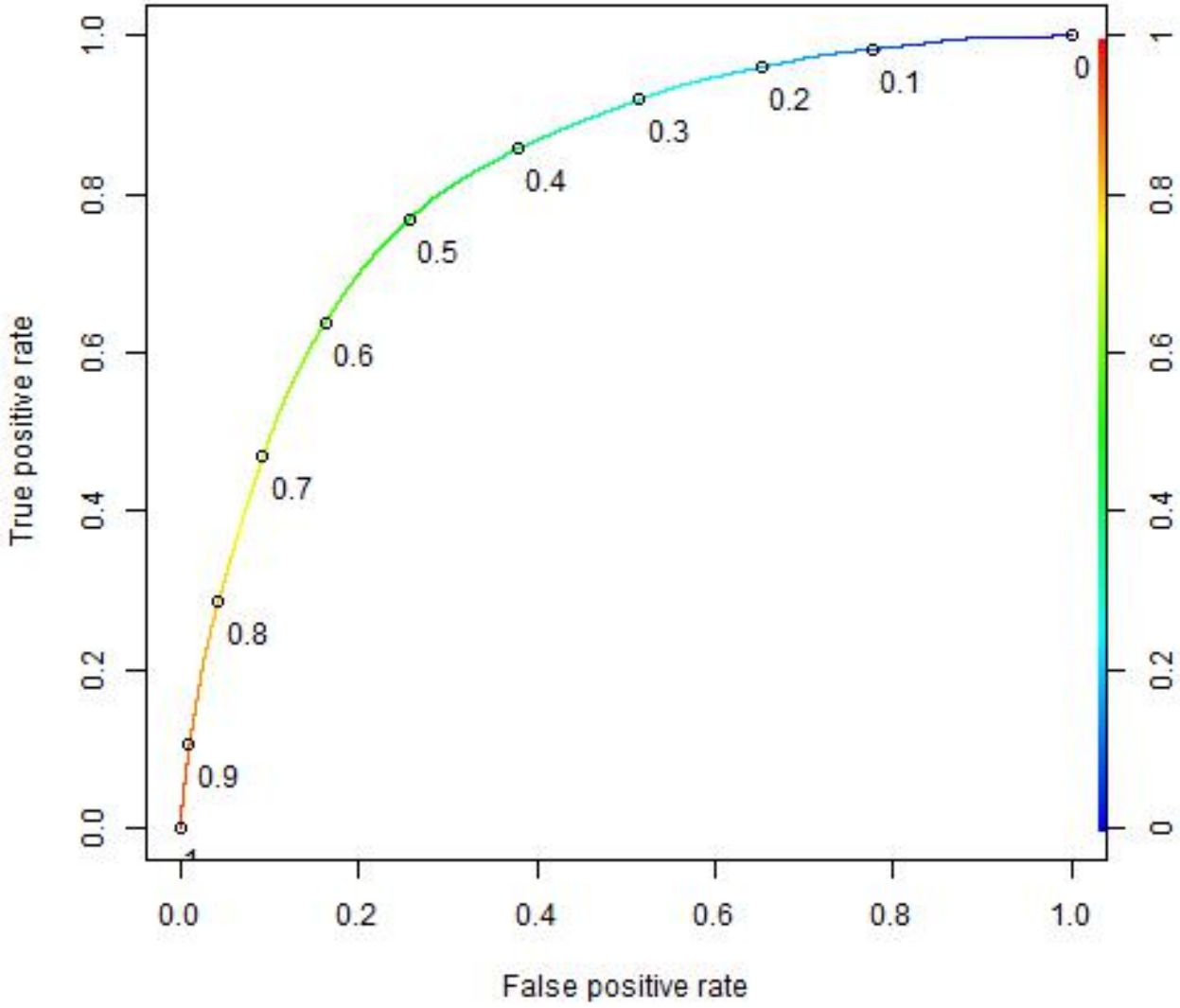

Figure 1: Lodgepole Pine ROC for All Aggregated Data

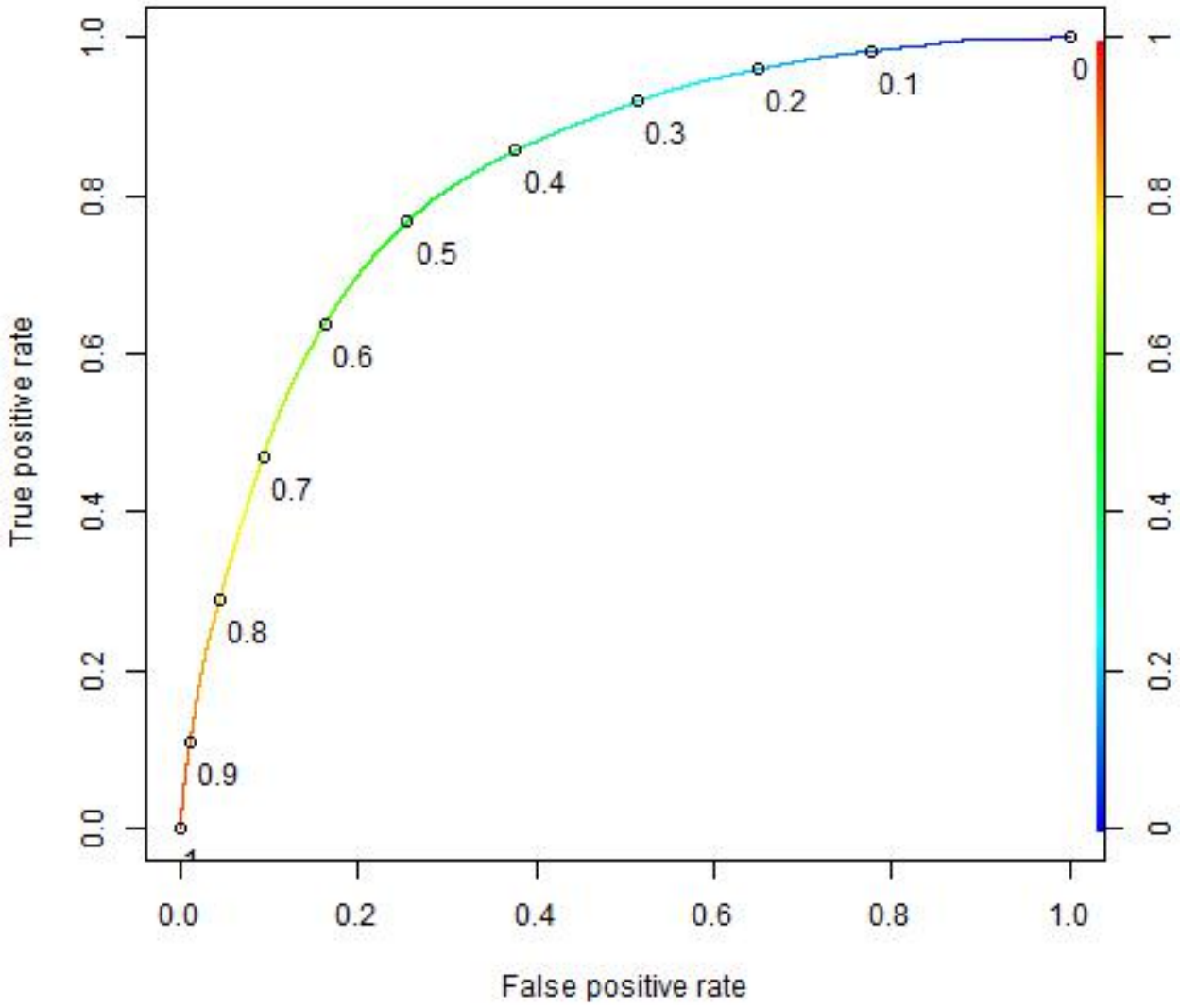

Figure 2: Lodgepole Pine ROC for All Individuated Data

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

```
## [1] "ROCR graph 2 completed at 2018-08-12 19:15:35"
```

Calculate Accuracy of Lodgepole Pine Logisitic Models - All Vars

Calculate Lodgepole Pine Aggregated Data Logisitic Model Accuracy - All Vars

Find best threshold for Lodgepole Pine using all aggregated data.

```
result = calcLogisticModelAccuracy (forestTrain$LodgepolePine, Lodge_Agg_Train_predict,
                        0.0, 1, 10, "Lodgepole", "Other", 1,1)
```

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
## [1] "Thresh=0, Accuracy=48.7%, BaseAcc(Other)=51.2%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=59.4%, BaseAcc(Other)=51.2%, Sens=98.3%, Spec=22.3%, Sens^2+Spec^2=1.016"
## [1] "Thresh=0.2, Accuracy=64.6%, BaseAcc(Other)=51.2%, Sens=96%, Spec=34.8%, Sens^2+Spec^2=1.043"
## [1] "Thresh=0.3, Accuracy=69.7%, BaseAcc(Other)=51.2%, Sens=91.9%, Spec=48.5%, Sens^2+Spec^2=1.081"
## [1] "Thresh=0.4, Accuracy=73.6%, BaseAcc(Other)=51.2%, Sens=85.7%, Spec=62.2%, Sens^2+Spec^2=1.121"
## [1] "Thresh=0.5, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=76.7%, Spec=74.4%, Sens^2+Spec^2=1.143"
## [1] "Thresh=0.6, Accuracy=74%, BaseAcc(Other)=51.2%, Sens=63.8%, Spec=83.8%, Sens^2+Spec^2=1.109"
## [1] "Thresh=0.7, Accuracy=69.4%, BaseAcc(Other)=51.2%, Sens=46.9%, Spec=90.7%, Sens^2+Spec^2=1.044"
## [1] "Thresh=0.8, Accuracy=63%, BaseAcc(Other)=51.2%, Sens=28.7%, Spec=95.7%, Sens^2+Spec^2=0.999"
## [1] "Thresh=0.9, Accuracy=55.9%, BaseAcc(Other)=51.2%, Sens=10.5%, Spec=99%, Sens^2+Spec^2=0.992"
## [1] "Thresh=1, Accuracy=51.2%, BaseAcc(Other)=51.2%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Best Sensitivity_Specificity threshold= 0.5 inc= 0.1"
## [1] "========================================="
## [1] "start= 0.4 end= 0.6 inc= 0.01"
## [1] "Thresh=0.4, Accuracy=73.6%, BaseAcc(Other)=51.2%, Sens=85.7%, Spec=62.2%, Sens^2+Spec^2=1.121"
## [1] "Thresh=0.41, Accuracy=73.9%, BaseAcc(Other)=51.2%, Sens=84.9%, Spec=63.5%, Sens^2+Spec^2=1.125"
## [1] "Thresh=0.42, Accuracy=74.2%, BaseAcc(Other)=51.2%, Sens=84.1%, Spec=64.8%, Sens^2+Spec^2=1.129"
## [1] "Thresh=0.43, Accuracy=74.5%, BaseAcc(Other)=51.2%, Sens=83.3%, Spec=66.1%, Sens^2+Spec^2=1.133"
## [1] "Thresh=0.44, Accuracy=74.8%, BaseAcc(Other)=51.2%, Sens=82.5%, Spec=67.4%, Sens^2+Spec^2=1.136"
## [1] "Thresh=0.45, Accuracy=75%, BaseAcc(Other)=51.2%, Sens=81.7%, Spec=68.6%, Sens^2+Spec^2=1.139"
## [1] "Thresh=0.46, Accuracy=75.2%, BaseAcc(Other)=51.2%, Sens=80.8%, Spec=69.8%, Sens^2+Spec^2=1.141"
## [1] "Thresh=0.47, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=79.9%, Spec=71.1%, Sens^2+Spec^2=1.144"
## [1] "Thresh=0.48, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.9%, Spec=72.2%, Sens^2+Spec^2=1.145"
## [1] "Thresh=0.49, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=77.8%, Spec=73.3%, Sens^2+Spec^2=1.144"
## [1] "Thresh=0.5, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=76.7%, Spec=74.4%, Sens^2+Spec^2=1.143"
## [1] "Thresh=0.51, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=75.6%, Spec=75.5%, Sens^2+Spec^2=1.141"
## [1] "Thresh=0.52, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=74.4%, Spec=76.5%, Sens^2+Spec^2=1.139"
## [1] "Thresh=0.53, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=73.2%, Spec=77.5%, Sens^2+Spec^2=1.138"
## [1] "Thresh=0.54, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=72%, Spec=78.5%, Sens^2+Spec^2=1.135"
## [1] "Thresh=0.55, Accuracy=75.2%, BaseAcc(Other)=51.2%, Sens=70.7%, Spec=79.5%, Sens^2+Spec^2=1.133"
## [1] "Thresh=0.56, Accuracy=75%, BaseAcc(Other)=51.2%, Sens=69.4%, Spec=80.3%, Sens^2+Spec^2=1.129"
## [1] "Thresh=0.57, Accuracy=74.8%, BaseAcc(Other)=51.2%, Sens=68.1%, Spec=81.2%, Sens^2+Spec^2=1.124"
## [1] "Thresh=0.58, Accuracy=74.6%, BaseAcc(Other)=51.2%, Sens=66.7%, Spec=82.1%, Sens^2+Spec^2=1.12"
## [1] "Thresh=0.59, Accuracy=74.3%, BaseAcc(Other)=51.2%, Sens=65.2%, Spec=82.9%, Sens^2+Spec^2=1.114"
## [1] "Best Sensitivity_Specificity threshold= 0.48 inc= 0.01"
## [1] "========================================="
## [1] "start= 0.47 end= 0.49 inc= 0.001"
## [1] "Thresh=0.47, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=79.9%, Spec=71.1%, Sens^2+Spec^2=1.144"
```

```
## [1] "Thresh=0.471, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.8%, Spec=71.2%, Sens^2+Spec^2=1.144
## [1] "Thresh=0.472, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.7%, Spec=71.3%, Sens^2+Spec^2=1.144
## [1] "Thresh=0.473, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.6%, Spec=71.4%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.474, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.5%, Spec=71.6%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.475, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.4%, Spec=71.7%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.476, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=79.3%, Spec=71.8%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.477, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.2%, Spec=71.9%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.478, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=79.1%, Spec=72%, Sens^2+Spec^2=1.145"
## [1] "Thresh=0.479, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=79%, Spec=72.1%, Sens^2+Spec^2=1.145"
## [1] "Thresh=0.48, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.9%, Spec=72.2%, Sens^2+Spec^2=1.145"
## [1] "Thresh=0.481, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.8%, Spec=72.3%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.482, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.6%, Spec=72.5%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.483, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.5%, Spec=72.6%, Sens^2+Spec^2=1.144
## [1] "Thresh=0.484, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.4%, Spec=72.7%, Sens^2+Spec^2=1.144
## [1] "Thresh=0.485, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.3%, Spec=72.8%, Sens^2+Spec^2=1.144
## [1] "Thresh=0.486, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.2%, Spec=72.9%, Sens^2+Spec^2=1.144
## [1] "Thresh=0.487, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.1%, Spec=73%, Sens^2+Spec^2=1.144"
## [1] "Thresh=0.488, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78%, Spec=73.1%, Sens^2+Spec^2=1.144"
## [1] "Thresh=0.489, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=77.9%, Spec=73.2%, Sens^2+Spec^2=1.144
## [1] "Thresh=0.49, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=77.8%, Spec=73.3%, Sens^2+Spec^2=1.144"
## [1] "======================================="
## [1] "Best Threshold=0.476"
## [1] "Best Sensitivity_Specificity=1.14567294963844"
```

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

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

```
result = calcLogisticModelAccuracy (forestTrain$LodgepolePine, Lodge_Agg_Train_predict,
                        curThresh, curThresh, 1, "Lodgepole", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.476"
## [1] "predicted performance="
##                       Predicted
## Actual                 FALSE=Predict:Other TRUE=Predict:Lodgepole
##   0=Actual:Other          149739 (TN)          58659 (FP)
##   1=Actual:Lodgepole      40982 (FN)           157329 (TP)
## [1] "Sensitivity= 0.793344796809052 (True positive rate of Lodgepole = TP/(TP+FN) = 157329 /( 157329
## [1] "Specificity= 0.718524170097602 (True negative rate of Other = TN/(TN+FP) = 149739 /( 149739 + 58
## [1] "Sens^2+Spec^2=1.145"
## [1] "Baseline (Other) Accuracy=0.5124"
## [1] "Logistic Accuracy=0.755006"
```

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

```
result = calcLogisticModelAccuracy (forestTest$LodgepolePine, Lodge_Agg_Test_predict,
                        curThresh, curThresh, 1, "Lodgepole", "Other", 3,
                        saveFile=saveFileName, desc="Lodgepole All Aggregate Vars",
                        AIC=Lodge_Agg_All_aic, AUC=Lodge_Agg_All_ROC_AUC)
```

```
## [1] "Model Performance for threshold= 0.476"
## [1] "predicted performance="
##                       Predicted
## Actual                 FALSE=Predict:Other TRUE=Predict:Lodgepole
```

```
##   0=Actual:Other         64287 (TN)          25026 (FP)
##   1=Actual:Lodgepole     17512 (FN)          67478 (TP)
## [1] "Sensitivity= 0.793952229674079 (True positive rate of Lodgepole = TP/(TP+FN) = 67478 /( 67478 +
## [1] "Specificity= 0.719794430821941 (True negative rate of Other = TN/(TN+FP) = 64287 /( 64287 + 250
## [1] "Sens^2+Spec^2=1.148"
## [1] "Baseline (Other) Accuracy=0.5124"
## [1] "Logistic Accuracy=0.755953"
```

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

Calculate Lodgepole Pine Individuated Data Logisitic Model Accuracy - All Vars

Find best threshold for Lodgepole Pine using all individuated data.

```
result = calcLogisticModelAccuracy (forestTrain$LodgepolePine, Lodge_Ind_Train_predict,
                        0.0, 1, 10, "Lodgepole", "Other", 1,1)
```

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
## [1] "Thresh=0, Accuracy=48.7%, BaseAcc(Other)=51.2%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=59.4%, BaseAcc(Other)=51.2%, Sens=98.3%, Spec=22.3%, Sens^2+Spec^2=1.016"
## [1] "Thresh=0.2, Accuracy=64.7%, BaseAcc(Other)=51.2%, Sens=95.9%, Spec=35%, Sens^2+Spec^2=1.043"
## [1] "Thresh=0.3, Accuracy=69.7%, BaseAcc(Other)=51.2%, Sens=91.9%, Spec=48.6%, Sens^2+Spec^2=1.081"
## [1] "Thresh=0.4, Accuracy=73.7%, BaseAcc(Other)=51.2%, Sens=85.7%, Spec=62.3%, Sens^2+Spec^2=1.122"
## [1] "Thresh=0.5, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=76.7%, Spec=74.4%, Sens^2+Spec^2=1.143"
## [1] "Thresh=0.6, Accuracy=74%, BaseAcc(Other)=51.2%, Sens=63.8%, Spec=83.8%, Sens^2+Spec^2=1.109"
## [1] "Thresh=0.7, Accuracy=69.4%, BaseAcc(Other)=51.2%, Sens=47%, Spec=90.7%, Sens^2+Spec^2=1.044"
## [1] "Thresh=0.8, Accuracy=63.1%, BaseAcc(Other)=51.2%, Sens=28.9%, Spec=95.6%, Sens^2+Spec^2=0.998"
## [1] "Thresh=0.9, Accuracy=56%, BaseAcc(Other)=51.2%, Sens=10.9%, Spec=98.9%, Sens^2+Spec^2=0.991"
## [1] "Thresh=1, Accuracy=51.2%, BaseAcc(Other)=51.2%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Best Sensitivity_Specificity threshold= 0.5 inc= 0.1"
## [1] "========================================"
## [1] "start= 0.4 end= 0.6 inc= 0.01"
## [1] "Thresh=0.4, Accuracy=73.7%, BaseAcc(Other)=51.2%, Sens=85.7%, Spec=62.3%, Sens^2+Spec^2=1.122"
## [1] "Thresh=0.41, Accuracy=74%, BaseAcc(Other)=51.2%, Sens=84.9%, Spec=63.6%, Sens^2+Spec^2=1.126"
## [1] "Thresh=0.42, Accuracy=74.3%, BaseAcc(Other)=51.2%, Sens=84.1%, Spec=64.9%, Sens^2+Spec^2=1.13"
## [1] "Thresh=0.43, Accuracy=74.6%, BaseAcc(Other)=51.2%, Sens=83.3%, Spec=66.2%, Sens^2+Spec^2=1.134"
## [1] "Thresh=0.44, Accuracy=74.8%, BaseAcc(Other)=51.2%, Sens=82.5%, Spec=67.5%, Sens^2+Spec^2=1.137"
## [1] "Thresh=0.45, Accuracy=75%, BaseAcc(Other)=51.2%, Sens=81.7%, Spec=68.7%, Sens^2+Spec^2=1.14"
## [1] "Thresh=0.46, Accuracy=75.2%, BaseAcc(Other)=51.2%, Sens=80.7%, Spec=69.9%, Sens^2+Spec^2=1.142"
## [1] "Thresh=0.47, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.8%, Spec=71.1%, Sens^2+Spec^2=1.144"
## [1] "Thresh=0.48, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.8%, Spec=72.3%, Sens^2+Spec^2=1.145"
## [1] "Thresh=0.49, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=77.8%, Spec=73.4%, Sens^2+Spec^2=1.145"
## [1] "Thresh=0.5, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=76.7%, Spec=74.4%, Sens^2+Spec^2=1.143"
```

```
## [1] "Thresh=0.51, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=75.5%, Spec=75.5%, Sens^2+Spec^2=1.141"
## [1] "Thresh=0.52, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=74.3%, Spec=76.5%, Sens^2+Spec^2=1.139"
## [1] "Thresh=0.53, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=73.2%, Spec=77.5%, Sens^2+Spec^2=1.137"
## [1] "Thresh=0.54, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=72%, Spec=78.5%, Sens^2+Spec^2=1.135"
## [1] "Thresh=0.55, Accuracy=75.2%, BaseAcc(Other)=51.2%, Sens=70.7%, Spec=79.5%, Sens^2+Spec^2=1.132"
## [1] "Thresh=0.56, Accuracy=75%, BaseAcc(Other)=51.2%, Sens=69.4%, Spec=80.3%, Sens^2+Spec^2=1.128"
## [1] "Thresh=0.57, Accuracy=74.8%, BaseAcc(Other)=51.2%, Sens=68.1%, Spec=81.2%, Sens^2+Spec^2=1.125"
## [1] "Thresh=0.58, Accuracy=74.6%, BaseAcc(Other)=51.2%, Sens=66.7%, Spec=82.1%, Sens^2+Spec^2=1.119"
## [1] "Thresh=0.59, Accuracy=74.3%, BaseAcc(Other)=51.2%, Sens=65.2%, Spec=82.9%, Sens^2+Spec^2=1.114"
## [1] "Best Sensitivity_Specificity threshold= 0.48 inc= 0.01"
## [1] "======================================="
## [1] "start= 0.47 end= 0.49 inc= 0.001"
## [1] "Thresh=0.47, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.8%, Spec=71.1%, Sens^2+Spec^2=1.144"
## [1] "Thresh=0.471, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.7%, Spec=71.3%, Sens^2+Spec^2=1.144
## [1] "Thresh=0.472, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.6%, Spec=71.4%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.473, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.5%, Spec=71.5%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.474, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.4%, Spec=71.6%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.475, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.3%, Spec=71.7%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.476, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.2%, Spec=71.8%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.477, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=79.1%, Spec=72%, Sens^2+Spec^2=1.145"
## [1] "Thresh=0.478, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=79%, Spec=72.1%, Sens^2+Spec^2=1.145"
## [1] "Thresh=0.479, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.9%, Spec=72.2%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.48, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.8%, Spec=72.3%, Sens^2+Spec^2=1.145"
## [1] "Thresh=0.481, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.7%, Spec=72.4%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.482, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.6%, Spec=72.5%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.483, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.5%, Spec=72.6%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.484, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.4%, Spec=72.7%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.485, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.3%, Spec=72.9%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.486, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.2%, Spec=73%, Sens^2+Spec^2=1.145"
## [1] "Thresh=0.487, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78.1%, Spec=73.1%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.488, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=78%, Spec=73.2%, Sens^2+Spec^2=1.145"
## [1] "Thresh=0.489, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=77.9%, Spec=73.3%, Sens^2+Spec^2=1.145
## [1] "Thresh=0.49, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=77.8%, Spec=73.4%, Sens^2+Spec^2=1.145"
## [1] "======================================="
## [1] "Best Threshold=0.481"
## [1] "Best Sensitivity_Specificity=1.14556823166058"
```

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

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

```
result = calcLogisticModelAccuracy (forestTrain$LodgepolePine, Lodge_Ind_Train_predict,
                        curThresh, curThresh, 1, "Lodgepole", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.481"
## [1] "predicted performance="
##                       Predicted
## Actual                 FALSE=Predict:Other TRUE=Predict:Lodgepole
##   0=Actual:Other          151027 (TN)          57371 (FP)
##   1=Actual:Lodgepole      42114 (FN)           156197 (TP)
## [1] "Sensitivity= 0.787636591011089 (True positive rate of Lodgepole = TP/(TP+FN) = 156197 /( 156197
## [1] "Specificity= 0.72470465167612 (True negative rate of Other = TN/(TN+FP) = 151027 /( 151027 + 573
## [1] "Sens^2+Spec^2=1.145"
```

```
## [1] "Baseline (Other) Accuracy=0.5124"
## [1] "Logistic Accuracy=0.75539"
```

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

```
result = calcLogisticModelAccuracy (forestTest$LodgepolePine, Lodge_Ind_Test_predict,
                       curThresh, curThresh, 1, "Lodgepole", "Other", 3,
                       saveFile=saveFileName, desc="Lodgepole All Individualized Vars",
                       AIC=Lodge_Ind_All_aic, AUC=Lodge_Ind_All_ROC_AUC)
```

```
## [1] "Model Performance for threshold= 0.481"
## [1] "predicted performance="
##                       Predicted
## Actual                 FALSE=Predict:Other TRUE=Predict:Lodgepole
##   0=Actual:Other          64882 (TN)          24431 (FP)
##   1=Actual:Lodgepole      17944 (FN)          67046 (TP)
## [1] "Sensitivity= 0.788869278738675 (True positive rate of Lodgepole = TP/(TP+FN) = 67046 /( 67046 +
## [1] "Specificity= 0.726456394925711 (True negative rate of Other = TN/(TN+FP) = 64882 /( 64882 + 244
## [1] "Sens^2+Spec^2=1.15"
## [1] "Baseline (Other) Accuracy=0.5124"
## [1] "Logistic Accuracy=0.756888"
```

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

The Lodgepole Pine 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.

Lodgepole Pine Logistic Regression - Significant Variables

Create Lodgepole Pine 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.

Lodgepole Pine Logistic Model using Significant Aggregated Data

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

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

  # save model for later use
  Lodge_Agg_Sig_LogMod = Lodge_Agg_LogMod
  save("Lodge_Agg_Sig_LogMod", file="Lodge_Agg_Sig_LogMod.Rdata")

  Lodge_Agg_Sig_aic<-as.integer(Lodge_Agg_LogMod$aic)
  Lodge_Agg_Sig_aic
```

```
## [1] 419472
```

Check the coefficients of the Lodgepole Pine model using significant aggregated data.

```
summary(Lodge_Agg_LogMod)
```

```
##
## Call:
## glm(formula = LodgepolePine ~ Elev + Aspect + Slope + H2OHD +
##     H2OVD + RoadHD + FirePtHD + Shade12PM + Shade3PM + RWwild +
##     NEwild + CMwild + ST02 + ST03 + ST04 + ST06 + ST08 + ST09 +
##     ST10 + ST11 + ST12 + ST13 + ST16 + ST18 + ST19 + ST20 + ST21 +
##     ST22 + ST23 + ST24 + ST25 + ST26 + ST27 + ST28 + ST29 + ST30 +
##     ST31 + ST32 + ST33 + ST34 + ST35 + ST36, family = binomial,
##     data = forestTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9057  -0.8850  -0.1397   0.8644   3.6729
##
```

```
## Coefficients:
##               Estimate Std. Error  z value Pr(>|z|)
## (Intercept)  3.097e+00  9.637e-02   32.136  < 2e-16 ***
## Elev        -5.309e-03  2.877e-05 -184.562  < 2e-16 ***
## Aspect      -2.867e-04  4.624e-05   -6.201 5.61e-10 ***
## Slope        1.410e-02  7.705e-04   18.294  < 2e-16 ***
## H2OHD        1.518e-03  2.455e-05   61.826  < 2e-16 ***
## H2OVD        4.717e-04  8.983e-05    5.252 1.51e-07 ***
## RoadHD       7.861e-05  3.098e-06   25.370  < 2e-16 ***
## FirePtHD     1.652e-05  3.418e-06    4.833 1.34e-06 ***
## Shade12PM    2.461e-02  3.249e-04   75.743  < 2e-16 ***
## Shade3PM    -2.869e-03  1.636e-04  -17.540  < 2e-16 ***
## RWwild       4.924e+00  3.940e-02  124.987  < 2e-16 ***
## NEwild       5.376e+00  4.192e-02  128.255  < 2e-16 ***
## CMwild       4.723e+00  3.528e-02  133.879  < 2e-16 ***
## ST02        -1.950e+00  5.470e-02  -35.656  < 2e-16 ***
## ST03        -4.212e-01  5.708e-02   -7.378 1.60e-13 ***
## ST04        -8.647e-01  4.002e-02  -21.604  < 2e-16 ***
## ST06         1.942e+00  5.880e-02   33.036  < 2e-16 ***
## ST08         2.181e+00  1.977e-01   11.030  < 2e-16 ***
## ST09         1.873e+00  1.058e-01   17.708  < 2e-16 ***
## ST10         1.441e+00  3.377e-02   42.669  < 2e-16 ***
## ST11         2.092e+00  3.862e-02   54.174  < 2e-16 ***
## ST12         3.302e+00  3.897e-02   84.745  < 2e-16 ***
## ST13         2.704e+00  3.626e-02   74.569  < 2e-16 ***
## ST16         1.779e+00  6.004e-02   29.630  < 2e-16 ***
## ST18         1.820e+00  9.554e-02   19.052  < 2e-16 ***
## ST19         1.810e+00  4.931e-02   36.712  < 2e-16 ***
## ST20         1.766e+00  3.931e-02   44.936  < 2e-16 ***
## ST21        -1.005e+00  2.732e-01   -3.680 0.000233 ***
## ST22         1.439e+00  3.264e-02   44.077  < 2e-16 ***
## ST23         1.782e+00  3.011e-02   59.176  < 2e-16 ***
## ST24         2.410e+00  3.313e-02   72.741  < 2e-16 ***
## ST25         3.633e+00  1.346e-01   26.982  < 2e-16 ***
## ST26         3.123e+00  7.219e-02   43.266  < 2e-16 ***
## ST27         2.069e+00  8.084e-02   25.596  < 2e-16 ***
## ST28         3.965e+00  1.814e-01   21.856  < 2e-16 ***
## ST29         2.522e+00  3.084e-02   81.779  < 2e-16 ***
## ST30         2.468e+00  3.401e-02   72.555  < 2e-16 ***
## ST31         2.505e+00  3.230e-02   77.549  < 2e-16 ***
## ST32         2.822e+00  3.022e-02   93.365  < 2e-16 ***
## ST33         2.573e+00  3.070e-02   83.820  < 2e-16 ***
## ST34         3.931e+00  1.030e-01   38.159  < 2e-16 ***
## ST35        -1.389e+00  3.800e-01   -3.655 0.000257 ***
## ST36         2.270e+00  2.249e-01   10.095  < 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: 563568  on 406708  degrees of freedom
## Residual deviance: 419386  on 406666  degrees of freedom
## AIC: 419472
##
```

```
## Number of Fisher Scoring iterations: 7
```

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

Lodgepole Pine Logistic Model using Significant Individuated Data

Create a logistic model for the significant individuated variables.

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

```
Lodge_Ind_LogMod =
  glm(LodgepolePine ~
        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:
          # Montane_low + Montane +  #  removed 2nd pass
          SubAlpine + Alpine +
          # Dry + Non_Dry +   # removed 2nd pass
        # Geology Zone:
          Alluvium + Glacial +
          # Sed_mix + Ign_Meta +
        # Soil Family:
          # Aquolis_cmplx +   # removed 2nd pass
          # 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 +  #  removed 2nd pass
            # 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)

  # save model for later use
  Lodge_Ind_Sig_LogMod = Lodge_Ind_LogMod
  save("Lodge_Ind_Sig_LogMod", file="Lodge_Ind_Sig_LogMod.Rdata")

  Lodge_Ind_Sig_aic<-as.integer(Lodge_Ind_LogMod$aic)
  Lodge_Ind_Sig_aic
```

```
## [1] 438293
```

```
  summary(Lodge_Ind_LogMod)
```

```
##
## Call:
## glm(formula = LodgepolePine ~ Elev + Aspect + Slope + H2OHD +
##     H2OVD + RoadHD + FirePtHD + Shade12PM + Shade3PM + RWwild +
##     NEwild + CMwild + SubAlpine + Alpine + Alluvium + Glacial +
##     Borohemists_cmplx + Bross + Bullwark + Catamount_cmplx +
##     Cryaquepts_Typic + Cryaquolls + Cryaquolls_cmplx + Cryaquolls_Typic_cmplx +
##     Cryorthents + Granile + Rock_Land + Rock_Land_cmplx, family = binomial,
##     data = forestTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8705  -0.9464  -0.1260   0.9295   3.2157
##
## Coefficients:
##                          Estimate Std. Error  z value Pr(>|z|)
## (Intercept)             3.102e+00  8.628e-02   35.954  < 2e-16 ***
## Elev                   -4.362e-03  2.755e-05 -158.365  < 2e-16 ***
## Aspect                 -3.301e-04  4.489e-05   -7.353 1.94e-13 ***
## Slope                   6.554e-03  7.043e-04    9.305  < 2e-16 ***
## H2OHD                   1.395e-03  2.376e-05   58.685  < 2e-16 ***
## H2OVD                   3.602e-04  8.604e-05    4.186 2.84e-05 ***
## RoadHD                  6.468e-05  3.037e-06   21.297  < 2e-16 ***
## FirePtHD                3.638e-05  3.295e-06   11.042  < 2e-16 ***
## Shade12PM               1.784e-02  3.005e-04   59.356  < 2e-16 ***
## Shade3PM               -4.397e-04  1.545e-04   -2.845 0.004441 **
## RWwild                  4.207e+00  3.045e-02  138.141  < 2e-16 ***
## NEwild                  4.576e+00  3.565e-02  128.329  < 2e-16 ***
## CMwild                  4.077e+00  2.861e-02  142.531  < 2e-16 ***
## SubAlpine               1.611e+00  1.759e-02   91.593  < 2e-16 ***
```

```
## Alpine                -1.834e+00  9.785e-02  -18.746  < 2e-16 ***
## Alluvium              -3.456e+00  1.047e-01  -33.018  < 2e-16 ***
## Glacial               -1.002e+00  1.662e-02  -60.268  < 2e-16 ***
## Borohemists_cmplx      2.762e+00  1.121e-01   24.645  < 2e-16 ***
## Bross                  2.234e+00  2.430e-01    9.191  < 2e-16 ***
## Bullwark               3.751e-01  2.012e-02   18.638  < 2e-16 ***
## Catamount_cmplx       -6.020e-02  1.717e-02   -3.505 0.000456 ***
## Cryaquepts_Typic       2.559e+00  1.094e-01   23.387  < 2e-16 ***
## Cryaquolls             2.900e+00  1.162e-01   24.962  < 2e-16 ***
## Cryaquolls_cmplx       1.229e+00  9.962e-02   12.342  < 2e-16 ***
## Cryaquolls_Typic_cmplx 2.794e-01  1.897e-02   14.732  < 2e-16 ***
## Cryorthents           -3.036e-01  9.258e-02   -3.280 0.001039 **
## Granile                8.062e-01  6.796e-02   11.864  < 2e-16 ***
## Rock_Land              8.842e-01  1.554e-02   56.897  < 2e-16 ***
## Rock_Land_cmplx        2.002e+00  2.229e-02   89.795  < 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: 563568  on 406708  degrees of freedom
## Residual deviance: 438235  on 406680  degrees of freedom
## AIC: 438293
##
## Number of Fisher Scoring iterations: 6
```

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

Predict Lodgepole Pine Logistic Model Probabilities - Sig Vars

Lodgepole Pine Probabilities using Significant Aggregated Data

Predict the probability of Lodgepole Pine for aggregated Data - significant variables.

```
# Predict Lodgepole Pine Agg Data - significant variables

  Lodge_Agg_Train_predict= predict(Lodge_Agg_LogMod, type="response")
  summary(Lodge_Agg_Train_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001177 0.262623 0.507043 0.487599 0.720227 0.989996
```

```
  Lodge_Agg_Test_predict= predict(Lodge_Agg_LogMod, type="response",newdata=forestTest)
  summary(Lodge_Agg_Test_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001063 0.263705 0.506269 0.487505 0.720219 0.989542
```

Lodgepole Pine Probabilities using Significant Individuated Data

Predict the probability of Lodgepole Pine using significant Individuated Data.

```
  Lodge_Ind_Train_predict= predict(Lodge_Ind_LogMod, type="response")
  summary(Lodge_Ind_Train_predict)
```

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

```
## 0.0006416 0.3008665 0.5067524 0.4875992 0.6901729 0.9837541
```

```
  Lodge_Ind_Test_predict= predict(Lodge_Ind_LogMod, type="response",newdata=forestTest)
  summary(Lodge_Ind_Test_predict)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0006601 0.3014278 0.5062873 0.4876456 0.6904649 0.9832762
```

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

```
## [1] "ROCR graph 2 completed at 2018-08-12 19:15:35"
```

Lodgepole Pine Receiver Operating Characteristic (ROC) - Sig Vars

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

```
  if (calcROC) {
    ROCpred_Lodge_Agg = prediction(Lodge_Agg_Train_predict, forestTrain$LodgepolePine)
    summary(ROCpred_Lodge_Agg)

    ROCperf_Lodge_Agg = performance(ROCpred_Lodge_Agg, "tpr", "fpr")
    summary(ROCperf_Lodge_Agg)

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

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

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

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

    ROCpred_Lodge_Ind = prediction(Lodge_Ind_Train_predict, forestTrain$LodgepolePine)
    summary(ROCpred_Lodge_Ind)

    ROCperf_Lodge_Ind = performance(ROCpred_Lodge_Ind, "tpr", "fpr")
    summary(ROCperf_Lodge_Ind)

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

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

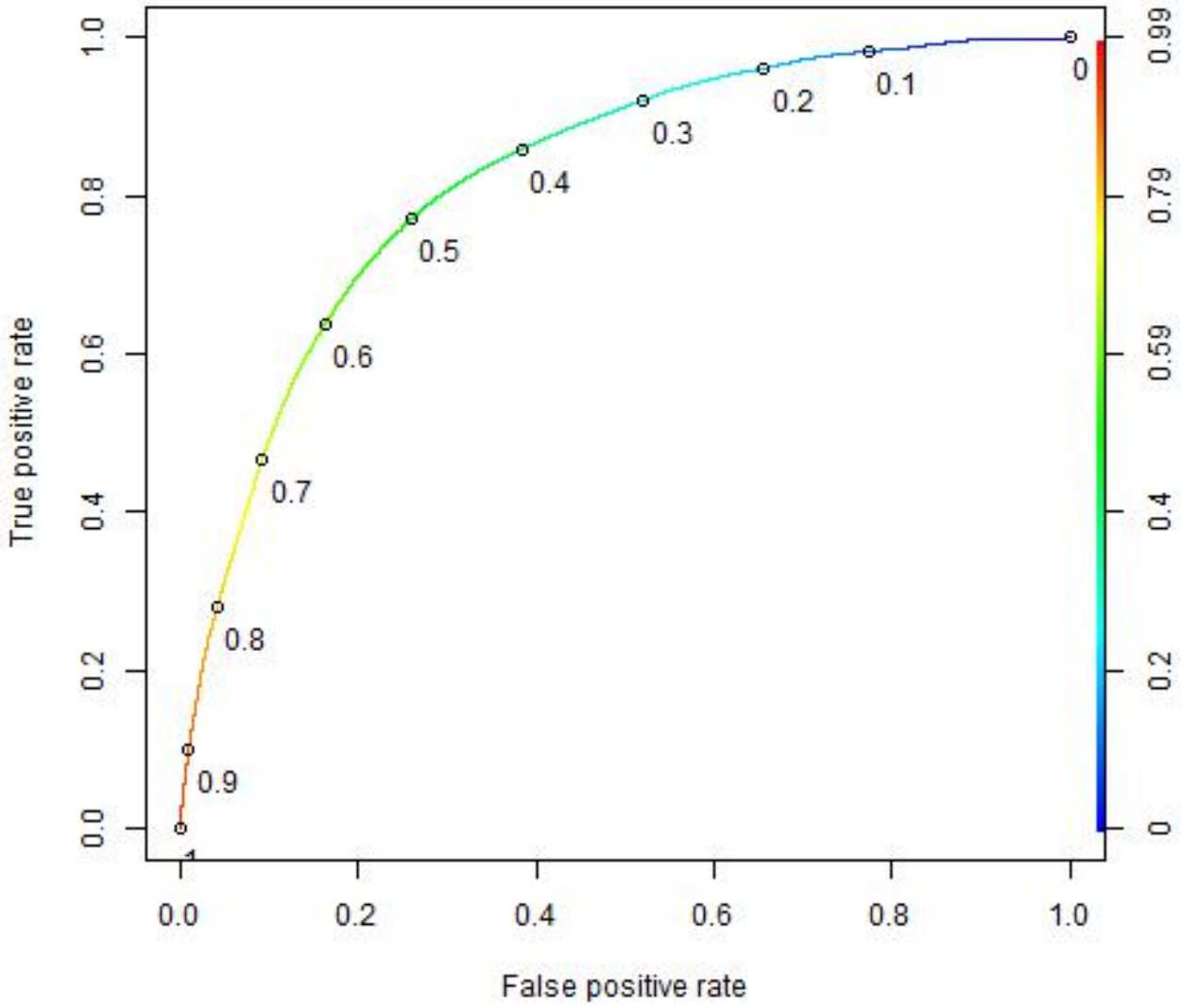

Figure 3: Lodgepole Pine ROC for Aggregated Significant Data

```
    Lodge_Ind_Sig_ROC_AUC = 83.8
  }
```

```
## [1] "ROCR graph 2 started at 2018-08-12 19:20:56"
```

```
## 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 Lodgepole Pine Logisitic Model - Sig Vars

Calculate Lodgepole Pine Aggregated Data Logisitic Model Accuracy - Significant Vars

Find best Lodgepole Pine threshold for Aggregated Data using significant variables.

```
result = calcLogisticModelAccuracy (forestTrain$LodgepolePine, Lodge_Agg_Train_predict,
                        0.0, 1, 10, "Lodgepole", "Other", 1,1)
```

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
## [1] "Thresh=0, Accuracy=48.7%, BaseAcc(Other)=51.2%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=59.5%, BaseAcc(Other)=51.2%, Sens=98.2%, Spec=22.7%, Sens^2+Spec^2=1.017"
## [1] "Thresh=0.2, Accuracy=64.5%, BaseAcc(Other)=51.2%, Sens=96.1%, Spec=34.4%, Sens^2+Spec^2=1.043"
## [1] "Thresh=0.3, Accuracy=69.5%, BaseAcc(Other)=51.2%, Sens=92.1%, Spec=47.9%, Sens^2+Spec^2=1.078"
## [1] "Thresh=0.4, Accuracy=73.4%, BaseAcc(Other)=51.2%, Sens=85.9%, Spec=61.6%, Sens^2+Spec^2=1.118"
## [1] "Thresh=0.5, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=77%, Spec=74.1%, Sens^2+Spec^2=1.142"
## [1] "Thresh=0.6, Accuracy=74%, BaseAcc(Other)=51.2%, Sens=63.8%, Spec=83.7%, Sens^2+Spec^2=1.109"
## [1] "Thresh=0.7, Accuracy=69.2%, BaseAcc(Other)=51.2%, Sens=46.5%, Spec=90.8%, Sens^2+Spec^2=1.041"
## [1] "Thresh=0.8, Accuracy=62.7%, BaseAcc(Other)=51.2%, Sens=27.9%, Spec=95.8%, Sens^2+Spec^2=0.997"
## [1] "Thresh=0.9, Accuracy=55.6%, BaseAcc(Other)=51.2%, Sens=10.1%, Spec=99%, Sens^2+Spec^2=0.991"
## [1] "Thresh=1, Accuracy=51.2%, BaseAcc(Other)=51.2%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Best Sensitivity_Specificity threshold= 0.5 inc= 0.1"
## [1] "======================================="
## [1] "start= 0.4 end= 0.6 inc= 0.01"
## [1] "Thresh=0.4, Accuracy=73.4%, BaseAcc(Other)=51.2%, Sens=85.9%, Spec=61.6%, Sens^2+Spec^2=1.118"
## [1] "Thresh=0.41, Accuracy=73.8%, BaseAcc(Other)=51.2%, Sens=85.2%, Spec=62.9%, Sens^2+Spec^2=1.122"
## [1] "Thresh=0.42, Accuracy=74.1%, BaseAcc(Other)=51.2%, Sens=84.4%, Spec=64.3%, Sens^2+Spec^2=1.126"
## [1] "Thresh=0.43, Accuracy=74.4%, BaseAcc(Other)=51.2%, Sens=83.6%, Spec=65.6%, Sens^2+Spec^2=1.131"
## [1] "Thresh=0.44, Accuracy=74.7%, BaseAcc(Other)=51.2%, Sens=82.8%, Spec=66.9%, Sens^2+Spec^2=1.135"
## [1] "Thresh=0.45, Accuracy=74.9%, BaseAcc(Other)=51.2%, Sens=82%, Spec=68.1%, Sens^2+Spec^2=1.137"
## [1] "Thresh=0.46, Accuracy=75.1%, BaseAcc(Other)=51.2%, Sens=81.1%, Spec=69.4%, Sens^2+Spec^2=1.139"
## [1] "Thresh=0.47, Accuracy=75.2%, BaseAcc(Other)=51.2%, Sens=80.1%, Spec=70.6%, Sens^2+Spec^2=1.141"
## [1] "Thresh=0.48, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.1%, Spec=71.8%, Sens^2+Spec^2=1.143"
## [1] "Thresh=0.49, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=78%, Spec=73%, Sens^2+Spec^2=1.143"
## [1] "Thresh=0.5, Accuracy=75.5%, BaseAcc(Other)=51.2%, Sens=77%, Spec=74.1%, Sens^2+Spec^2=1.142"
## [1] "Thresh=0.51, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=75.7%, Spec=75.2%, Sens^2+Spec^2=1.14"
## [1] "Thresh=0.52, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=74.5%, Spec=76.3%, Sens^2+Spec^2=1.138"
## [1] "Thresh=0.53, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=73.3%, Spec=77.3%, Sens^2+Spec^2=1.136"
## [1] "Thresh=0.54, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=72.1%, Spec=78.3%, Sens^2+Spec^2=1.134"
## [1] "Thresh=0.55, Accuracy=75.2%, BaseAcc(Other)=51.2%, Sens=70.8%, Spec=79.3%, Sens^2+Spec^2=1.131"
## [1] "Thresh=0.56, Accuracy=75%, BaseAcc(Other)=51.2%, Sens=69.5%, Spec=80.2%, Sens^2+Spec^2=1.128"
## [1] "Thresh=0.57, Accuracy=74.8%, BaseAcc(Other)=51.2%, Sens=68.2%, Spec=81.1%, Sens^2+Spec^2=1.124"
```

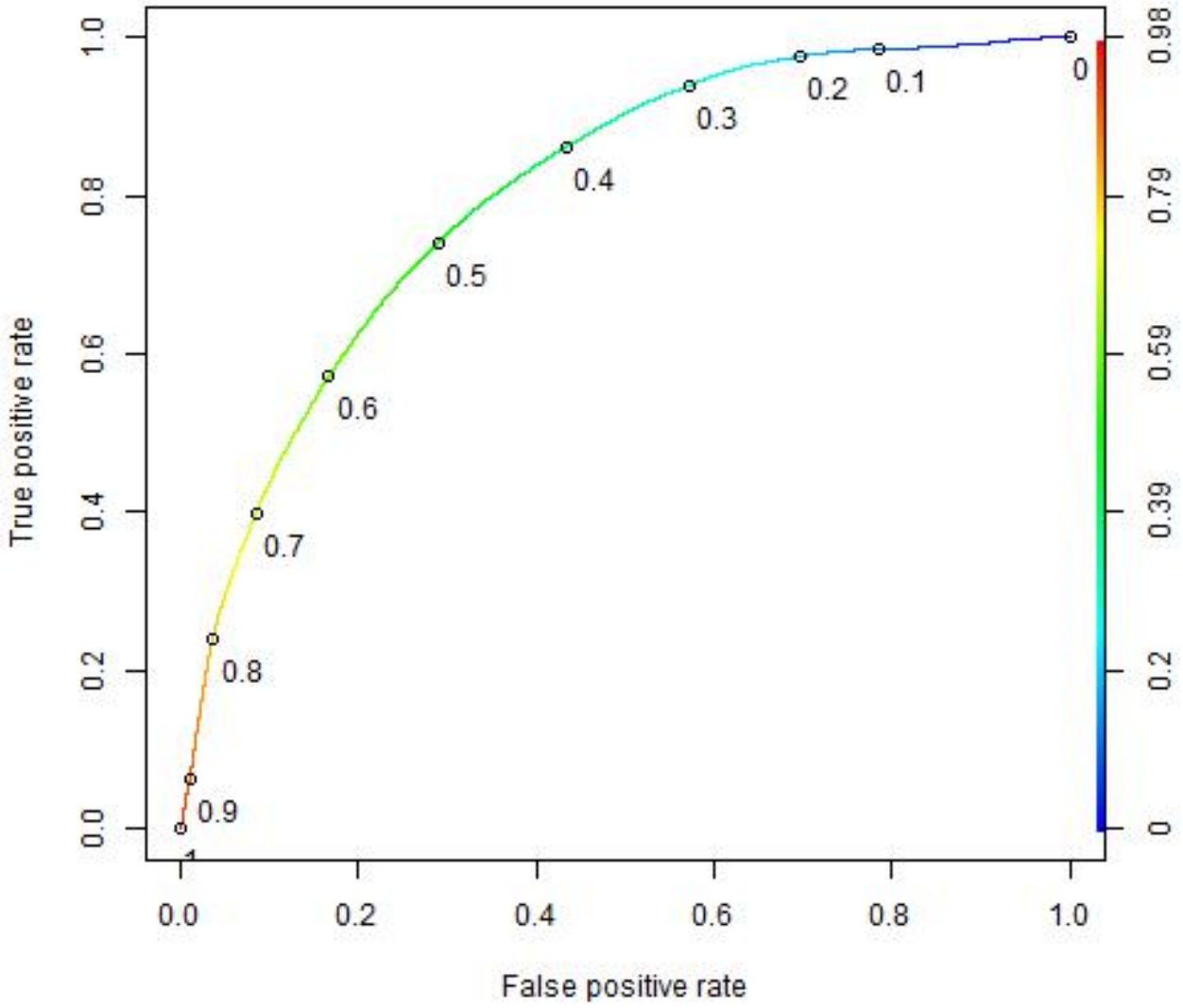

Figure 4: Lodgepole Pine ROC for Individuated Significant Data

```
## [1] "Thresh=0.58, Accuracy=74.5%, BaseAcc(Other)=51.2%, Sens=66.7%, Spec=82%, Sens^2+Spec^2=1.118"
## [1] "Thresh=0.59, Accuracy=74.3%, BaseAcc(Other)=51.2%, Sens=65.2%, Spec=82.8%, Sens^2+Spec^2=1.113"
## [1] "Best Sensitivity_Specificity threshold= 0.48 inc= 0.01"
## [1] "========================================"
## [1] "start= 0.47 end= 0.49 inc= 0.001"
## [1] "Thresh=0.47, Accuracy=75.2%, BaseAcc(Other)=51.2%, Sens=80.1%, Spec=70.6%, Sens^2+Spec^2=1.141"
## [1] "Thresh=0.471, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=80%, Spec=70.7%, Sens^2+Spec^2=1.141"
## [1] "Thresh=0.472, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=79.9%, Spec=70.9%, Sens^2+Spec^2=1.142
## [1] "Thresh=0.473, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=79.8%, Spec=71%, Sens^2+Spec^2=1.142"
## [1] "Thresh=0.474, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=79.7%, Spec=71.1%, Sens^2+Spec^2=1.142
## [1] "Thresh=0.475, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=79.6%, Spec=71.2%, Sens^2+Spec^2=1.142
## [1] "Thresh=0.476, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=79.5%, Spec=71.3%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.477, Accuracy=75.3%, BaseAcc(Other)=51.2%, Sens=79.4%, Spec=71.5%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.478, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.3%, Spec=71.6%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.479, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.2%, Spec=71.7%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.48, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79.1%, Spec=71.8%, Sens^2+Spec^2=1.143"
## [1] "Thresh=0.481, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=79%, Spec=72%, Sens^2+Spec^2=1.143"
## [1] "Thresh=0.482, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=78.9%, Spec=72.1%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.483, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=78.8%, Spec=72.2%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.484, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=78.7%, Spec=72.3%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.485, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=78.6%, Spec=72.4%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.486, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=78.5%, Spec=72.5%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.487, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=78.4%, Spec=72.6%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.488, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=78.3%, Spec=72.8%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.489, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=78.1%, Spec=72.9%, Sens^2+Spec^2=1.143
## [1] "Thresh=0.49, Accuracy=75.4%, BaseAcc(Other)=51.2%, Sens=78%, Spec=73%, Sens^2+Spec^2=1.143"
## [1] "========================================"
## [1] "Best Threshold=0.482"
## [1] "Best Sensitivity_Specificity=1.14365324448173"
```

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

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

```
result = calcLogisticModelAccuracy (forestTrain$LodgepolePine, Lodge_Agg_Train_predict,
                        curThresh, curThresh, 1, "Lodgepole", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.482"
## [1] "predicted performance="
##                       Predicted
## Actual                 FALSE=Predict:Other TRUE=Predict:Lodgepole
##   0=Actual:Other          150312 (TN)          58086 (FP)
##   1=Actual:Lodgepole      41731 (FN)           156580 (TP)
## [1] "Sensitivity= 0.789567900923297 (True positive rate of Lodgepole = TP/(TP+FN) = 156580 /( 156580
## [1] "Specificity= 0.721273716638355 (True negative rate of Other = TN/(TN+FP) = 150312 /( 150312 + 58
## [1] "Sens^2+Spec^2=1.143"
## [1] "Baseline (Other) Accuracy=0.5124"
## [1] "Logistic Accuracy=0.754573"
```

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

```
result = calcLogisticModelAccuracy (forestTest$LodgepolePine, Lodge_Agg_Test_predict,
                        curThresh, curThresh, 1, "Lodgepole", "Other", 3,
```

```
                        saveFile=saveFileName, desc="Lodgepole Sig Aggregate Vars",
                        AIC=Lodge_Agg_Sig_aic, AUC=Lodge_Agg_Sig_ROC_AUC)
```

```
## [1] "Model Performance for threshold= 0.482"
## [1] "predicted performance="
##                       Predicted
## Actual                 FALSE=Predict:Other TRUE=Predict:Lodgepole
##   0=Actual:Other          64522 (TN)          24791 (FP)
##   1=Actual:Lodgepole      17844 (FN)          67146 (TP)
## [1] "Sensitivity= 0.7900458877515 (True positive rate of Lodgepole = TP/(TP+FN) = 67146 /( 67146 + 1
## [1] "Specificity= 0.722425626728472 (True negative rate of Other = TN/(TN+FP) = 64522 /( 64522 + 247
## [1] "Sens^2+Spec^2=1.146"
## [1] "Baseline (Other) Accuracy=0.5124"
## [1] "Logistic Accuracy=0.755397"
```

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

Calculate Lodgepole Pine Individuated Data Logisitic Model Accuracy - Significant Vars

Find best Lodgepole Pine threshold for Inividuated Data using significant variables.

```
result = calcLogisticModelAccuracy (forestTrain$LodgepolePine, Lodge_Ind_Train_predict,
                        0.0, 1, 10, "Lodgepole", "Other", 1,1)
```

```
## [1] "Searching for threshold producing best Sensitivity_Specificity"
## [1] "start= 0 end= 1 inc= 0.1"
## [1] "Thresh=0, Accuracy=48.7%, BaseAcc(Other)=51.2%, Sens=100%, Spec=0%, Sens^2+Spec^2=-2"
## [1] "Thresh=0.1, Accuracy=59%, BaseAcc(Other)=51.2%, Sens=98.5%, Spec=21.4%, Sens^2+Spec^2=1.016"
## [1] "Thresh=0.2, Accuracy=63.1%, BaseAcc(Other)=51.2%, Sens=97.6%, Spec=30.3%, Sens^2+Spec^2=1.045"
## [1] "Thresh=0.3, Accuracy=67.7%, BaseAcc(Other)=51.2%, Sens=93.9%, Spec=42.8%, Sens^2+Spec^2=1.066"
## [1] "Thresh=0.4, Accuracy=71%, BaseAcc(Other)=51.2%, Sens=86.1%, Spec=56.7%, Sens^2+Spec^2=1.064"
## [1] "Thresh=0.5, Accuracy=72.5%, BaseAcc(Other)=51.2%, Sens=74.1%, Spec=70.9%, Sens^2+Spec^2=1.053"
## [1] "Thresh=0.6, Accuracy=70.6%, BaseAcc(Other)=51.2%, Sens=57.2%, Spec=83.3%, Sens^2+Spec^2=1.022"
## [1] "Thresh=0.7, Accuracy=66.3%, BaseAcc(Other)=51.2%, Sens=39.9%, Spec=91.5%, Sens^2+Spec^2=0.996"
## [1] "Thresh=0.8, Accuracy=61.1%, BaseAcc(Other)=51.2%, Sens=24%, Spec=96.3%, Sens^2+Spec^2=0.986"
## [1] "Thresh=0.9, Accuracy=53.7%, BaseAcc(Other)=51.2%, Sens=6.1%, Spec=98.9%, Sens^2+Spec^2=0.983"
## [1] "Thresh=1, Accuracy=51.2%, BaseAcc(Other)=51.2%, Sens=0%, Spec=100%, Sens^2+Spec^2=-2"
## [1] "Best Sensitivity_Specificity threshold= 0.3 inc= 0.1"
## [1] "======================================="
## [1] "start= 0.2 end= 0.4 inc= 0.01"
## [1] "Thresh=0.2, Accuracy=63.1%, BaseAcc(Other)=51.2%, Sens=97.6%, Spec=30.3%, Sens^2+Spec^2=1.045"
## [1] "Thresh=0.21, Accuracy=63.6%, BaseAcc(Other)=51.2%, Sens=97.4%, Spec=31.4%, Sens^2+Spec^2=1.048"
## [1] "Thresh=0.22, Accuracy=64.1%, BaseAcc(Other)=51.2%, Sens=97.1%, Spec=32.6%, Sens^2+Spec^2=1.051"
## [1] "Thresh=0.23, Accuracy=64.6%, BaseAcc(Other)=51.2%, Sens=96.9%, Spec=33.8%, Sens^2+Spec^2=1.054"
## [1] "Thresh=0.24, Accuracy=65.1%, BaseAcc(Other)=51.2%, Sens=96.6%, Spec=35.1%, Sens^2+Spec^2=1.058"
## [1] "Thresh=0.25, Accuracy=65.6%, BaseAcc(Other)=51.2%, Sens=96.3%, Spec=36.4%, Sens^2+Spec^2=1.061"
```

```
## [1] "Thresh=0.26, Accuracy=66.1%, BaseAcc(Other)=51.2%, Sens=95.9%, Spec=37.8%, Sens^2+Spec^2=1.064"
## [1] "Thresh=0.27, Accuracy=66.5%, BaseAcc(Other)=51.2%, Sens=95.5%, Spec=39%, Sens^2+Spec^2=1.065"
## [1] "Thresh=0.28, Accuracy=66.9%, BaseAcc(Other)=51.2%, Sens=95%, Spec=40.2%, Sens^2+Spec^2=1.066"
## [1] "Thresh=0.29, Accuracy=67.3%, BaseAcc(Other)=51.2%, Sens=94.5%, Spec=41.5%, Sens^2+Spec^2=1.066"
## [1] "Thresh=0.3, Accuracy=67.7%, BaseAcc(Other)=51.2%, Sens=93.9%, Spec=42.8%, Sens^2+Spec^2=1.066"
## [1] "Thresh=0.31, Accuracy=68.1%, BaseAcc(Other)=51.2%, Sens=93.3%, Spec=44.1%, Sens^2+Spec^2=1.066"
## [1] "Thresh=0.32, Accuracy=68.5%, BaseAcc(Other)=51.2%, Sens=92.7%, Spec=45.5%, Sens^2+Spec^2=1.068"
## [1] "Thresh=0.33, Accuracy=68.9%, BaseAcc(Other)=51.2%, Sens=92.1%, Spec=46.9%, Sens^2+Spec^2=1.068"
## [1] "Thresh=0.34, Accuracy=69.3%, BaseAcc(Other)=51.2%, Sens=91.3%, Spec=48.3%, Sens^2+Spec^2=1.068"
## [1] "Thresh=0.35, Accuracy=69.6%, BaseAcc(Other)=51.2%, Sens=90.6%, Spec=49.7%, Sens^2+Spec^2=1.068"
## [1] "Thresh=0.36, Accuracy=69.9%, BaseAcc(Other)=51.2%, Sens=89.7%, Spec=51%, Sens^2+Spec^2=1.067"
## [1] "Thresh=0.37, Accuracy=70.2%, BaseAcc(Other)=51.2%, Sens=88.9%, Spec=52.4%, Sens^2+Spec^2=1.066"
## [1] "Thresh=0.38, Accuracy=70.5%, BaseAcc(Other)=51.2%, Sens=88%, Spec=53.7%, Sens^2+Spec^2=1.065"
## [1] "Thresh=0.39, Accuracy=70.7%, BaseAcc(Other)=51.2%, Sens=87.1%, Spec=55.1%, Sens^2+Spec^2=1.063"
## [1] "Best Sensitivity_Specificity threshold= 0.34 inc= 0.01"
## [1] "========================================"
## [1] "start= 0.33 end= 0.35 inc= 0.001"
## [1] "Thresh=0.33, Accuracy=68.9%, BaseAcc(Other)=51.2%, Sens=92.1%, Spec=46.9%, Sens^2+Spec^2=1.068"
## [1] "Thresh=0.331, Accuracy=68.9%, BaseAcc(Other)=51.2%, Sens=92%, Spec=47%, Sens^2+Spec^2=1.068"
## [1] "Thresh=0.332, Accuracy=69%, BaseAcc(Other)=51.2%, Sens=91.9%, Spec=47.2%, Sens^2+Spec^2=1.068"
## [1] "Thresh=0.333, Accuracy=69%, BaseAcc(Other)=51.2%, Sens=91.8%, Spec=47.3%, Sens^2+Spec^2=1.068"
## [1] "Thresh=0.334, Accuracy=69.1%, BaseAcc(Other)=51.2%, Sens=91.8%, Spec=47.5%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.335, Accuracy=69.1%, BaseAcc(Other)=51.2%, Sens=91.7%, Spec=47.6%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.336, Accuracy=69.1%, BaseAcc(Other)=51.2%, Sens=91.6%, Spec=47.7%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.337, Accuracy=69.2%, BaseAcc(Other)=51.2%, Sens=91.6%, Spec=47.9%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.338, Accuracy=69.2%, BaseAcc(Other)=51.2%, Sens=91.5%, Spec=48%, Sens^2+Spec^2=1.068"
## [1] "Thresh=0.339, Accuracy=69.2%, BaseAcc(Other)=51.2%, Sens=91.4%, Spec=48.2%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.34, Accuracy=69.3%, BaseAcc(Other)=51.2%, Sens=91.3%, Spec=48.3%, Sens^2+Spec^2=1.068"
## [1] "Thresh=0.341, Accuracy=69.3%, BaseAcc(Other)=51.2%, Sens=91.3%, Spec=48.4%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.342, Accuracy=69.4%, BaseAcc(Other)=51.2%, Sens=91.2%, Spec=48.6%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.343, Accuracy=69.4%, BaseAcc(Other)=51.2%, Sens=91.1%, Spec=48.7%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.344, Accuracy=69.4%, BaseAcc(Other)=51.2%, Sens=91%, Spec=48.9%, Sens^2+Spec^2=1.068"
## [1] "Thresh=0.345, Accuracy=69.5%, BaseAcc(Other)=51.2%, Sens=91%, Spec=49%, Sens^2+Spec^2=1.069"
## [1] "Thresh=0.346, Accuracy=69.5%, BaseAcc(Other)=51.2%, Sens=90.9%, Spec=49.1%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.347, Accuracy=69.5%, BaseAcc(Other)=51.2%, Sens=90.8%, Spec=49.3%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.348, Accuracy=69.5%, BaseAcc(Other)=51.2%, Sens=90.7%, Spec=49.4%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.349, Accuracy=69.6%, BaseAcc(Other)=51.2%, Sens=90.6%, Spec=49.5%, Sens^2+Spec^2=1.068
## [1] "Thresh=0.35, Accuracy=69.6%, BaseAcc(Other)=51.2%, Sens=90.6%, Spec=49.7%, Sens^2+Spec^2=1.068"
## [1] "========================================"
## [1] "Best Threshold=0.345"
## [1] "Best Sensitivity_Specificity=1.06905744668083"
```

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

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

```
result = calcLogisticModelAccuracy (forestTrain$LodgepolePine, Lodge_Ind_Train_predict,
                        curThresh, curThresh, 1, "Lodgepole", "Other", 3)
```

```
## [1] "Model Performance for threshold= 0.345"
## [1] "predicted performance="
##                       Predicted
## Actual                 FALSE=Predict:Other TRUE=Predict:Lodgepole
```

```
##   0=Actual:Other          102229 (TN)          106169 (FP)
##   1=Actual:Lodgepole      17813 (FN)           180498 (TP)
## [1] "Sensitivity= 0.910176440036105 (True positive rate of Lodgepole = TP/(TP+FN) = 180498 /( 180498
## [1] "Specificity= 0.490546934231614 (True negative rate of Other = TN/(TN+FP) = 102229 /( 102229 + 1
## [1] "Sens^2+Spec^2=1.069"
## [1] "Baseline (Other) Accuracy=0.5124"
## [1] "Logistic Accuracy=0.695157"
```

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

```
result = calcLogisticModelAccuracy (forestTest$LodgepolePine, Lodge_Ind_Test_predict,
                       curThresh, curThresh, 1, "Lodgepole", "Other", 3,
                       saveFile=saveFileName, desc="Lodgepole Sig Individualized Vars",
                       AIC=Lodge_Ind_Sig_aic, AUC=Lodge_Ind_Sig_ROC_AUC)
```

```
## [1] "Model Performance for threshold= 0.345"
## [1] "predicted performance="
##                       Predicted
## Actual                FALSE=Predict:Other TRUE=Predict:Lodgepole
##   0=Actual:Other          43983 (TN)           45330 (FP)
##   1=Actual:Lodgepole      7544 (FN)            77446 (TP)
## [1] "Sensitivity= 0.911236616072479 (True positive rate of Lodgepole = TP/(TP+FN) = 77446 /( 77446 +
## [1] "Specificity= 0.492459104497666 (True negative rate of Other = TN/(TN+FP) = 43983 /( 43983 + 453
## [1] "Sens^2+Spec^2=1.072"
## [1] "Baseline (Other) Accuracy=0.5124"
## [1] "Logistic Accuracy=0.696654"
```

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

############## 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
Lodgepole Pine Aggregate All Vars	75.5%	79.3%	71.9%	418219	82.6%	0.476
Lodgepole Pine Individual All Vars	75.6%	78.8%	72.6%	418474	82.6%	0.481
Lodgepole Pine Aggregate Sig Vars	75.5%	79%	72.2%	419472	82.6%	0.482
Lodgepole Pine Individual Sig Vars	69.6%	91.1%	49.2%	438293	80.3%	0.345

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 19:24:44"
```