COVID-19 Cases Model

Starr Corbin

7/24/2020

library(tidyr)  
library(dplyr)  
library(stringr)  
library(ggplot2)  
library(readr)  
library(zoo)  
library(scales)  
library(sf)  
library(urbnmapr)  
library(plotly)  
library(tidyverse)  
library(mlbench)  
library(caret)

#Merge census data with JHU data

usfacts\_data <- read.csv("census\_data\_0718.csv")  
cases <- counties\_sf  
pop\_density <- read.csv("USA\_Population\_Density.csv")  
names(cases)[7] <- "county\_fips\_code"  
cases\_new <- merge(usfacts\_data, cases, by="county\_fips\_code")  
dim(cases\_new)

## [1] 3142 287

cases\_new <- merge(cases\_new, pop\_density, by="county\_fips\_code")  
#Make characters factor for analysis  
cases\_new <- cases\_new %>% mutate\_if(is.character, factor)  
dim(cases\_new)

## [1] 3140 293

#Select Variables

cases\_sel <- cases\_new %>% select(county\_name.x, state,total\_pop, pop\_density,   
 male\_pop, female\_pop, median\_age, white\_pop,   
 black\_pop, asian\_pop, hispanic\_pop, amerindian\_pop,  
 commuters\_by\_public\_transportation, median\_income,   
 employed\_pop, unemployed\_pop, cases\_category)  
  
summary(cases\_sel)

## county\_name.x state total\_pop pop\_density male\_pop female\_pop median\_age white\_pop black\_pop   
## Washington County: 30 TX : 254 Min. : 74 Min. : 0.0 Min. : 39 Min. : 35 Min. :21.60 Min. : 18 Min. : 0.0   
## Jefferson County : 25 GA : 159 1st Qu.: 10952 1st Qu.: 17.1 1st Qu.: 5518 1st Qu.: 5464 1st Qu.:37.90 1st Qu.: 8112 1st Qu.: 95.0   
## Franklin County : 24 VA : 133 Median : 25704 Median : 45.5 Median : 12808 Median : 12887 Median :41.20 Median : 20215 Median : 761.5   
## Lincoln County : 23 KY : 120 Mean : 102224 Mean : 262.8 Mean : 50321 Mean : 51903 Mean :41.16 Mean : 62827 Mean : 12562.2   
## Jackson County : 22 MO : 115 3rd Qu.: 67501 3rd Qu.: 115.6 3rd Qu.: 33510 3rd Qu.: 34114 3rd Qu.:44.23 3rd Qu.: 53551 3rd Qu.: 5397.5   
## Madison County : 19 KS : 105 Max. :10105722 Max. :70148.7 Max. :4979641 Max. :5126081 Max. :66.40 Max. :2676982 Max. :1226134.0   
## (Other) :2997 (Other):2254   
## asian\_pop hispanic\_pop amerindian\_pop commuters\_by\_public\_transportation median\_income employed\_pop unemployed\_pop cases\_category  
## Min. : 0.0 Min. : 0 Min. : 0.0 Min. : 0.0 Min. : 19264 Min. : 39 Min. : 0 High:1842   
## 1st Qu.: 31.0 1st Qu.: 323 1st Qu.: 24.0 1st Qu.: 6.0 1st Qu.: 41126 1st Qu.: 4555 1st Qu.: 286 Med : 544   
## Median : 138.5 Median : 1028 Median : 95.0 Median : 33.0 Median : 48073 Median : 10710 Median : 745 Low : 754   
## Mean : 5410.7 Mean : 17997 Mean : 662.0 Mean : 2422.9 Mean : 49765 Mean : 47960 Mean : 3363   
## 3rd Qu.: 713.8 3rd Qu.: 4874 3rd Qu.: 347.2 3rd Qu.: 145.2 3rd Qu.: 55771 3rd Qu.: 29548 3rd Qu.: 2102   
## Max. :1442577.0 Max. :4893579 Max. :64102.0 Max. :735534.0 Max. :129588 Max. :4805817 Max. :406426   
##

table(complete.cases(cases\_sel))

##   
## TRUE   
## 3140

#Normalize by population

cases\_sel <- cases\_sel %>% mutate(  
 female\_pop = female\_pop / total\_pop,  
 male\_pop = male\_pop / total\_pop,  
 white\_pop = white\_pop / total\_pop,   
 black\_pop = black\_pop / total\_pop,   
 asian\_pop = asian\_pop / total\_pop,   
 hispanic\_pop = hispanic\_pop / total\_pop,   
 amerindian\_pop = amerindian\_pop / total\_pop,  
 commuters\_by\_public\_transportation = commuters\_by\_public\_transportation/ total\_pop,   
 employed\_pop = employed\_pop / total\_pop,   
 unemployed\_pop = unemployed\_pop / total\_pop,   
 )  
  
summary(cases\_sel)

## county\_name.x state total\_pop pop\_density male\_pop female\_pop median\_age white\_pop black\_pop   
## Washington County: 30 TX : 254 Min. : 74 Min. : 0.0 Min. :0.4190 Min. :0.1917 Min. :21.60 Min. :0.006354 Min. :0.000000   
## Jefferson County : 25 GA : 159 1st Qu.: 10952 1st Qu.: 17.1 1st Qu.:0.4890 1st Qu.:0.4942 1st Qu.:37.90 1st Qu.:0.651267 1st Qu.:0.006043   
## Franklin County : 24 VA : 133 Median : 25704 Median : 45.5 Median :0.4960 Median :0.5040 Median :41.20 Median :0.842359 Median :0.021296   
## Lincoln County : 23 KY : 120 Mean : 102224 Mean : 262.8 Mean :0.5008 Mean :0.4992 Mean :41.16 Mean :0.768057 Mean :0.089016   
## Jackson County : 22 MO : 115 3rd Qu.: 67501 3rd Qu.: 115.6 3rd Qu.:0.5058 3rd Qu.:0.5110 3rd Qu.:44.23 3rd Qu.:0.929511 3rd Qu.:0.098830   
## Madison County : 19 KS : 105 Max. :10105722 Max. :70148.7 Max. :0.8083 Max. :0.5810 Max. :66.40 Max. :1.000000 Max. :0.869207   
## (Other) :2997 (Other):2254   
## asian\_pop hispanic\_pop amerindian\_pop commuters\_by\_public\_transportation median\_income employed\_pop unemployed\_pop cases\_category  
## Min. :0.000000 Min. :0.00000 Min. :0.000000 Min. :0.0000000 Min. : 19264 Min. :0.1017 Min. :0.00000 High:1842   
## 1st Qu.:0.002710 1st Qu.:0.02053 1st Qu.:0.001221 1st Qu.:0.0004087 1st Qu.: 41126 1st Qu.:0.3960 1st Qu.:0.02152 Med : 544   
## Median :0.005766 Median :0.03982 Median :0.002704 Median :0.0013915 Median : 48073 Median :0.4429 Median :0.02822 Low : 754   
## Mean :0.013209 Mean :0.09125 Mean :0.017563 Mean :0.0043925 Mean : 49765 Mean :0.4384 Mean :0.02877   
## 3rd Qu.:0.012262 3rd Qu.:0.09289 3rd Qu.:0.006359 3rd Qu.:0.0032615 3rd Qu.: 55771 3rd Qu.:0.4861 3rd Qu.:0.03494   
## Max. :0.418079 Max. :0.99185 Max. :0.822237 Max. :0.3194996 Max. :129588 Max. :0.7326 Max. :0.12619   
##

##Focus on states with Covid-19 outbreaks

cases\_sel %>% pull(cases\_category) %>% table()

## .  
## High Med Low   
## 1842 544 754

## Filter Hard Hit States - Cases

cases\_sel %>% group\_by(state) %>%   
 summarize(high\_pct = sum(cases\_category == "High")/n()) %>%  
 arrange(desc(high\_pct))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 51 x 2  
## state high\_pct  
## <fct> <dbl>  
## 1 TN 0.853  
## 2 WI 0.833  
## 3 MO 0.826  
## 4 CA 0.810  
## 5 MS 0.780  
## 6 OK 0.766  
## 7 KY 0.75   
## 8 ID 0.727  
## 9 WA 0.718  
## 10 SC 0.717  
## # ... with 41 more rows

## Designate States to use for Training model

cases\_train <- cases\_sel %>% filter(  
 state == "NY" |  
 state == "CA" |  
 state == "SC"   
 )  
  
cases\_train %>% pull(cases\_category) %>% table()

## .  
## High Med Low   
## 85 24 57

## Plot Map

counties <- as\_tibble(map\_data("county"))  
counties <- counties %>%   
 rename(c(county = subregion, state = region)) %>%  
 mutate(state = state.abb[match(state, tolower(state.name))]) %>%  
 select(state, county, long, lat, group)  
counties

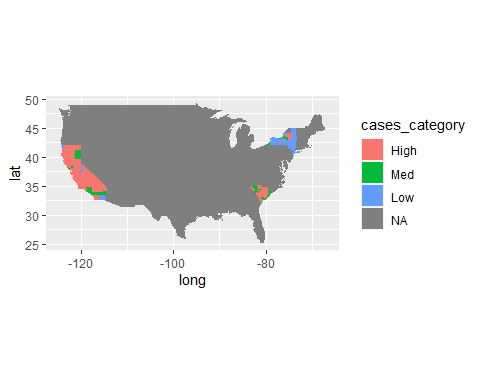
## # A tibble: 87,949 x 5  
## state county long lat group  
## <chr> <chr> <dbl> <dbl> <dbl>  
## 1 AL autauga -86.5 32.3 1  
## 2 AL autauga -86.5 32.4 1  
## 3 AL autauga -86.5 32.4 1  
## 4 AL autauga -86.6 32.4 1  
## 5 AL autauga -86.6 32.4 1  
## 6 AL autauga -86.6 32.4 1  
## 7 AL autauga -86.6 32.4 1  
## 8 AL autauga -86.6 32.4 1  
## 9 AL autauga -86.6 32.4 1  
## 10 AL autauga -86.6 32.4 1  
## # ... with 87,939 more rows

##Add Variables to Map Data

counties\_all <- counties %>% left\_join(cases\_train %>%   
 mutate(county = county\_name.x %>% str\_to\_lower() %>%   
 str\_replace('\\s+county\\s\*$', '')))

## Joining, by = c("state", "county")

ggplot(counties\_all, aes(long, lat)) +   
 geom\_polygon(aes(group = group, fill = cases\_category)) +   
 coord\_quickmap()



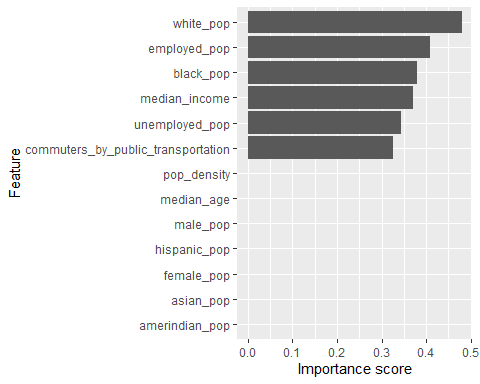
## Check Variable Importance

library(FSelector)  
cases\_weight <- cases\_train %>% select(-county\_name.x, -state, - total\_pop)  
cases\_weight <- cases\_weight %>% chi.squared(cases\_category ~ ., data = .) %>%  
 as\_tibble(rownames = "feature") %>%  
 arrange(desc(attr\_importance))  
cases\_weight

## # A tibble: 13 x 2  
## feature attr\_importance  
## <chr> <dbl>  
## 1 white\_pop 0.479  
## 2 employed\_pop 0.409  
## 3 black\_pop 0.378  
## 4 median\_income 0.371  
## 5 unemployed\_pop 0.344  
## 6 commuters\_by\_public\_transportation 0.325  
## 7 pop\_density 0   
## 8 male\_pop 0   
## 9 female\_pop 0   
## 10 median\_age 0   
## 11 asian\_pop 0   
## 12 hispanic\_pop 0   
## 13 amerindian\_pop 0

#Plot the importance in descending order

ggplot(cases\_weight,  
 aes(x = attr\_importance, y = reorder(feature, attr\_importance))) +  
 geom\_bar(stat = "identity") +  
 xlab("Importance score") + ylab("Feature")



#Get the 5 best features:

subset\_tx <- cutoff.k(cases\_weight %>% column\_to\_rownames("feature"), 5)  
subset\_tx

## [1] "white\_pop" "employed\_pop" "black\_pop" "median\_income" "unemployed\_pop"

#Univariate importance scores

library(rpart)  
library(rpart.plot)  
#deaths\_tx\_new <- deaths\_tx %>% select(-county\_name.x, - total\_pop, -state)  
cases\_train %>% gain.ratio(cases\_category ~ ., data = .) %>%  
 as\_tibble(rownames = "feature") %>%  
 arrange(desc(attr\_importance))

## # A tibble: 16 x 2  
## feature attr\_importance  
## <chr> <dbl>  
## 1 state 0.384  
## 2 county\_name.x 0.191  
## 3 white\_pop 0.182  
## 4 black\_pop 0.153  
## 5 employed\_pop 0.129  
## 6 commuters\_by\_public\_transportation 0.124  
## 7 median\_income 0.119  
## 8 unemployed\_pop 0.101  
## 9 total\_pop 0   
## 10 pop\_density 0   
## 11 male\_pop 0   
## 12 female\_pop 0   
## 13 median\_age 0   
## 14 asian\_pop 0   
## 15 hispanic\_pop 0   
## 16 amerindian\_pop 0

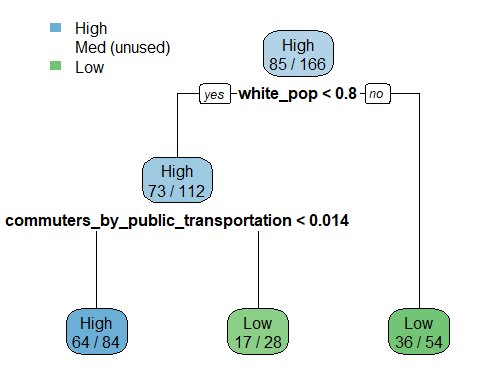
## Build a Model - Decision Tree

library(caret)  
fit\_cases <- cases\_train %>%  
 train(cases\_category ~ . - county\_name.x - state - total\_pop,   
 data = . ,  
 method = "rpart",  
 trControl = trainControl(method = "cv", number = 10)  
 )

fit\_cases

## CART   
##   
## 166 samples  
## 16 predictor  
## 3 classes: 'High', 'Med', 'Low'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 150, 148, 149, 150, 150, 151, ...   
## Resampling results across tuning parameters:  
##   
## cp Accuracy Kappa   
## 0.07407407 0.6459232 0.36634434  
## 0.09876543 0.5931373 0.26136782  
## 0.29629630 0.5317402 0.07952407  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.07407407.

library(rpart.plot)  
rpart.plot(fit\_cases$finalModel, extra = 2)



varImp(fit\_cases)

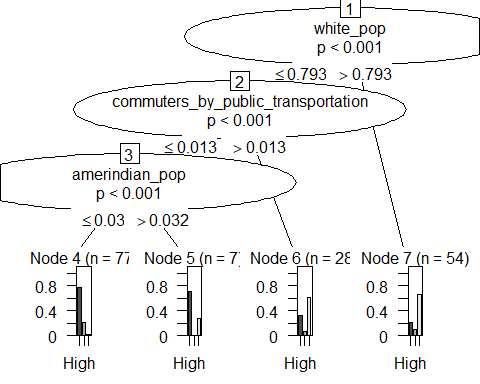
## rpart variable importance  
##   
## Overall  
## employed\_pop 100.00  
## white\_pop 96.50  
## amerindian\_pop 93.35  
## black\_pop 76.91  
## commuters\_by\_public\_transportation 69.60  
## pop\_density 54.92  
## median\_income 52.71  
## hispanic\_pop 0.00  
## median\_age 0.00  
## male\_pop 0.00  
## female\_pop 0.00  
## unemployed\_pop 0.00  
## asian\_pop 0.00

## Build a Model - Conditional Inference Tree

ctreeFit\_cases <- cases\_train %>%  
train(cases\_category ~ . - county\_name.x - state - total\_pop,  
 data = . ,  
 method = "ctree",  
 trControl = trainControl(method = "cv", number = 10)  
 )  
ctreeFit\_cases

## Conditional Inference Tree   
##   
## 166 samples  
## 16 predictor  
## 3 classes: 'High', 'Med', 'Low'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 150, 150, 149, 149, 150, 150, ...   
## Resampling results across tuning parameters:  
##   
## mincriterion Accuracy Kappa   
## 0.01 0.6292974 0.3408391  
## 0.50 0.6466176 0.3668450  
## 0.99 0.6594853 0.4038657  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mincriterion = 0.99.

plot(ctreeFit\_cases$finalModel)



predict(ctreeFit\_cases, head(cases\_train))

## [1] Low High High High Low High  
## Levels: High Med Low

##Build a Model - C.45

C45fit\_cases <- cases\_train %>%  
train(cases\_category ~ . - county\_name.x - state - total\_pop,  
 data = . ,  
 method = "J48",  
 trControl = trainControl(method = "cv", number = 10)  
 )  
C45fit\_cases

## C4.5-like Trees   
##   
## 166 samples  
## 16 predictor  
## 3 classes: 'High', 'Med', 'Low'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 150, 151, 150, 148, 148, 149, ...   
## Resampling results across tuning parameters:  
##   
## C M Accuracy Kappa   
## 0.010 1 0.6354739 0.3620164  
## 0.010 2 0.6528350 0.3907327  
## 0.010 3 0.6284886 0.3371782  
## 0.255 1 0.6217320 0.3493007  
## 0.255 2 0.6346487 0.3696478  
## 0.255 3 0.6259886 0.3522991  
## 0.500 1 0.6217320 0.3493007  
## 0.500 2 0.6346487 0.3696478  
## 0.500 3 0.6259886 0.3522991  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were C = 0.01 and M = 2.

C45fit\_cases$finalModel

## J48 pruned tree  
## ------------------  
##   
## white\_pop <= 0.792661  
## | commuters\_by\_public\_transportation <= 0.013532  
## | | employed\_pop <= 0.332063  
## | | | black\_pop <= 0.051724: Low (3.0)  
## | | | black\_pop > 0.051724: Med (3.0/1.0)  
## | | employed\_pop > 0.332063: High (78.0/15.0)  
## | commuters\_by\_public\_transportation > 0.013532  
## | | black\_pop <= 0.08605  
## | | | white\_pop <= 0.666508: High (9.0/1.0)  
## | | | white\_pop > 0.666508: Low (3.0/1.0)  
## | | black\_pop > 0.08605: Low (16.0/1.0)  
## white\_pop > 0.792661  
## | black\_pop <= 0.010904  
## | | pop\_density <= 56.7: High (9.0/1.0)  
## | | pop\_density > 56.7: Low (5.0/2.0)  
## | black\_pop > 0.010904  
## | | median\_income <= 45332: Med (3.0/1.0)  
## | | median\_income > 45332: Low (37.0/4.0)  
##   
## Number of Leaves : 10  
##   
## Size of the tree : 19

##Build a Model - PART

#Remove county name since it has too many levels which will make the code really slow  
rulesfit\_cases <- cases\_train %>%  
train(cases\_category ~ . - county\_name.x - state - total\_pop,  
 data = . ,  
 method = "PART",  
 trControl = trainControl(method = "cv", number = 10)  
 )  
rulesfit\_cases

## Rule-Based Classifier   
##   
## 166 samples  
## 16 predictor  
## 3 classes: 'High', 'Med', 'Low'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 149, 149, 150, 150, 150, 149, ...   
## Resampling results across tuning parameters:  
##   
## threshold pruned Accuracy Kappa   
## 0.010 yes 0.6444853 0.3902148  
## 0.010 no 0.6694853 0.4273642  
## 0.255 yes 0.6209559 0.3573303  
## 0.255 no 0.6694853 0.4273642  
## 0.500 yes 0.6147059 0.3481792  
## 0.500 no 0.6694853 0.4273642  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were threshold = 0.5 and pruned = no.

rulesfit\_cases$finalModel

## PART decision list  
## ------------------  
##   
## white\_pop > 0.792661 AND  
## black\_pop > 0.010904 AND  
## median\_income > 45332: Low (37.0/4.0)  
##   
## commuters\_by\_public\_transportation <= 0.013532 AND  
## employed\_pop > 0.332063 AND  
## white\_pop <= 0.947953 AND  
## male\_pop <= 0.503818: High (70.0/18.0)  
##   
## male\_pop > 0.50093 AND  
## male\_pop <= 0.552539 AND  
## employed\_pop > 0.332063: High (27.0/1.0)  
##   
## asian\_pop <= 0.11878 AND  
## employed\_pop > 0.287639: Low (23.0/1.0)  
##   
## male\_pop <= 0.492691 AND  
## pop\_density <= 1814.6: High (3.0)  
##   
## male\_pop > 0.491572 AND  
## pop\_density <= 25.3: High (2.0/1.0)  
##   
## male\_pop <= 0.491572: High (2.0/1.0)  
##   
## : Med (2.0)  
##   
## Number of Rules : 8

## Build a Model - Random Forest Fit

randomForestFit\_cases <- cases\_train %>%   
 train(cases\_category ~ . - county\_name.x - state - total\_pop,  
 data = . ,  
 method = "rf",  
 trControl = trainControl(method = "cv", number = 10)  
 )  
randomForestFit\_cases

## Random Forest   
##   
## 166 samples  
## 16 predictor  
## 3 classes: 'High', 'Med', 'Low'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 149, 149, 151, 151, 149, 150, ...   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 2 0.7607680 0.5722855  
## 7 0.7439052 0.5450339  
## 13 0.7320997 0.5314329  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 2.

## Artificial Neural Network

nnetFit\_case <- cases\_train %>% train(cases\_category ~ . - county\_name.x - state - total\_pop,  
 method = "nnet",  
 data = .,  
 tuneLength = 5,  
 trControl = trainControl(method = "cv", number = 10),  
 trace = FALSE)  
nnetFit\_case

## Neural Network   
##   
## 166 samples  
## 16 predictor  
## 3 classes: 'High', 'Med', 'Low'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 149, 150, 150, 148, 150, 149, ...   
## Resampling results across tuning parameters:  
##   
## size decay Accuracy Kappa   
## 1 0e+00 0.5125245 0.00000000  
## 1 1e-04 0.5125245 0.00000000  
## 1 1e-03 0.5125245 0.00000000  
## 1 1e-02 0.5125245 0.00000000  
## 1 1e-01 0.5187745 0.02631579  
## 3 0e+00 0.5242892 0.02816901  
## 3 1e-04 0.5125245 0.00000000  
## 3 1e-03 0.5843791 0.17426425  
## 3 1e-02 0.5485539 0.09914965  
## 3 1e-01 0.5172467 0.03280658  
## 5 0e+00 0.5368382 0.05922417  
## 5 1e-04 0.5423938 0.07195144  
## 5 1e-03 0.5678186 0.16962067  
## 5 1e-02 0.6030392 0.24635778  
## 5 1e-01 0.5780474 0.20305433  
## 7 0e+00 0.5250735 0.03105516  
## 7 1e-04 0.5309559 0.04898619  
## 7 1e-03 0.5542892 0.11347280  
## 7 1e-02 0.6748121 0.38197068  
## 7 1e-01 0.5031291 0.06014253  
## 9 0e+00 0.5309559 0.04898619  
## 9 1e-04 0.5357271 0.05528478  
## 9 1e-03 0.5178431 0.06155136  
## 9 1e-02 0.5621569 0.15423663  
## 9 1e-01 0.5110458 0.05177729  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were size = 7 and decay = 0.01.

nnetFit\_case$finalModel

## a 13-7-3 network with 122 weights  
## inputs: pop\_density male\_pop female\_pop median\_age white\_pop black\_pop asian\_pop hispanic\_pop amerindian\_pop commuters\_by\_public\_transportation median\_income employed\_pop unemployed\_pop   
## output(s): .outcome   
## options were - softmax modelling decay=0.01

## Compare Models

resamps <- resamples(list(  
 decision = fit\_cases,  
 ctree = ctreeFit\_cases,  
 rules = rulesfit\_cases,  
 randomForest = randomForestFit\_cases,  
 NeuralNet = nnetFit\_case,   
 C45fit = C45fit\_cases  
 ))  
resamps

##   
## Call:  
## resamples.default(x = list(decision = fit\_cases, ctree = ctreeFit\_cases, rules = rulesfit\_cases, randomForest = randomForestFit\_cases, NeuralNet = nnetFit\_case, C45fit  
## = C45fit\_cases))  
##   
## Models: decision, ctree, rules, randomForest, NeuralNet, C45fit   
## Number of resamples: 10   
## Performance metrics: Accuracy, Kappa   
## Time estimates for: everything, final model fit

summary(resamps)

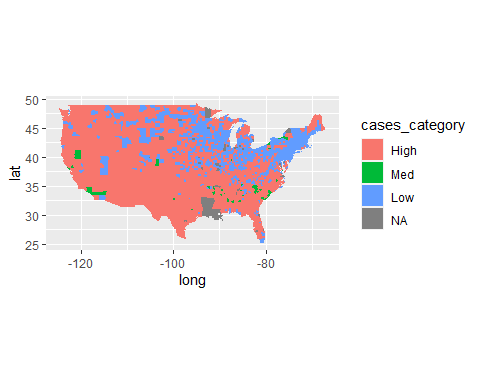
##   
## Call:  
## summary.resamples(object = resamps)  
##   
## Models: decision, ctree, rules, randomForest, NeuralNet, C45fit   
## Number of resamples: 10   
##   
## Accuracy   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## decision 0.5000000 0.5974265 0.6568627 0.6459232 0.7012868 0.7500000 0  
## ctree 0.4705882 0.5939542 0.6562500 0.6594853 0.7218750 0.8235294 0  
## rules 0.5294118 0.5836397 0.6470588 0.6694853 0.7058824 0.8750000 0  
## randomForest 0.6111111 0.7127451 0.7823529 0.7607680 0.8235294 0.8750000 0  
## NeuralNet 0.3529412 0.5885417 0.7261029 0.6748121 0.7944444 0.8235294 0  
## C45fit 0.5000000 0.5939542 0.6568627 0.6528350 0.7135417 0.8000000 0  
##   
## Kappa   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## decision 0.04477612 0.2693665 0.4075674 0.3663443 0.4505796 0.5616438 0  
## ctree 0.07272727 0.2960199 0.3926941 0.4038657 0.4898649 0.6812500 0  
## rules 0.12500000 0.2646703 0.4220546 0.4273642 0.5135151 0.7746479 0  
## randomForest 0.33333333 0.4758321 0.6169924 0.5722855 0.6719323 0.7714286 0  
## NeuralNet -0.19108280 0.1024590 0.5122493 0.3819707 0.6239148 0.6871166 0  
## C45fit 0.13513514 0.2708333 0.3803071 0.3907327 0.5020121 0.6739130 0

## Use model from hard hit states for all of US

cases\_pred <- na.omit(cases\_sel)  
cases\_pred$cases\_category <- predict(randomForestFit\_cases, cases\_pred)  
counties\_pred <- counties %>% left\_join(cases\_pred %>%   
 mutate(county = county\_name.x %>% str\_to\_lower() %>%   
 str\_replace('\\s+county\\s\*$', '')))

## Joining, by = c("state", "county")

ggplot(counties\_pred, aes(long, lat)) +   
 geom\_polygon(aes(group = group, fill = cases\_category)) +   
 coord\_quickmap()



write.csv(cases\_pred, "C:/Users/Starr/Google Drive (starr.corbin@gmail.com)/School/SMU/Data Mining/Project 3/case\_predictor\_rules.csv")