Chicago Crime 2

Gil Graybill

2/27/2020

# Summary of Models

## Linear regression on Training Data

AIC: 125535

## Empty Model on Training Data

AIC: 184117

## Forward Stepwise regression on Training Data

AIC: 125534

## Backward Stepwise regression on Training Data

AIC: 125534

## Logical Threshold

AUC is 0.8297931 determine the probability threshold that best balances speciﬁcity and sensitivity is 0.1812335. Accuracy is 73.1% Best accuracy is 86.9% at p > 0.6

## Decision tree 1

One variable – FBI Code If it is one of those codes, there’s a 14% chance of arrest If it’s not, there’s a 91% chance

No info rate – 0.7991 Training data – 0.8676 Testing data - 0.8676 - No overfit

## Decision tree 2

Make cp = 0.001 Plot still indicates one variable is best fit

## Decision Tree 3 - Regression Tree

2nd variable helps a little

## Random Forest

No Information Rate : 0.7991 Accuracy on training data : 0.8706 Testing data accuracy is 0.8633 FBI Code is most important variable 100, In\_residence is 12, Domestic is 7,

Accuracy : 0.8731 training Accuracy : 0.8733 testing

## Ensemble:

Rpart and glm has a correlation of 0.8. The resulting ROC is: 0.8368, higher than glm (0.83) ranger(0.81) rpart(0.79) Accuracy : 0.8701 on training Accuracy : 0.8701 on testing

## Stacking:

ROC is 0.836783 Training Accuracy: 0.8701 Testing Accuracy: 0.8701

library(MASS)  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.6.2

## -- Attaching packages ---------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::select() masks MASS::select()

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

library(GGally)

## Warning: package 'GGally' was built under R version 3.6.2

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

##   
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':  
##   
## nasa

library(leaps)

## Warning: package 'leaps' was built under R version 3.6.2

library(caret)

## Warning: package 'caret' was built under R version 3.6.2

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(e1071)

## Warning: package 'e1071' was built under R version 3.6.2

library(ROCR)

## Warning: package 'ROCR' was built under R version 3.6.2

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.6.2

##   
## Attaching package: 'gplots'

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

library(rpart) #for classification trees  
library(RColorBrewer) #better visualization of classification trees  
library(rattle) #better visualization of classification trees

## Warning: package 'rattle' was built under R version 3.6.2

## Rattle: A free graphical interface for data science with R.  
## Version 5.3.0 Copyright (c) 2006-2018 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(stringr)  
library(data.table)

## Warning: package 'data.table' was built under R version 3.6.2

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':  
##   
## hour, isoweek, mday, minute, month, quarter, second, wday,  
## week, yday, year

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

## The following object is masked from 'package:purrr':  
##   
## transpose

library(VIM)

## Warning: package 'VIM' was built under R version 3.6.2

## Loading required package: colorspace

## Loading required package: grid

## VIM is ready to use.   
## Since version 4.0.0 the GUI is in its own package VIMGUI.  
##   
## Please use the package to use the new (and old) GUI.

## Suggestions and bug-reports can be submitted at: https://github.com/alexkowa/VIM/issues

##   
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':  
##   
## sleep

library(mice)

## Warning: package 'mice' was built under R version 3.6.2

## Registered S3 methods overwritten by 'lme4':  
## method from  
## cooks.distance.influence.merMod car   
## influence.merMod car   
## dfbeta.influence.merMod car   
## dfbetas.influence.merMod car

##   
## Attaching package: 'mice'

## The following object is masked from 'package:tidyr':  
##   
## complete

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(ranger)

## Warning: package 'ranger' was built under R version 3.6.2

##   
## Attaching package: 'ranger'

## The following object is masked from 'package:rattle':  
##   
## importance

library(caretEnsemble)

## Warning: package 'caretEnsemble' was built under R version 3.6.2

##   
## Attaching package: 'caretEnsemble'

## The following object is masked from 'package:ggplot2':  
##   
## autoplot

library(nnet)

Chicago\_crime <- read\_csv("chicago.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## ID = col\_double(),  
## Arrest = col\_logical(),  
## Domestic = col\_logical(),  
## Ward = col\_double(),  
## `Community Area` = col\_double(),  
## `X Coordinate` = col\_double(),  
## `Y Coordinate` = col\_double(),  
## Year = col\_double(),  
## Latitude = col\_double(),  
## Longitude = col\_double()  
## )

## See spec(...) for full column specifications.

Chicago\_crime = Chicago\_crime %>% select(-ID,-`Case Number`,-`Updated On`,-`X Coordinate`,-`Y Coordinate`)  
str(Chicago\_crime)

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 267185 obs. of 17 variables:  
## $ Date : chr "01/01/2018 12:00:00 AM" "01/01/2018 12:00:00 AM" "01/01/2018 12:00:00 AM" "01/01/2018 12:00:00 AM" ...  
## $ Block : chr "069XX N CLARK ST" "070XX N KEDZIE AVE" "072XX W BALMORAL AVE" "047XX N ARTESIAN AVE" ...  
## $ IUCR : chr "1753" "1130" "1153" "1752" ...  
## $ Primary Type : chr "OFFENSE INVOLVING CHILDREN" "DECEPTIVE PRACTICE" "DECEPTIVE PRACTICE" "OFFENSE INVOLVING CHILDREN" ...  
## $ Description : chr "SEX ASSLT OF CHILD BY FAM MBR" "FRAUD OR CONFIDENCE GAME" "FINANCIAL IDENTITY THEFT OVER $ 300" "AGG CRIM SEX ABUSE FAM MEMBER" ...  
## $ Location Description: chr "RESIDENCE-GARAGE" "APARTMENT" "RESIDENCE" "RESIDENCE" ...  
## $ Arrest : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Domestic : logi FALSE FALSE FALSE TRUE FALSE FALSE ...  
## $ Beat : chr "2431" "2411" "1613" "1911" ...  
## $ District : chr "024" "024" "016" "019" ...  
## $ Ward : num 49 50 41 40 16 21 27 15 20 41 ...  
## $ Community Area : num 1 2 10 4 61 73 28 67 61 76 ...  
## $ FBI Code : chr "02" "11" "11" "17" ...  
## $ Year : num 2018 2018 2018 2018 2018 ...  
## $ Latitude : num NA NA NA NA NA ...  
## $ Longitude : num NA NA NA NA NA ...  
## $ Location : chr NA NA NA NA ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. ID = col\_double(),  
## .. `Case Number` = col\_character(),  
## .. Date = col\_character(),  
## .. Block = col\_character(),  
## .. IUCR = col\_character(),  
## .. `Primary Type` = col\_character(),  
## .. Description = col\_character(),  
## .. `Location Description` = col\_character(),  
## .. Arrest = col\_logical(),  
## .. Domestic = col\_logical(),  
## .. Beat = col\_character(),  
## .. District = col\_character(),  
## .. Ward = col\_double(),  
## .. `Community Area` = col\_double(),  
## .. `FBI Code` = col\_character(),  
## .. `X Coordinate` = col\_double(),  
## .. `Y Coordinate` = col\_double(),  
## .. Year = col\_double(),  
## .. `Updated On` = col\_character(),  
## .. Latitude = col\_double(),  
## .. Longitude = col\_double(),  
## .. Location = col\_character()  
## .. )

summary(Chicago\_crime)

## Date Block IUCR   
## Length:267185 Length:267185 Length:267185   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## Primary Type Description Location Description  
## Length:267185 Length:267185 Length:267185   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## Arrest Domestic Beat District   
## Mode :logical Mode :logical Length:267185 Length:267185   
## FALSE:213769 FALSE:223427 Class :character Class :character   
## TRUE :53416 TRUE :43758 Mode :character Mode :character   
##   
##   
##   
##   
## Ward Community Area FBI Code Year   
## Min. : 1.00 Min. : 0.00 Length:267185 Min. :2018   
## 1st Qu.:10.00 1st Qu.:23.00 Class :character 1st Qu.:2018   
## Median :24.00 Median :32.00 Mode :character Median :2018   
## Mean :23.45 Mean :36.47 Mean :2018   
## 3rd Qu.:35.00 3rd Qu.:53.00 3rd Qu.:2018   
## Max. :50.00 Max. :77.00 Max. :2018   
## NA's :4 NA's :2   
## Latitude Longitude Location   
## Min. :41.65 Min. :-87.93 Length:267185   
## 1st Qu.:41.77 1st Qu.:-87.71 Class :character   
## Median :41.87 Median :-87.66 Mode :character   
## Mean :41.84 Mean :-87.67   
## 3rd Qu.:41.91 3rd Qu.:-87.63   
## Max. :42.02 Max. :-87.53   
## NA's :4365 NA's :4365

Mutate the dates

Chicago\_crime = Chicago\_crime %>% mutate(Date = mdy\_hms(Date))  
Chicago\_crime = Chicago\_crime %>% mutate(Month = month(Date))  
Chicago\_crime = Chicago\_crime %>% mutate(Day = day(Date))  
Chicago\_crime = Chicago\_crime %>% mutate(Hour = hour(Date))  
str(Chicago\_crime)

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 267185 obs. of 20 variables:  
## $ Date : POSIXct, format: "2018-01-01 00:00:00" "2018-01-01 00:00:00" ...  
## $ Block : chr "069XX N CLARK ST" "070XX N KEDZIE AVE" "072XX W BALMORAL AVE" "047XX N ARTESIAN AVE" ...  
## $ IUCR : chr "1753" "1130" "1153" "1752" ...  
## $ Primary Type : chr "OFFENSE INVOLVING CHILDREN" "DECEPTIVE PRACTICE" "DECEPTIVE PRACTICE" "OFFENSE INVOLVING CHILDREN" ...  
## $ Description : chr "SEX ASSLT OF CHILD BY FAM MBR" "FRAUD OR CONFIDENCE GAME" "FINANCIAL IDENTITY THEFT OVER $ 300" "AGG CRIM SEX ABUSE FAM MEMBER" ...  
## $ Location Description: chr "RESIDENCE-GARAGE" "APARTMENT" "RESIDENCE" "RESIDENCE" ...  
## $ Arrest : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Domestic : logi FALSE FALSE FALSE TRUE FALSE FALSE ...  
## $ Beat : chr "2431" "2411" "1613" "1911" ...  
## $ District : chr "024" "024" "016" "019" ...  
## $ Ward : num 49 50 41 40 16 21 27 15 20 41 ...  
## $ Community Area : num 1 2 10 4 61 73 28 67 61 76 ...  
## $ FBI Code : chr "02" "11" "11" "17" ...  
## $ Year : num 2018 2018 2018 2018 2018 ...  
## $ Latitude : num NA NA NA NA NA ...  
## $ Longitude : num NA NA NA NA NA ...  
## $ Location : chr NA NA NA NA ...  
## $ Month : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Day : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Hour : int 0 0 0 0 0 0 0 0 0 0 ...

Factor the data

Chicago\_crime = Chicago\_crime %>% mutate(IUCR = as.factor(IUCR)) %>%  
 mutate(Beat = as.factor(Beat)) %>%  
 mutate(Arrest = as.factor(Arrest)) %>%  
 mutate(Domestic = as.factor(Domestic)) %>%  
 mutate(`Primary Type` = as.factor(`Primary Type`)) %>%  
 mutate(District = as.factor(District)) %>%  
 mutate(`Location Description` = as.factor(`Location Description`)) %>%  
 mutate(Ward = as.factor(Ward)) %>%  
 mutate(`Community Area` = as.factor(`Community Area`)) %>%  
 mutate(Month = as.factor(Month)) %>%  
 mutate(Hour = as.factor(Hour)) %>%  
 mutate(`FBI Code` = as.factor(`FBI Code`))  
summary(Chicago\_crime)

## Date Block IUCR   
## Min. :2018-01-01 00:00:00 Length:267185 0820 : 24772   
## 1st Qu.:2018-04-12 16:00:00 Class :character 0486 : 24221   
## Median :2018-07-06 13:15:00 Mode :character 0460 : 16079   
## Mean :2018-07-05 06:54:08 0810 : 15251   
## 3rd Qu.:2018-09-28 09:00:00 0560 : 13428   
## Max. :2018-12-31 00:00:00 1310 : 13098   
## (Other):160336   
## Primary Type Description Location Description  
## THEFT :65088 Length:267185 STREET :58900   
## BATTERY :49704 Class :character RESIDENCE:44814   
## CRIMINAL DAMAGE :27727 Mode :character APARTMENT:34559   
## ASSAULT :20358 SIDEWALK :21098   
## DECEPTIVE PRACTICE:19300 OTHER :10896   
## OTHER OFFENSE :17205 (Other) :95975   
## (Other) :67803 NA's : 943   
## Arrest Domestic Beat District   
## FALSE:213769 FALSE:223427 1834 : 3326 011 : 19146   
## TRUE : 53416 TRUE : 43758 0112 : 2543 006 : 16455   
## 1831 : 2366 008 : 16337   
## 0111 : 2306 018 : 16172   
## 1112 : 2189 001 : 15639   
## 0421 : 2133 007 : 14266   
## (Other):252322 (Other):169170   
## Ward Community Area FBI Code Year   
## 42 : 18107 25 : 15105 06 :65088 Min. :2018   
## 24 : 12616 8 : 13061 08B :42047 1st Qu.:2018   
## 28 : 11901 32 : 10860 14 :27727 Median :2018   
## 27 : 11212 28 : 9424 26 :24758 Mean :2018   
## 2 : 10072 29 : 9395 11 :17639 3rd Qu.:2018   
## (Other):203273 (Other):209338 08A :14622 Max. :2018   
## NA's : 4 NA's : 2 (Other):75304   
## Latitude Longitude Location Month   
## Min. :41.65 Min. :-87.93 Length:267185 8 : 25356   
## 1st Qu.:41.77 1st Qu.:-87.71 Class :character 7 : 25201   
## Median :41.87 Median :-87.66 Mode :character 5 : 24682   
## Mean :41.84 Mean :-87.67 6 : 24184   
## 3rd Qu.:41.91 3rd Qu.:-87.63 9 : 23033   
## Max. :42.02 Max. :-87.53 10 : 22789   
## NA's :4365 NA's :4365 (Other):121940   
## Day Hour   
## Min. : 1.00 12 : 16303   
## 1st Qu.: 8.00 18 : 15205   
## Median :16.00 19 : 15193   
## Mean :15.67 15 : 14959   
## 3rd Qu.:23.00 17 : 14788   
## Max. :31.00 16 : 14337   
## (Other):176400

arrest\_percent = count(Chicago\_crime %>% filter(Arrest == TRUE))/nrow(Chicago\_crime)   
arrest\_percent

## n  
## 1 0.1999214

nlevels(Chicago\_crime$IUCR)

## [1] 322

nlevels(Chicago\_crime$Ward)

## [1] 50

nlevels(Chicago\_crime$Beat)

## [1] 274

nlevels(Chicago\_crime$District)

## [1] 23

nlevels(Chicago\_crime$`Community Area`)

## [1] 78

nlevels(Chicago\_crime$`Primary Type`)

## [1] 32

nlevels(Chicago\_crime$`Location Description`)

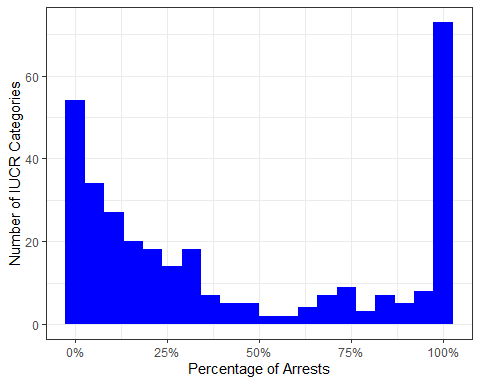
## [1] 132

nlevels(Chicago\_crime$`FBI Code`)

## [1] 26

Analyze IUCR

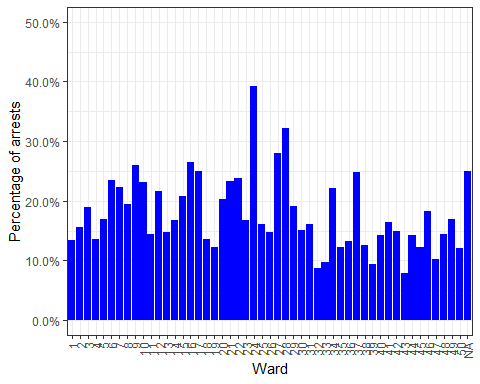
t1 = table(Chicago\_crime$Arrest, Chicago\_crime$IUCR)   
IUCR\_probs = as.data.frame(prop.table(t1, margin = 2))  
IUCR\_probs\_true = IUCR\_probs[IUCR\_probs$Var1==TRUE,]   
ggplot(IUCR\_probs\_true,aes(x=Freq)) + geom\_histogram(bins = 20, fill = "blue") +  
 theme\_bw() +  
 labs(y = "Number of IUCR Categories", x = 'Percentage of Arrests') +  
 scale\_x\_continuous(labels = scales::percent\_format())



Analyze Ward

ggplot(Chicago\_crime, aes(x=Ward, fill = Arrest)) + geom\_bar(position="fill") + theme\_bw() +  
 scale\_fill\_manual("legend", values = c("TRUE" = "Blue", "FALSE" = "White")) +  
 theme(legend.position = "none") +  
 labs(y = "Percentage of arrests") +   
 theme(axis.text.x = element\_text(angle = 90,vjust =0.5, hjust=1)) +  
 scale\_y\_continuous(labels = scales::percent\_format(), limits = c(0, 0.5))

## Warning: Removed 51 rows containing missing values (geom\_bar).



t2 = table(Chicago\_crime$Arrest, Chicago\_crime$Ward) #create a table object  
ward\_prop\_table = as.data.frame(prop.table(t2, margin = 2 )) %>% filter(Var1 == "TRUE")  
summary(ward\_prop\_table$Freq)

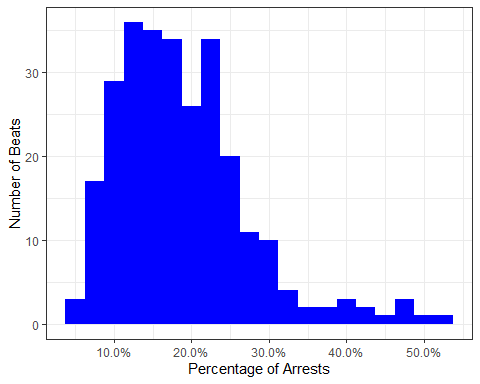
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.07872 0.13557 0.16271 0.17751 0.22041 0.39157

Analyze Beat

t3 = table(Chicago\_crime$Arrest, Chicago\_crime$Beat)   
Beat\_probs = as.data.frame(prop.table(t3, margin = 2))  
Beat\_probs\_true = Beat\_probs[Beat\_probs$Var1==TRUE,]  
summary(Beat\_probs\_true$Freq)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.04691 0.12825 0.17438 0.18869 0.23334 0.52183

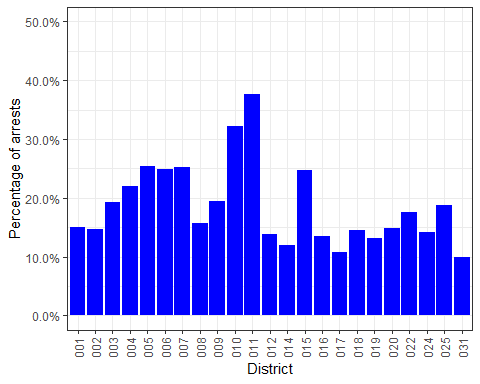
ggplot(Beat\_probs\_true,aes(x=Freq)) + geom\_histogram(bins = 20, fill = "blue") +  
 theme\_bw() +  
 labs(y = "Number of Beats", x = 'Percentage of Arrests') +  
 scale\_x\_continuous(labels = scales::percent\_format())



Analyze District

ggplot(Chicago\_crime, aes(x=District, fill = Arrest)) + geom\_bar(position="fill") + theme\_bw() +  
 scale\_fill\_manual("legend", values = c("TRUE" = "Blue", "FALSE" = "White")) +  
 theme(legend.position = "none") +  
 labs(y = "Percentage of arrests") +   
 theme(axis.text.x = element\_text(angle = 90,vjust =0.5, hjust=1)) +  
 scale\_y\_continuous(labels = scales::percent\_format(), limits = c(0, 0.5))

## Warning: Removed 23 rows containing missing values (geom\_bar).



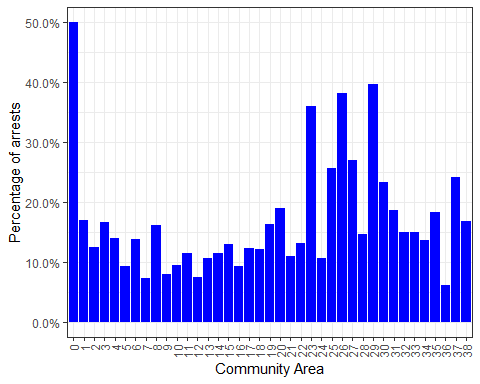
t4 = table(Chicago\_crime$Arrest, Chicago\_crime$District) #create a table object  
district\_prop\_table = as.data.frame(prop.table(t4, margin = 2 )) %>% filter(Var1 == "TRUE")  
summary(district\_prop\_table$Freq)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.1000 0.1398 0.1571 0.1867 0.2337 0.3764

Analyze Community Areas

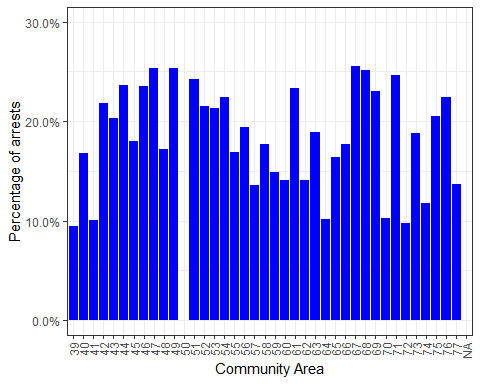
community\_areas = levels(Chicago\_crime$`Community Area`)  
half = length(community\_areas) / 2  
first\_half = community\_areas[1:half]  
Chicago\_crime %>%  
 filter(`Community Area` %in% first\_half) %>%   
 ggplot(aes(x=`Community Area`, fill = Arrest)) +   
 geom\_bar(position="fill") + theme\_bw() +  
 scale\_fill\_manual("legend", values = c("TRUE" = "Blue", "FALSE" = "White")) +  
 theme(legend.position = "none") +  
 labs(y = "Percentage of arrests") +   
 theme(axis.text.x = element\_text(angle = 90,vjust =0.5, hjust=1)) +  
 scale\_y\_continuous(labels = scales::percent\_format(), limits = c(0, 0.5))

## Warning: Removed 39 rows containing missing values (geom\_bar).



second\_half = community\_areas[half+1:length(community\_areas) ]  
Chicago\_crime %>%  
 filter(`Community Area` %in% second\_half) %>%   
 ggplot(aes(x=`Community Area`, fill = Arrest)) +   
 geom\_bar(position="fill") + theme\_bw() +  
 scale\_fill\_manual("legend", values = c("TRUE" = "Blue", "FALSE" = "White")) +  
 theme(legend.position = "none") +  
 labs(y = "Percentage of arrests") +   
 theme(axis.text.x = element\_text(angle = 90,vjust =0.5, hjust=1)) +  
 scale\_y\_continuous(labels = scales::percent\_format(), limits = c(0, 0.3))

## Warning: Removed 41 rows containing missing values (geom\_bar).



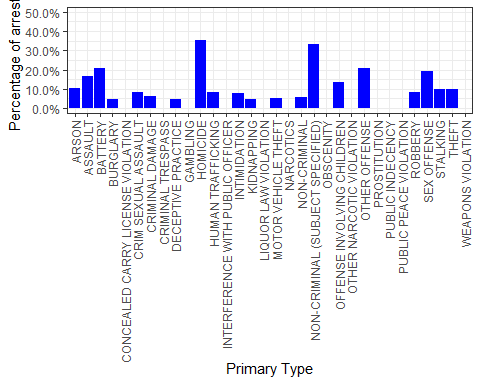
t5 = table(Chicago\_crime$Arrest, Chicago\_crime$`Community Area`) #create a table object  
comm\_prop\_table = as.data.frame(prop.table(t5, margin = 2 )) %>% filter(Var1 == "TRUE")  
summary(comm\_prop\_table$Freq)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.06092 0.12328 0.16778 0.17931 0.22401 0.50000

Analyze Primary Type

ggplot(Chicago\_crime, aes(x=`Primary Type`, fill = Arrest)) + geom\_bar(position="fill") + theme\_bw() +  
 scale\_fill\_manual("legend", values = c("TRUE" = "Blue", "FALSE" = "White")) +  
 theme(legend.position = "none") +  
 labs(y = "Percentage of arrests") +   
 theme(axis.text.x = element\_text(angle = 90,vjust =0.5, hjust=1)) +  
 scale\_y\_continuous(labels = scales::percent\_format(), limits = c(0, 0.5))

## Warning: Removed 39 rows containing missing values (geom\_bar).



t6 = table(Chicago\_crime$Arrest, Chicago\_crime$`Primary Type`) #create a table object  
primary\_prop\_table = as.data.frame(prop.table(t6, margin = 2 )) %>% filter(Var1 == "TRUE")  
summary(primary\_prop\_table$Freq)

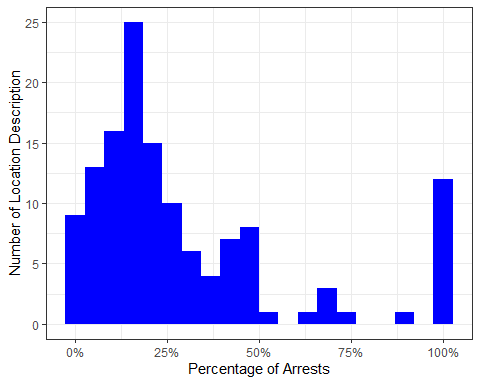
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0810 0.1780 0.3784 0.7226 1.0000

Analyze Location Description

t7 = table(Chicago\_crime$Arrest, Chicago\_crime$`Location Description`)   
Location\_probs = as.data.frame(prop.table(t7, margin = 2))  
Location\_probs\_true = Location\_probs[Location\_probs$Var1==TRUE,]  
summary(Location\_probs\_true$Freq)

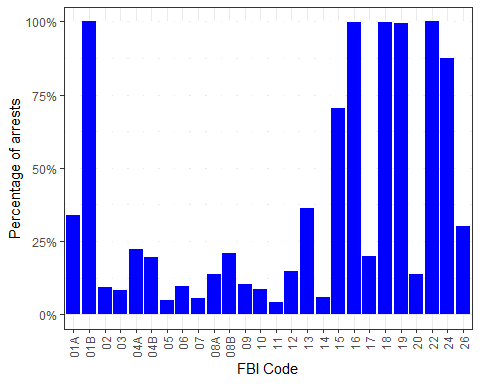
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.1213 0.1921 0.2973 0.3999 1.0000

ggplot(Location\_probs\_true,aes(x=Freq)) + geom\_histogram(bins = 20, fill = "blue") +  
 theme\_bw() +  
 labs(y = "Number of Location Description", x = 'Percentage of Arrests') +  
 scale\_x\_continuous(labels = scales::percent\_format())



Analyze FBI Code

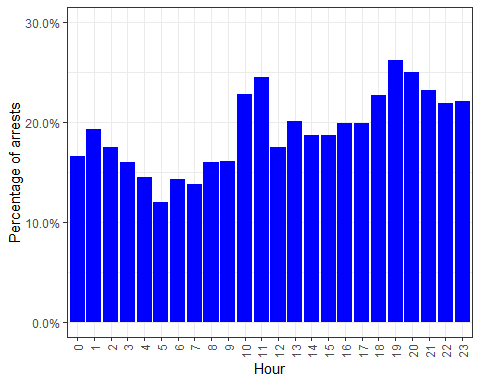
ggplot(Chicago\_crime, aes(x=`FBI Code`, fill = Arrest)) + geom\_bar(position="fill") + theme\_bw() +  
 scale\_fill\_manual("legend", values = c("TRUE" = "Blue", "FALSE" = "White")) +  
 theme(legend.position = "none") +  
 labs(y = "Percentage of arrests") +   
 theme(axis.text.x = element\_text(angle = 90,vjust =0.5, hjust=1)) +  
 scale\_y\_continuous(labels = scales::percent\_format())



Analyze Hour

ggplot(Chicago\_crime, aes(x=Hour, fill = Arrest)) + geom\_bar(position="fill") + theme\_bw() +  
 scale\_fill\_manual("legend", values = c("TRUE" = "Blue", "FALSE" = "White")) +  
 theme(legend.position = "none") +  
 labs(y = "Percentage of arrests") +   
 theme(axis.text.x = element\_text(angle = 90,vjust =0.5, hjust=1)) +  
 scale\_y\_continuous(labels = scales::percent\_format(), limits = c(0, 0.3))

## Warning: Removed 24 rows containing missing values (geom\_bar).



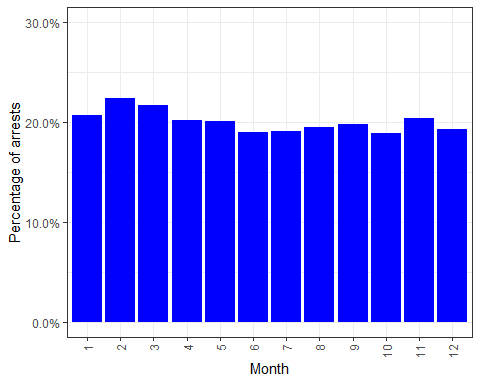
t8 = table(Chicago\_crime$Arrest, Chicago\_crime$Hour) #create a table object  
hour\_prop\_table = as.data.frame(prop.table(t8, margin = 2 )) %>% filter(Var1 == "TRUE")  
summary(hour\_prop\_table$Freq)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.1201 0.1610 0.1900 0.1912 0.2224 0.2616

Analyze Month

ggplot(Chicago\_crime, aes(x=Month, fill = Arrest)) + geom\_bar(position="fill") +  
 theme\_bw() +  
 scale\_fill\_manual("legend", values = c("TRUE" = "Blue", "FALSE" = "White")) +  
 theme(legend.position = "none") +  
 labs(y = "Percentage of arrests") +   
 theme(axis.text.x = element\_text(angle = 90,vjust =0.5, hjust=1)) +  
 scale\_y\_continuous(labels = scales::percent\_format(), limits = c(0, 0.3))

## Warning: Removed 12 rows containing missing values (geom\_bar).



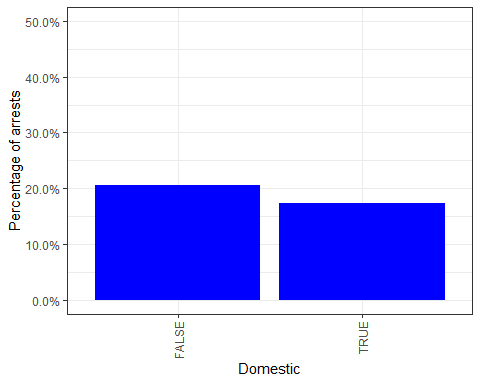
t9 = table(Chicago\_crime$Arrest, Chicago\_crime$Month) #create a table object  
month\_prop\_table = as.data.frame(prop.table(t9, margin = 2 )) %>% filter(Var1 == "TRUE")  
summary(month\_prop\_table$Freq)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.1892 0.1922 0.1992 0.2007 0.2043 0.2241

Analyze Domestic

ggplot(Chicago\_crime, aes(x=Domestic, fill = Arrest)) + geom\_bar(position="fill") + theme\_bw() +  
 scale\_fill\_manual("legend", values = c("TRUE" = "Blue", "FALSE" = "White")) +  
 theme(legend.position = "none") +  
 labs(y = "Percentage of arrests") +   
 theme(axis.text.x = element\_text(angle = 90,vjust =0.5, hjust=1)) +  
 scale\_y\_continuous(labels = scales::percent\_format(), limits = c(0, 0.5))

## Warning: Removed 2 rows containing missing values (geom\_bar).



t10 = table(Chicago\_crime$Arrest, Chicago\_crime$Domestic) #create a table object  
domestic\_prop\_table = as.data.frame(prop.table(t10, margin = 2 )) %>% filter(Var1 == "TRUE")  
summary(domestic\_prop\_table$Freq)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.1738 0.1816 0.1894 0.1894 0.1972 0.2050

What words occur in location descriptions?

loc\_desc\_words <- data.frame(x = unlist(str\_split(Chicago\_crime$`Location Description`, " ")))  
loc\_desc\_total <- setDT(loc\_desc\_words)[, .(greq =.N), x]  
loc\_desc\_total[order(-greq)]

## x greq  
## 1: STREET 58900  
## 2: RESIDENCE 49539  
## 3: APARTMENT 35248  
## 4: SIDEWALK 21098  
## 5: STORE 19375  
## ---   
## 182: RIVER 1  
## 183: DRIVE/PROP. 1  
## 184: BARBER 1  
## 185: SHOP/BEAUTY 1  
## 186: SALON 1

Chicago\_crime$In\_street <- (grepl("STREET", Chicago\_crime$`Location Description`) | grepl("SIDEWALK", Chicago\_crime$`Location Description`))  
  
Chicago\_crime$In\_residence <- (grepl("RESIDENCE", Chicago\_crime$`Location Description`) |  
 grepl("HOME", Chicago\_crime$`Location Description`) |  
 grepl("APARTMENT", Chicago\_crime$`Location Description`))  
  
Chicago\_crime$In\_business <- (grepl("STORE", Chicago\_crime$`Location Description`) |   
 grepl("RESTAURANT", Chicago\_crime$`Location Description`) |  
 grepl("GAS", Chicago\_crime$`Location Description`) |  
 grepl("GROCERY", Chicago\_crime$`Location Description`) |  
 grepl("TAVERN", Chicago\_crime$`Location Description`) |  
 grepl("RETAIL", Chicago\_crime$`Location Description`))  
   
Chicago\_crime = Chicago\_crime %>% mutate(In\_street = as.factor(In\_street)) %>%  
 mutate(In\_residence = as.factor(In\_residence)) %>%  
 mutate(In\_business = as.factor(In\_business)) %>%  
 na.omit(Chicago\_crime)

What words occur in descriptions?

desc\_words <- data.frame(x = unlist(str\_split(Chicago\_crime$`Description`, " ")))  
desc\_total <- setDT(desc\_words)[, .(greq =.N), x]  
desc\_total[order(-greq)]

## x greq  
## 1: SIMPLE 53746  
## 2: $500 39274  
## 3: TO 34358  
## 4: DOMESTIC 26702  
## 5: UNDER 25494  
## ---   
## 428: PATRONIZE 1  
## 429: INTOXICATING 1  
## 430: COMPOUNDS 1  
## 431: HIGHER 1  
## 432: EDUCATION 1

Trim the data

Chicago\_crime\_trimmed <- Chicago\_crime %>% select(Arrest, Domestic, District, `FBI Code`, Month, Hour, In\_street, In\_business, In\_residence)  
str(Chicago\_crime\_trimmed)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 262178 obs. of 9 variables:  
## $ Arrest : Factor w/ 2 levels "FALSE","TRUE": 1 2 1 2 1 1 1 1 1 1 ...  
## $ Domestic : Factor w/ 2 levels "FALSE","TRUE": 1 1 2 2 2 2 1 1 2 1 ...  
## $ District : Factor w/ 23 levels "001","002","003",..: 12 7 22 13 18 4 20 2 8 9 ...  
## $ FBI Code : Factor w/ 26 levels "01A","01B","02",..: 3 20 20 3 20 20 17 9 11 14 ...  
## $ Month : Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ Hour : Factor w/ 24 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ In\_street : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...  
## $ In\_business : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...  
## $ In\_residence: Factor w/ 2 levels "FALSE","TRUE": 1 2 2 2 1 2 1 1 2 1 ...  
## - attr(\*, "na.action")= 'omit' Named int 1 2 3 4 5 6 9 10 11 13 ...  
## ..- attr(\*, "names")= chr "1" "2" "3" "4" ...

Split the data

set.seed(1234)   
train.rows = createDataPartition(y = Chicago\_crime\_trimmed$Arrest, p=0.7, list = FALSE) #70% in training  
Chicago\_crime\_train = Chicago\_crime\_trimmed[train.rows,]   
Chicago\_crime\_test = Chicago\_crime\_trimmed[-train.rows,]

Mod1: All variables using glm AIC: 177677 AIC: 124279 training

#allmod = glm(Arrest ~., Chicago\_crime\_train, family = "binomial")  
#saveRDS(allmod, "allmod.rds")  
allmod = readRDS("allmod.rds")  
summary(allmod)

##   
## Call:  
## glm(formula = Arrest ~ ., family = "binomial", data = Chicago\_crime\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.0993 -0.6077 -0.3631 -0.2579 2.8847   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.830558 0.121933 -6.812 9.65e-12 \*\*\*  
## DomesticTRUE 0.095060 0.021369 4.449 8.64e-06 \*\*\*  
## District002 -0.065856 0.047493 -1.387 0.165546   
## District003 0.126963 0.045813 2.771 0.005583 \*\*   
## District004 0.299412 0.042533 7.040 1.93e-12 \*\*\*  
## District005 0.375220 0.043745 8.577 < 2e-16 \*\*\*  
## District006 0.363627 0.040697 8.935 < 2e-16 \*\*\*  
## District007 0.176978 0.043201 4.097 4.19e-05 \*\*\*  
## District008 -0.105368 0.044137 -2.387 0.016972 \*   
## District009 0.142810 0.046307 3.084 0.002042 \*\*   
## District010 0.321952 0.044798 7.187 6.63e-13 \*\*\*  
## District011 0.074877 0.042756 1.751 0.079901 .   
## District012 -0.010657 0.045866 -0.232 0.816259   
## District014 -0.110794 0.051920 -2.134 0.032847 \*   
## District015 -0.036262 0.049987 -0.725 0.468180   
## District016 -0.045728 0.053078 -0.862 0.388947   
## District017 -0.344856 0.059156 -5.830 5.56e-09 \*\*\*  
## District018 -0.042389 0.042028 -1.009 0.313173   
## District019 -0.042535 0.046709 -0.911 0.362490   
## District020 0.074204 0.062393 1.189 0.234322   
## District022 0.133646 0.050282 2.658 0.007862 \*\*   
## District024 0.034783 0.053872 0.646 0.518503   
## District025 0.057543 0.044489 1.293 0.195866   
## District031 0.203775 1.117490 0.182 0.855308   
## `FBI Code`01B 16.459281 647.414452 0.025 0.979717   
## `FBI Code`02 -1.243898 0.148198 -8.394 < 2e-16 \*\*\*  
## `FBI Code`03 -1.763129 0.118537 -14.874 < 2e-16 \*\*\*  
## `FBI Code`04A -0.438646 0.116173 -3.776 0.000160 \*\*\*  
## `FBI Code`04B -0.574943 0.115153 -4.993 5.95e-07 \*\*\*  
## `FBI Code`05 -2.037101 0.122646 -16.610 < 2e-16 \*\*\*  
## `FBI Code`06 -1.760172 0.111655 -15.764 < 2e-16 \*\*\*  
## `FBI Code`07 -2.119135 0.122205 -17.341 < 2e-16 \*\*\*  
## `FBI Code`08A -1.104842 0.113936 -9.697 < 2e-16 \*\*\*  
## `FBI Code`08B -0.474737 0.111192 -4.270 1.96e-05 \*\*\*  
## `FBI Code`09 -1.310124 0.232924 -5.625 1.86e-08 \*\*\*  
## `FBI Code`10 -1.915835 0.159997 -11.974 < 2e-16 \*\*\*  
## `FBI Code`11 -2.375303 0.119990 -19.796 < 2e-16 \*\*\*  
## `FBI Code`12 -0.492910 0.593829 -0.830 0.406508   
## `FBI Code`13 -0.087902 0.303814 -0.289 0.772329   
## `FBI Code`14 -1.901131 0.113961 -16.682 < 2e-16 \*\*\*  
## `FBI Code`15 1.654861 0.115377 14.343 < 2e-16 \*\*\*  
## `FBI Code`16 16.359169 64.334435 0.254 0.799277   
## `FBI Code`17 -0.398553 0.135340 -2.945 0.003231 \*\*   
## `FBI Code`18 7.957930 0.393807 20.208 < 2e-16 \*\*\*  
## `FBI Code`19 16.353424 124.486026 0.131 0.895485   
## `FBI Code`20 -0.802582 0.138184 -5.808 6.32e-09 \*\*\*  
## `FBI Code`22 15.522580 104.055172 0.149 0.881414   
## `FBI Code`24 2.748784 0.134274 20.472 < 2e-16 \*\*\*  
## `FBI Code`26 0.011937 0.111407 0.107 0.914670   
## Month2 0.111481 0.037481 2.974 0.002937 \*\*   
## Month3 -0.004365 0.036286 -0.120 0.904261   
## Month4 -0.096608 0.036742 -2.629 0.008554 \*\*   
## Month5 -0.075732 0.035158 -2.154 0.031238 \*   
## Month6 -0.105930 0.035612 -2.975 0.002934 \*\*   
## Month7 -0.129798 0.035356 -3.671 0.000241 \*\*\*  
## Month8 -0.074046 0.035153 -2.106 0.035169 \*   
## Month9 -0.120660 0.036171 -3.336 0.000851 \*\*\*  
## Month10 -0.099822 0.036200 -2.757 0.005825 \*\*   
## Month11 -0.055583 0.037049 -1.500 0.133550   
## Month12 -0.065306 0.036820 -1.774 0.076118 .   
## Hour1 0.037836 0.052407 0.722 0.470313   
## Hour2 -0.032772 0.056293 -0.582 0.560457   
## Hour3 -0.015732 0.059458 -0.265 0.791324   
## Hour4 -0.087805 0.066063 -1.329 0.183809   
## Hour5 -0.215171 0.071869 -2.994 0.002754 \*\*   
## Hour6 0.002986 0.066256 0.045 0.964053   
## Hour7 -0.069764 0.059479 -1.173 0.240826   
## Hour8 -0.094899 0.053685 -1.768 0.077114 .   
## Hour9 -0.159477 0.051013 -3.126 0.001771 \*\*   
## Hour10 0.017828 0.049097 0.363 0.716523   
## Hour11 0.044862 0.047681 0.941 0.346769   
## Hour12 -0.160190 0.047074 -3.403 0.000667 \*\*\*  
## Hour13 -0.083340 0.048167 -1.730 0.083592 .   
## Hour14 -0.117894 0.047544 -2.480 0.013150 \*   
## Hour15 -0.039914 0.046040 -0.867 0.385976   
## Hour16 0.029513 0.045894 0.643 0.520182   
## Hour17 0.055407 0.045734 1.212 0.225695   
## Hour18 0.022864 0.045822 0.499 0.617791   
## Hour19 0.090862 0.045455 1.999 0.045614 \*   
## Hour20 0.065109 0.046259 1.407 0.159287   
## Hour21 0.028407 0.047566 0.597 0.550361   
## Hour22 0.141825 0.046801 3.030 0.002442 \*\*   
## Hour23 0.106193 0.048689 2.181 0.029180 \*   
## In\_streetTRUE -0.011820 0.021263 -0.556 0.578287   
## In\_businessTRUE 1.230526 0.023709 51.900 < 2e-16 \*\*\*  
## In\_residenceTRUE -0.434142 0.022128 -19.620 < 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: 184115 on 183524 degrees of freedom  
## Residual deviance: 125363 on 183439 degrees of freedom  
## AIC: 125535  
##   
## Number of Fisher Scoring iterations: 14

Create the emptymod

emptymod = glm(Arrest~1, Chicago\_crime\_train, family = "binomial") #use ~1 to build an empty model  
summary(emptymod)

##   
## Call:  
## glm(formula = Arrest ~ 1, family = "binomial", data = Chicago\_crime\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.6697 -0.6697 -0.6697 -0.6697 1.7917   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.380854 0.005826 -237 <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: 184115 on 183524 degrees of freedom  
## Residual deviance: 184115 on 183524 degrees of freedom  
## AIC: 184117  
##   
## Number of Fisher Scoring iterations: 4

Forward

# forwardmod = stepAIC(emptymod, direction = "forward", scope=list(upper=allmod,lower=emptymod), trace = TRUE)  
# saveRDS(forwardmod, "forwardmod.rds")  
forwardmod = readRDS("forwardmod.rds")

Backward Stepwise

# backmod = stepAIC(allmod, direction = "backward", trace = TRUE)   
# saveRDS(backmod, "backmod.rds")  
backmod = readRDS("backmod.rds")  
summary(backmod)

##   
## Call:  
## glm(formula = Arrest ~ Domestic + District + `FBI Code` + Month +   
## Hour + In\_business + In\_residence, family = "binomial", data = Chicago\_crime\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.0988 -0.6077 -0.3632 -0.2579 2.8842   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.836324 0.121490 -6.884 5.82e-12 \*\*\*  
## DomesticTRUE 0.094703 0.021358 4.434 9.25e-06 \*\*\*  
## District002 -0.066966 0.047451 -1.411 0.158164   
## District003 0.125353 0.045723 2.742 0.006114 \*\*   
## District004 0.298109 0.042469 7.019 2.23e-12 \*\*\*  
## District005 0.374040 0.043695 8.560 < 2e-16 \*\*\*  
## District006 0.362297 0.040628 8.917 < 2e-16 \*\*\*  
## District007 0.175321 0.043099 4.068 4.74e-05 \*\*\*  
## District008 -0.106238 0.044109 -2.409 0.016017 \*   
## District009 0.141855 0.046276 3.065 0.002173 \*\*   
## District010 0.320504 0.044723 7.166 7.70e-13 \*\*\*  
## District011 0.073275 0.042660 1.718 0.085864 .   
## District012 -0.011600 0.045835 -0.253 0.800209   
## District014 -0.112034 0.051872 -2.160 0.030786 \*   
## District015 -0.037832 0.049908 -0.758 0.448426   
## District016 -0.046018 0.053074 -0.867 0.385920   
## District017 -0.345950 0.059123 -5.851 4.88e-09 \*\*\*  
## District018 -0.043197 0.042003 -1.028 0.303746   
## District019 -0.043444 0.046680 -0.931 0.352016   
## District020 0.073626 0.062384 1.180 0.237914   
## District022 0.132678 0.050253 2.640 0.008286 \*\*   
## District024 0.033678 0.053836 0.626 0.531596   
## District025 0.056352 0.044438 1.268 0.204762   
## District031 0.202716 1.117483 0.181 0.856050   
## `FBI Code`01B 16.454246 647.417045 0.025 0.979724   
## `FBI Code`02 -1.241460 0.148130 -8.381 < 2e-16 \*\*\*  
## `FBI Code`03 -1.764080 0.118521 -14.884 < 2e-16 \*\*\*  
## `FBI Code`04A -0.439122 0.116168 -3.780 0.000157 \*\*\*  
## `FBI Code`04B -0.575359 0.115148 -4.997 5.83e-07 \*\*\*  
## `FBI Code`05 -2.035848 0.122624 -16.602 < 2e-16 \*\*\*  
## `FBI Code`06 -1.759667 0.111649 -15.761 < 2e-16 \*\*\*  
## `FBI Code`07 -2.121243 0.122144 -17.367 < 2e-16 \*\*\*  
## `FBI Code`08A -1.103953 0.113922 -9.690 < 2e-16 \*\*\*  
## `FBI Code`08B -0.474105 0.111184 -4.264 2.01e-05 \*\*\*  
## `FBI Code`09 -1.306676 0.232842 -5.612 2.00e-08 \*\*\*  
## `FBI Code`10 -1.913914 0.159965 -11.965 < 2e-16 \*\*\*  
## `FBI Code`11 -2.373840 0.119961 -19.788 < 2e-16 \*\*\*  
## `FBI Code`12 -0.488253 0.593858 -0.822 0.410980   
## `FBI Code`13 -0.083777 0.303742 -0.276 0.782689   
## `FBI Code`14 -1.901088 0.113959 -16.682 < 2e-16 \*\*\*  
## `FBI Code`15 1.654104 0.115367 14.338 < 2e-16 \*\*\*  
## `FBI Code`16 16.355353 64.333686 0.254 0.799320   
## `FBI Code`17 -0.396440 0.135284 -2.930 0.003385 \*\*   
## `FBI Code`18 7.956862 0.393801 20.205 < 2e-16 \*\*\*  
## `FBI Code`19 16.349763 124.489952 0.131 0.895511   
## `FBI Code`20 -0.801496 0.138168 -5.801 6.60e-09 \*\*\*  
## `FBI Code`22 15.524006 104.048609 0.149 0.881396   
## `FBI Code`24 2.747330 0.134246 20.465 < 2e-16 \*\*\*  
## `FBI Code`26 0.012875 0.111392 0.116 0.907984   
## Month2 0.111566 0.037481 2.977 0.002914 \*\*   
## Month3 -0.004342 0.036286 -0.120 0.904761   
## Month4 -0.096573 0.036741 -2.628 0.008577 \*\*   
## Month5 -0.075763 0.035158 -2.155 0.031167 \*   
## Month6 -0.106078 0.035611 -2.979 0.002894 \*\*   
## Month7 -0.129971 0.035355 -3.676 0.000237 \*\*\*  
## Month8 -0.074303 0.035149 -2.114 0.034521 \*   
## Month9 -0.120870 0.036169 -3.342 0.000832 \*\*\*  
## Month10 -0.099901 0.036200 -2.760 0.005785 \*\*   
## Month11 -0.055607 0.037049 -1.501 0.133375   
## Month12 -0.065384 0.036819 -1.776 0.075767 .   
## Hour1 0.037623 0.052404 0.718 0.472796   
## Hour2 -0.032881 0.056292 -0.584 0.559145   
## Hour3 -0.015846 0.059457 -0.267 0.789848   
## Hour4 -0.087793 0.066062 -1.329 0.183863   
## Hour5 -0.214845 0.071865 -2.990 0.002794 \*\*   
## Hour6 0.003408 0.066251 0.051 0.958979   
## Hour7 -0.069247 0.059470 -1.164 0.244265   
## Hour8 -0.094298 0.053674 -1.757 0.078938 .   
## Hour9 -0.158705 0.050994 -3.112 0.001857 \*\*   
## Hour10 0.018553 0.049080 0.378 0.705414   
## Hour11 0.045661 0.047659 0.958 0.338030   
## Hour12 -0.159505 0.047058 -3.390 0.000700 \*\*\*  
## Hour13 -0.082627 0.048150 -1.716 0.086156 .   
## Hour14 -0.117363 0.047535 -2.469 0.013549 \*   
## Hour15 -0.039438 0.046031 -0.857 0.391571   
## Hour16 0.029847 0.045889 0.650 0.515431   
## Hour17 0.055610 0.045732 1.216 0.223982   
## Hour18 0.022922 0.045822 0.500 0.616896   
## Hour19 0.090817 0.045455 1.998 0.045721 \*   
## Hour20 0.065012 0.046259 1.405 0.159903   
## Hour21 0.028267 0.047564 0.594 0.552325   
## Hour22 0.141621 0.046799 3.026 0.002477 \*\*   
## Hour23 0.106035 0.048688 2.178 0.029417 \*   
## In\_businessTRUE 1.236462 0.021180 58.378 < 2e-16 \*\*\*  
## In\_residenceTRUE -0.427811 0.018980 -22.540 < 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: 184115 on 183524 degrees of freedom  
## Residual deviance: 125364 on 183440 degrees of freedom  
## AIC: 125534  
##   
## Number of Fisher Scoring iterations: 14

K-Fold with training data

ctrl = trainControl(method = "cv",number = 10)   
set.seed(1234)  
# modkFold = train(Arrest ~., Chicago\_crime\_train, method = "glm", trControl = ctrl)  
# saveRDS(modkFold, "modkFold.rds")  
modkFold = readRDS("modkFold.rds")  
summary(modkFold)

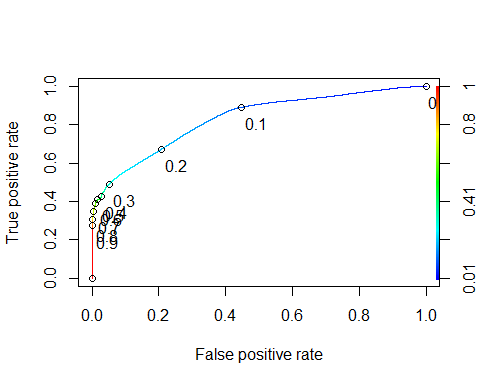
##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.0993 -0.6077 -0.3631 -0.2579 2.8847   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.830558 0.121933 -6.812 9.65e-12 \*\*\*  
## DomesticTRUE 0.095060 0.021369 4.449 8.64e-06 \*\*\*  
## District002 -0.065856 0.047493 -1.387 0.165546   
## District003 0.126963 0.045813 2.771 0.005583 \*\*   
## District004 0.299412 0.042533 7.040 1.93e-12 \*\*\*  
## District005 0.375220 0.043745 8.577 < 2e-16 \*\*\*  
## District006 0.363627 0.040697 8.935 < 2e-16 \*\*\*  
## District007 0.176978 0.043201 4.097 4.19e-05 \*\*\*  
## District008 -0.105368 0.044137 -2.387 0.016972 \*   
## District009 0.142810 0.046307 3.084 0.002042 \*\*   
## District010 0.321952 0.044798 7.187 6.63e-13 \*\*\*  
## District011 0.074877 0.042756 1.751 0.079901 .   
## District012 -0.010657 0.045866 -0.232 0.816259   
## District014 -0.110794 0.051920 -2.134 0.032847 \*   
## District015 -0.036262 0.049987 -0.725 0.468180   
## District016 -0.045728 0.053078 -0.862 0.388947   
## District017 -0.344856 0.059156 -5.830 5.56e-09 \*\*\*  
## District018 -0.042389 0.042028 -1.009 0.313173   
## District019 -0.042535 0.046709 -0.911 0.362490   
## District020 0.074204 0.062393 1.189 0.234322   
## District022 0.133646 0.050282 2.658 0.007862 \*\*   
## District024 0.034783 0.053872 0.646 0.518503   
## District025 0.057543 0.044489 1.293 0.195866   
## District031 0.203775 1.117490 0.182 0.855308   
## `\\`FBI Code\\`01B` 16.459281 647.414452 0.025 0.979717   
## `\\`FBI Code\\`02` -1.243898 0.148198 -8.394 < 2e-16 \*\*\*  
## `\\`FBI Code\\`03` -1.763129 0.118537 -14.874 < 2e-16 \*\*\*  
## `\\`FBI Code\\`04A` -0.438646 0.116173 -3.776 0.000160 \*\*\*  
## `\\`FBI Code\\`04B` -0.574943 0.115153 -4.993 5.95e-07 \*\*\*  
## `\\`FBI Code\\`05` -2.037101 0.122646 -16.610 < 2e-16 \*\*\*  
## `\\`FBI Code\\`06` -1.760172 0.111655 -15.764 < 2e-16 \*\*\*  
## `\\`FBI Code\\`07` -2.119135 0.122205 -17.341 < 2e-16 \*\*\*  
## `\\`FBI Code\\`08A` -1.104842 0.113936 -9.697 < 2e-16 \*\*\*  
## `\\`FBI Code\\`08B` -0.474737 0.111192 -4.270 1.96e-05 \*\*\*  
## `\\`FBI Code\\`09` -1.310124 0.232924 -5.625 1.86e-08 \*\*\*  
## `\\`FBI Code\\`10` -1.915835 0.159997 -11.974 < 2e-16 \*\*\*  
## `\\`FBI Code\\`11` -2.375303 0.119990 -19.796 < 2e-16 \*\*\*  
## `\\`FBI Code\\`12` -0.492910 0.593829 -0.830 0.406508   
## `\\`FBI Code\\`13` -0.087902 0.303814 -0.289 0.772329   
## `\\`FBI Code\\`14` -1.901131 0.113961 -16.682 < 2e-16 \*\*\*  
## `\\`FBI Code\\`15` 1.654861 0.115377 14.343 < 2e-16 \*\*\*  
## `\\`FBI Code\\`16` 16.359169 64.334435 0.254 0.799277   
## `\\`FBI Code\\`17` -0.398553 0.135340 -2.945 0.003231 \*\*   
## `\\`FBI Code\\`18` 7.957930 0.393807 20.208 < 2e-16 \*\*\*  
## `\\`FBI Code\\`19` 16.353424 124.486026 0.131 0.895485   
## `\\`FBI Code\\`20` -0.802582 0.138184 -5.808 6.32e-09 \*\*\*  
## `\\`FBI Code\\`22` 15.522580 104.055172 0.149 0.881414   
## `\\`FBI Code\\`24` 2.748784 0.134274 20.472 < 2e-16 \*\*\*  
## `\\`FBI Code\\`26` 0.011937 0.111407 0.107 0.914670   
## Month2 0.111481 0.037481 2.974 0.002937 \*\*   
## Month3 -0.004365 0.036286 -0.120 0.904261   
## Month4 -0.096608 0.036742 -2.629 0.008554 \*\*   
## Month5 -0.075732 0.035158 -2.154 0.031238 \*   
## Month6 -0.105930 0.035612 -2.975 0.002934 \*\*   
## Month7 -0.129798 0.035356 -3.671 0.000241 \*\*\*  
## Month8 -0.074046 0.035153 -2.106 0.035169 \*   
## Month9 -0.120660 0.036171 -3.336 0.000851 \*\*\*  
## Month10 -0.099822 0.036200 -2.757 0.005825 \*\*   
## Month11 -0.055583 0.037049 -1.500 0.133550   
## Month12 -0.065306 0.036820 -1.774 0.076118 .   
## Hour1 0.037836 0.052407 0.722 0.470313   
## Hour2 -0.032772 0.056293 -0.582 0.560457   
## Hour3 -0.015732 0.059458 -0.265 0.791324   
## Hour4 -0.087805 0.066063 -1.329 0.183809   
## Hour5 -0.215171 0.071869 -2.994 0.002754 \*\*   
## Hour6 0.002986 0.066256 0.045 0.964053   
## Hour7 -0.069764 0.059479 -1.173 0.240826   
## Hour8 -0.094899 0.053685 -1.768 0.077114 .   
## Hour9 -0.159477 0.051013 -3.126 0.001771 \*\*   
## Hour10 0.017828 0.049097 0.363 0.716523   
## Hour11 0.044862 0.047681 0.941 0.346769   
## Hour12 -0.160190 0.047074 -3.403 0.000667 \*\*\*  
## Hour13 -0.083340 0.048167 -1.730 0.083592 .   
## Hour14 -0.117894 0.047544 -2.480 0.013150 \*   
## Hour15 -0.039914 0.046040 -0.867 0.385976   
## Hour16 0.029513 0.045894 0.643 0.520182   
## Hour17 0.055407 0.045734 1.212 0.225695   
## Hour18 0.022864 0.045822 0.499 0.617791   
## Hour19 0.090862 0.045455 1.999 0.045614 \*   
## Hour20 0.065109 0.046259 1.407 0.159287   
## Hour21 0.028407 0.047566 0.597 0.550361   
## Hour22 0.141825 0.046801 3.030 0.002442 \*\*   
## Hour23 0.106193 0.048689 2.181 0.029180 \*   
## In\_streetTRUE -0.011820 0.021263 -0.556 0.578287   
## In\_businessTRUE 1.230526 0.023709 51.900 < 2e-16 \*\*\*  
## In\_residenceTRUE -0.434142 0.022128 -19.620 < 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: 184115 on 183524 degrees of freedom  
## Residual deviance: 125363 on 183439 degrees of freedom  
## AIC: 125535  
##   
## Number of Fisher Scoring iterations: 14

Logistic Threshold

predictions = predict(modkFold, type="prob")[,2] #develop predicted probabilities  
head(predictions)

## [1] 0.11055216 0.18084247 0.07416989 0.23566662 0.04672834 0.14807930

ROCRpred = prediction(predictions, Chicago\_crime\_train$Arrest)   
ROCRperf = performance(ROCRpred, "tpr", "fpr")  
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))



as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.8297931

Threshold balance

opt.cut = function(perf, pred){  
 cut.ind = mapply(FUN=function(x, y, p){  
 d = (x - 0)^2 + (y-1)^2  
 ind = which(d == min(d))  
 c(sensitivity = y[[ind]], specificity = 1-x[[ind]],   
 cutoff = p[[ind]])  
 }, perf@x.values, perf@y.values, pred@cutoffs)  
}  
print(opt.cut(ROCRperf, ROCRpred))

## [,1]  
## sensitivity 0.7461820  
## specificity 0.7266603  
## cutoff 0.1812335

Test Thresholds and evaluate accuracy

t1 = table(Chicago\_crime\_train$Arrest,predictions > 0.1812335)  
t1

##   
## FALSE TRUE  
## FALSE 106572 40088  
## TRUE 9358 27507

(t1[1,1]+t1[2,2])/nrow(Chicago\_crime\_train)

## [1] 0.7305762

Find an optimal one

t1 = table(Chicago\_crime\_train$Arrest,predictions > 0.1)  
t1

##   
## FALSE TRUE  
## FALSE 81070 65590  
## TRUE 3994 32871

(t1[1,1]+t1[2,2])/nrow(Chicago\_crime\_train)

## [1] 0.6208473

t1 = table(Chicago\_crime\_train$Arrest,predictions > 0.2)  
t1

##   
## FALSE TRUE  
## FALSE 116232 30428  
## TRUE 12063 24802

(t1[1,1]+t1[2,2])/nrow(Chicago\_crime\_train)

## [1] 0.768473

t1 = table(Chicago\_crime\_train$Arrest,predictions > 0.3)  
t1

##   
## FALSE TRUE  
## FALSE 139167 7493  
## TRUE 18794 18071

(t1[1,1]+t1[2,2])/nrow(Chicago\_crime\_train)

## [1] 0.8567661

t1 = table(Chicago\_crime\_train$Arrest,predictions > 0.4)  
t1

##   
## FALSE TRUE  
## FALSE 142646 4014  
## TRUE 21202 15663

(t1[1,1]+t1[2,2])/nrow(Chicago\_crime\_train)

## [1] 0.8626018

t1 = table(Chicago\_crime\_train$Arrest,predictions > 0.5)  
t1

##   
## FALSE TRUE  
## FALSE 144070 2590  
## TRUE 21767 15098

(t1[1,1]+t1[2,2])/nrow(Chicago\_crime\_train)

## [1] 0.8672824

t1 = table(Chicago\_crime\_train$Arrest,predictions > 0.6)  
t1

##   
## FALSE TRUE  
## FALSE 145069 1591  
## TRUE 22422 14443

(t1[1,1]+t1[2,2])/nrow(Chicago\_crime\_train)

## [1] 0.8691568

t1 = table(Chicago\_crime\_train$Arrest,predictions > 0.7)  
t1

##   
## FALSE TRUE  
## FALSE 145917 743  
## TRUE 23963 12902

(t1[1,1]+t1[2,2])/nrow(Chicago\_crime\_train)

## [1] 0.8653807

t1 = table(Chicago\_crime\_train$Arrest,predictions > 0.8)  
t1

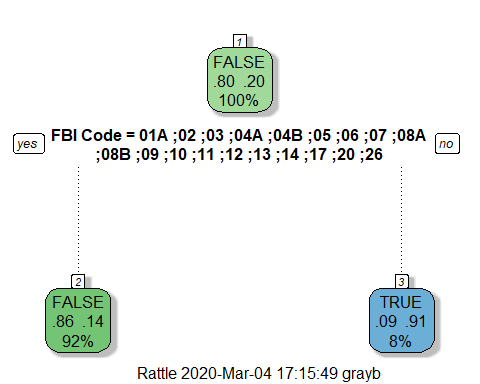
##   
## FALSE TRUE  
## FALSE 146466 194  
## TRUE 25626 11239

(t1[1,1]+t1[2,2])/nrow(Chicago\_crime\_train)

## [1] 0.8593107

Decision Tree

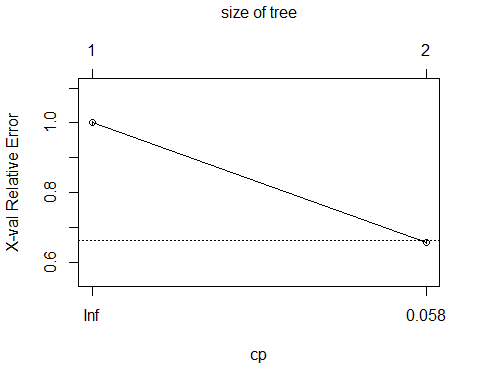
tree1 = rpart(Arrest ~., Chicago\_crime\_train, method="class")  
  
split.fun <- function(x, labs, digits, varlen, faclen)  
{  
 # replace commas with spaces (needed for strwrap)  
 labs <- gsub(",", " ;", labs)  
 for(i in 1:length(labs)) {  
 # split labs[i] into multiple lines  
 labs[i] <- paste(strwrap(labs[i], width=50), collapse="\n")  
 }  
 labs  
}  
  
fancyRpartPlot(tree1, cex=1.0, split.fun=split.fun)



printcp(tree1)

##   
## Classification tree:  
## rpart(formula = Arrest ~ ., data = Chicago\_crime\_train, method = "class")  
##   
## Variables actually used in tree construction:  
## [1] FBI Code  
##   
## Root node error: 36865/183525 = 0.20087  
##   
## n= 183525   
##   
## CP nsplit rel error xerror xstd  
## 1 0.34076 0 1.00000 1.00000 0.0046559  
## 2 0.01000 1 0.65924 0.65924 0.0039389

plotcp(tree1)



Confusion matrix on training data

treepred\_dt = predict(tree1, Chicago\_crime\_train, type = "class")  
head(treepred\_dt)

## 1 2 3 4 5 6   
## FALSE FALSE FALSE FALSE FALSE FALSE   
## Levels: FALSE TRUE

confusionMatrix(treepred\_dt,Chicago\_crime\_train$Arrest)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 145326 22969  
## TRUE 1334 13896  
##   
## Accuracy : 0.8676   
## 95% CI : (0.866, 0.8691)  
## No Information Rate : 0.7991   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.4714   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9909   
## Specificity : 0.3769   
## Pos Pred Value : 0.8635   
## Neg Pred Value : 0.9124   
## Prevalence : 0.7991   
## Detection Rate : 0.7919   
## Detection Prevalence : 0.9170   
## Balanced Accuracy : 0.6839   
##   
## 'Positive' Class : FALSE   
##

Predictions on testing set

treepred\_dt\_test = predict(tree1, Chicago\_crime\_test, type = "class")  
head(treepred\_dt\_test)

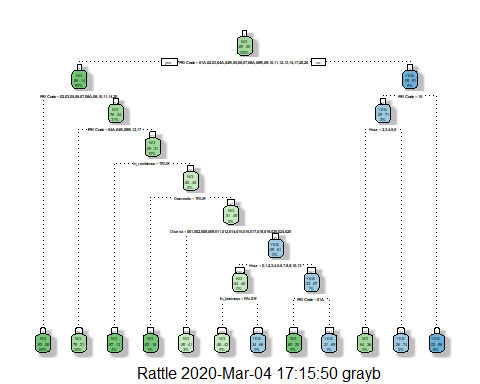
## 1 2 3 4 5 6   
## FALSE FALSE FALSE FALSE FALSE FALSE   
## Levels: FALSE TRUE

confusionMatrix(treepred\_dt\_test,Chicago\_crime\_test$Arrest,positive="TRUE")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 62248 9806  
## TRUE 606 5993  
##   
## Accuracy : 0.8676   
## 95% CI : (0.8652, 0.87)  
## No Information Rate : 0.7991   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.4727   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.3793   
## Specificity : 0.9904   
## Pos Pred Value : 0.9082   
## Neg Pred Value : 0.8639   
## Prevalence : 0.2009   
## Detection Rate : 0.0762   
## Detection Prevalence : 0.0839   
## Balanced Accuracy : 0.6848   
##   
## 'Positive' Class : TRUE   
##

Decision tree with change in cp

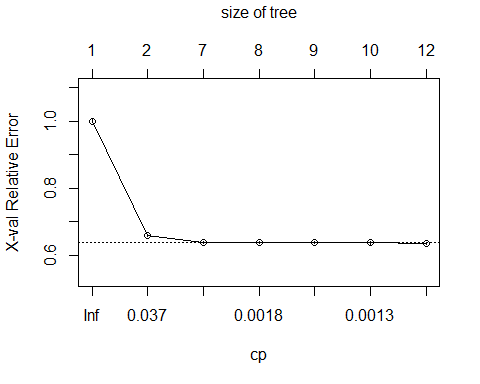
# tree2 = rpart(Arrest ~., Chicago\_crime\_train, cp = 0.001, method="class")  
# saveRDS(tree2, "tree2.rds")  
tree2 = readRDS("tree2.rds")  
fancyRpartPlot(tree2)



printcp(tree2)

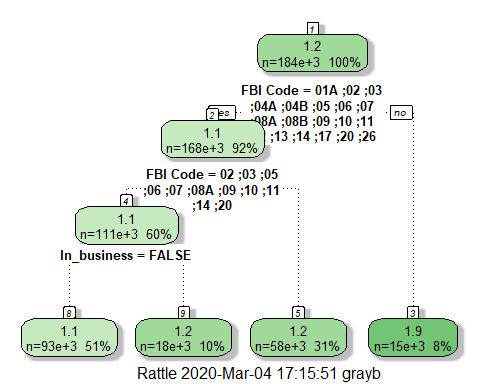
##   
## Classification tree:  
## rpart(formula = Arrest ~ ., data = Chicago\_crime\_train, method = "class",   
## cp = 0.001)  
##   
## Variables actually used in tree construction:  
## [1] District Domestic FBI Code Hour In\_business   
## [6] In\_residence  
##   
## Root node error: 36865/183525 = 0.20087  
##   
## n= 183525   
##   
## CP nsplit rel error xerror xstd  
## 1 0.3407568 0 1.00000 1.00000 0.0046559  
## 2 0.0040960 1 0.65924 0.65924 0.0039389  
## 3 0.0018988 6 0.63876 0.63817 0.0038848  
## 4 0.0017361 7 0.63686 0.63792 0.0038842  
## 5 0.0014377 8 0.63513 0.63741 0.0038828  
## 6 0.0012478 9 0.63369 0.63695 0.0038816  
## 7 0.0010000 11 0.63119 0.63456 0.0038754

plotcp(tree2)



Regression Tree

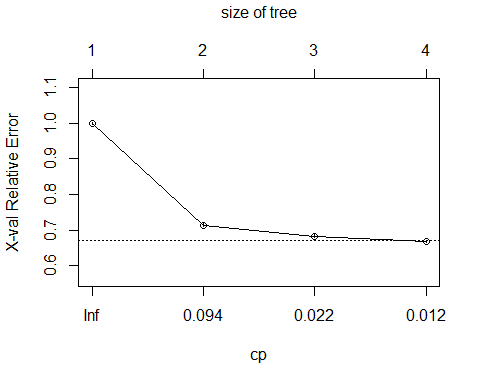
# regtree1 = rpart(Arrest~., method="anova", Chicago\_crime\_train)  
  
split.fun <- function(x, labs, digits, varlen, faclen)  
{  
 # replace commas with spaces (needed for strwrap)  
 labs <- gsub(",", " ;", labs)  
 for(i in 1:length(labs)) {  
 # split labs[i] into multiple lines  
 labs[i] <- paste(strwrap(labs[i], width=25), collapse="\n")  
 }  
 labs  
}  
# saveRDS(regtree1, "regtree1.rds")  
regtree1 = readRDS("regtree1.rds")  
fancyRpartPlot(regtree1, cex=0.8, split.fun=split.fun)



printcp(regtree1)

##   
## Regression tree:  
## rpart(formula = Arrest ~ ., data = Chicago\_crime\_train, method = "anova")  
##   
## Variables actually used in tree construction:  
## [1] FBI Code In\_business  
##   
## Root node error: 29460/183525 = 0.16052  
##   
## n= 183525   
##   
## CP nsplit rel error xerror xstd  
## 1 0.285423 0 1.00000 1.00002 0.0034856  
## 2 0.031157 1 0.71458 0.71460 0.0036136  
## 3 0.015461 2 0.68342 0.68347 0.0033928  
## 4 0.010000 3 0.66796 0.66803 0.0032446

plotcp(regtree1)



regtree1

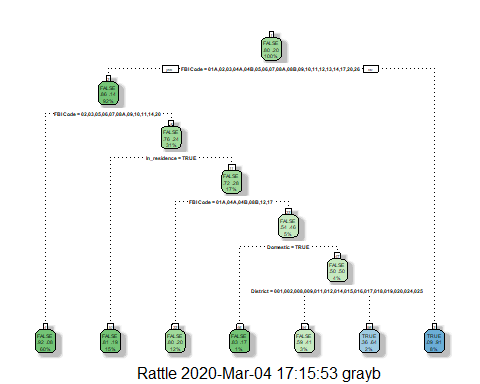
## n= 183525   
##   
## node), split, n, deviance, yval  
## \* denotes terminal node  
##   
## 1) root 183525 29459.860 1.200872   
## 2) FBI Code=01A,02,03,04A,04B,05,06,07,08A,08B,09,10,11,12,13,14,17,20,26 168295 19834.180 1.136481   
## 4) FBI Code=02,03,05,06,07,08A,09,10,11,14,20 110722 8448.068 1.083226   
## 8) In\_business=FALSE 92788 4825.024 1.055029 \*  
## 9) In\_business=TRUE 17934 3167.555 1.229118 \*  
## 5) FBI Code=01A,04A,04B,08B,12,13,17,26 57573 10468.210 1.238897 \*  
## 3) FBI Code=01B,15,16,18,19,22,24 15230 1217.155 1.912410 \*

let’s build a classification tree. Here we use caret to manage the model building.

# rpart\_fit = train(x=Chicago\_crime\_trimmed[,-1], y=Chicago\_crime\_trimmed$Arrest,  
# method = "rpart",   
# trControl = ctrl)  
  
# saveRDS(rpart\_fit, "rpart\_fit.rds")  
rpart\_fit = readRDS("rpart\_fit.rds")  
rpart\_fit

## CART   
##   
## 262178 samples  
## 8 predictor  
## 2 classes: 'FALSE', 'TRUE'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 235959, 235960, 235959, 235960, 235961, 235960, ...   
## Resampling results across tuning parameters:  
##   
## cp Accuracy Kappa   
## 0.001310193 0.8726018 0.5091772  
## 0.004845815 0.8688029 0.4825634  
## 0.340821054 0.8399596 0.2818875  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.001310193.

fancyRpartPlot(rpart\_fit$finalModel)



Random Forest

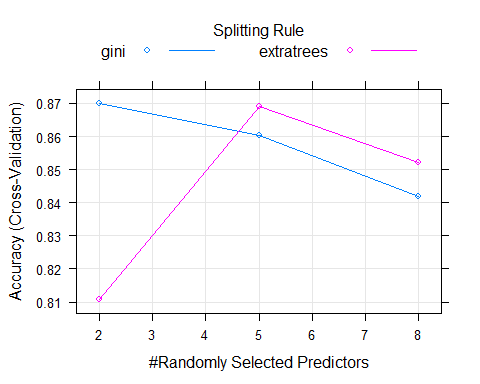
nctrl = trainControl(method = "cv",number = 5)  
# rf1\_fit = train(x=as.matrix(Chicago\_crime\_train[,-1]),  
# y=as.matrix(Chicago\_crime\_train$Arrest),  
# method = "ranger",   
# num.trees = 100,   
# importance = "permutation",  
# trControl = nctrl)  
# saveRDS(rf1\_fit, "rf1\_fit.rds")  
rf1\_fit = readRDS("rf1\_fit.rds")  
varImp(rf1\_fit)

## ranger variable importance  
##   
## Overall  
## FBI Code 100.000  
## In\_residence 11.968  
## Domestic 7.356  
## District 3.964  
## In\_street 3.489  
## In\_business 1.840  
## Hour 1.789  
## Month 0.000

rf1\_fit

## Random Forest   
##   
## 183525 samples  
## 8 predictor  
## 2 classes: 'FALSE', 'TRUE'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 146820, 146820, 146820, 146820, 146820   
## Resampling results across tuning parameters:  
##   
## mtry splitrule Accuracy Kappa   
## 2 gini 0.8700667 0.50504978  
## 2 extratrees 0.8107451 0.08844869  
## 5 gini 0.8602043 0.49798808  
## 5 extratrees 0.8689879 0.51334632  
## 8 gini 0.8419071 0.46339425  
## 8 extratrees 0.8522817 0.48243519  
##   
## Tuning parameter 'min.node.size' was held constant at a value of 1  
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were mtry = 2, splitrule = gini  
## and min.node.size = 1.

plot(rf1\_fit)



predRF = predict(rf1\_fit)  
head(predRF)

## [1] FALSE FALSE FALSE FALSE FALSE FALSE  
## Levels: FALSE TRUE

confusionMatrix(predRF, Chicago\_crime\_train$Arrest)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 144022 21118  
## TRUE 2638 15747  
##   
## Accuracy : 0.8706   
## 95% CI : (0.869, 0.8721)  
## No Information Rate : 0.7991   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5037   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9820   
## Specificity : 0.4272   
## Pos Pred Value : 0.8721   
## Neg Pred Value : 0.8565   
## Prevalence : 0.7991   
## Detection Rate : 0.7848   
## Detection Prevalence : 0.8998   
## Balanced Accuracy : 0.7046   
##   
## 'Positive' Class : FALSE   
##

predRF\_test = predict.train(rf1\_fit, newdata = Chicago\_crime\_test)  
head(predRF\_test)

## [1] FALSE FALSE FALSE FALSE FALSE FALSE  
## Levels: FALSE TRUE

confusionMatrix(predRF\_test, Chicago\_crime\_test$Arrest)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 61619 9519  
## TRUE 1235 6280  
##   
## Accuracy : 0.8633   
## 95% CI : (0.8609, 0.8657)  
## No Information Rate : 0.7991   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.4701   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9804   
## Specificity : 0.3975   
## Pos Pred Value : 0.8662   
## Neg Pred Value : 0.8357   
## Prevalence : 0.7991   
## Detection Rate : 0.7834   
## Detection Prevalence : 0.9045   
## Balanced Accuracy : 0.6889   
##   
## 'Positive' Class : FALSE   
##

Neural Network

start\_time = Sys.time() #for timing  
nnControl = trainControl(method = "cv",   
 number = 5)  
  
nnetGrid <- expand.grid(size = 8, decay = 0.1)  
  
set.seed(1234)  
# nnetBasic = train(x=as.data.frame(Chicago\_crime\_train[,-1]),   
# y=as.matrix(Chicago\_crime\_train$Arrest),  
# method = "nnet",  
# num.trees = 50,  
# tuneGrid = nnetGrid,  
# trControl = nnControl,  
# trace = FALSE)  
  
end\_time = Sys.time()  
end\_time-start\_time

## Time difference of 0.003957987 secs

# saveRDS(nnetBasic, "nnetBasic.rds")  
nnetBasic = readRDS("nnetBasic.rds")  
nnetBasic

## Neural Network   
##   
## 183525 samples  
## 8 predictor  
## 2 classes: 'FALSE', 'TRUE'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 146820, 146820, 146820, 146820, 146820   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.873712 0.5191292  
##   
## Tuning parameter 'size' was held constant at a value of 8  
## Tuning  
## parameter 'decay' was held constant at a value of 0.1

NN Predictions

predNetBasic = predict(nnetBasic, Chicago\_crime\_train)  
confusionMatrix(predNetBasic, Chicago\_crime\_train$Arrest)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 144437 21063  
## TRUE 2223 15802  
##   
## Accuracy : 0.8731   
## 95% CI : (0.8716, 0.8746)  
## No Information Rate : 0.7991   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5113   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9848   
## Specificity : 0.4286   
## Pos Pred Value : 0.8727   
## Neg Pred Value : 0.8767   
## Prevalence : 0.7991   
## Detection Rate : 0.7870   
## Detection Prevalence : 0.9018   
## Balanced Accuracy : 0.7067   
##   
## 'Positive' Class : FALSE   
##

NN Predictions on testing set

predNetBasictest = predict(nnetBasic, Chicago\_crime\_test)  
confusionMatrix(predNetBasictest, Chicago\_crime\_test$Arrest)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 61867 8976  
## TRUE 987 6823  
##   
## Accuracy : 0.8733   
## 95% CI : (0.871, 0.8756)  
## No Information Rate : 0.7991   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5133   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9843   
## Specificity : 0.4319   
## Pos Pred Value : 0.8733   
## Neg Pred Value : 0.8736   
## Prevalence : 0.7991   
## Detection Rate : 0.7866   
## Detection Prevalence : 0.9007   
## Balanced Accuracy : 0.7081   
##   
## 'Positive' Class : FALSE   
##

Ensemble model

control = trainControl(  
 method = "cv",  
 number = 5, #to save time, we'll use 5 fold cross-validation rather than 10  
 savePredictions = "final",  
 classProbs = TRUE,  
 summaryFunction = twoClassSummary #enables calculation of AUC  
 )  
  
Chicago\_crime\_trimmed = Chicago\_crime\_trimmed %>% mutate(Arrest = as\_factor(as.character(Arrest))) %>%  
 mutate(Arrest = fct\_recode(Arrest, "YES" = "TRUE", "NO" = "FALSE"))  
  
set.seed(1234)  
# model\_list = caretList(x=as.matrix(Chicago\_crime\_trimmed[,-1]),   
# y=as.matrix(Chicago\_crime\_trimmed$Arrest),   
# metric = "ROC", #specify that maximizing AUC is our objective  
# trControl=control, #using the previously defined trControl object  
# methodList=c("glm","ranger","rpart") #specifying the model methods to use  
# )  
  
# saveRDS(model\_list, "model\_list.rds")  
model\_list = readRDS("model\_list.rds")

Split data and More Ensemble Processing

set.seed(1234)   
train.rows = createDataPartition(y = Chicago\_crime\_trimmed$Arrest, p=0.7, list = FALSE) #70% in training  
Chicago\_crime\_train = Chicago\_crime\_trimmed[train.rows,]   
Chicago\_crime\_test = Chicago\_crime\_trimmed[-train.rows,]  
  
as.data.frame(predict(model\_list, newdata=head(Chicago\_crime\_trimmed)))

## glm ranger rpart  
## 1 0.8936178 0.9093758 0.9165080  
## 2 0.8081992 0.6850997 0.8107643  
## 3 0.8126885 0.8260265 0.8107643  
## 4 0.9286189 0.7928191 0.9165080  
## 5 0.7553551 0.7930353 0.7963452  
## 6 0.7778357 0.8276266 0.8107643

modelCor(resamples(model\_list))

## glm ranger rpart  
## glm 1.00000000 0.04983136 0.7956640  
## ranger 0.04983136 1.00000000 0.2842149  
## rpart 0.79566396 0.28421494 1.0000000

Assemble the Ensemble model

ensemble = caretEnsemble(  
 model\_list,   
 metric="ROC",  
 trControl=trainControl(  
 method = "cv", #cross-validation during ensembling  
 number= 5, #number of folds  
 summaryFunction=twoClassSummary,  
 classProbs=TRUE  
 ))  
summary(ensemble)

## The following models were ensembled: glm, ranger, rpart   
## They were weighted:   
## 4.1332 -4.3793 -2.3912 -0.4371  
## The resulting ROC is: 0.8368  
## The fit for each individual model on the ROC is:   
## method ROC ROCSD  
## glm 0.8286900 0.002409500  
## ranger 0.8085436 0.052719530  
## rpart 0.7884286 0.001575057

Evaluate model on training set

pred\_ensemble = predict(ensemble, Chicago\_crime\_train, type = "raw")  
confusionMatrix(pred\_ensemble,Chicago\_crime\_train$Arrest)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction NO YES  
## NO 143991 21179  
## YES 2669 15686  
##   
## Accuracy : 0.8701   
## 95% CI : (0.8685, 0.8716)  
## No Information Rate : 0.7991   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5016   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9818   
## Specificity : 0.4255   
## Pos Pred Value : 0.8718   
## Neg Pred Value : 0.8546   
## Prevalence : 0.7991   
## Detection Rate : 0.7846   
## Detection Prevalence : 0.9000   
## Balanced Accuracy : 0.7036   
##   
## 'Positive' Class : NO   
##

Evaluate model on testing set

pred\_ensemble = predict(ensemble, Chicago\_crime\_test, type = "raw")  
confusionMatrix(pred\_ensemble,Chicago\_crime\_test$Arrest)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction NO YES  
## NO 61674 9036  
## YES 1180 6763  
##   
## Accuracy : 0.8701   
## 95% CI : (0.8677, 0.8725)  
## No Information Rate : 0.7991   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5029   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9812   
## Specificity : 0.4281   
## Pos Pred Value : 0.8722   
## Neg Pred Value : 0.8514   
## Prevalence : 0.7991   
## Detection Rate : 0.7841   
## Detection Prevalence : 0.8990   
## Balanced Accuracy : 0.7046   
##   
## 'Positive' Class : NO   
##

Stacking

stack = caretStack(  
 model\_list, #use the list of models already specified  
 method ="glm", #stack models linearly  
 metric ="ROC", #maximize AUC  
 trControl = trainControl(  
 method = "cv", #k-fold cross-validation  
 number = 5, #5 folds  
 savePredictions = "final",  
 classProbs = TRUE, #save probabilities  
 summaryFunction = twoClassSummary #calculate AUC values  
 )  
)  
  
print(stack)

## A glm ensemble of 3 base models: glm, ranger, rpart  
##   
## Ensemble results:  
## Generalized Linear Model   
##   
## 262178 samples  
## 3 predictor  
## 2 classes: 'NO', 'YES'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 209742, 209743, 209743, 209742, 209742   
## Resampling results:  
##   
## ROC Sens Spec   
## 0.836783 0.9768226 0.4512571

Evaluate Stacking on training set

pred\_stack = predict(stack, Chicago\_crime\_train, type = "raw")  
confusionMatrix(pred\_stack,Chicago\_crime\_train$Arrest)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction NO YES  
## NO 143991 21179  
## YES 2669 15686  
##   
## Accuracy : 0.8701   
## 95% CI : (0.8685, 0.8716)  
## No Information Rate : 0.7991   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5016   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9818   
## Specificity : 0.4255   
## Pos Pred Value : 0.8718   
## Neg Pred Value : 0.8546   
## Prevalence : 0.7991   
## Detection Rate : 0.7846   
## Detection Prevalence : 0.9000   
## Balanced Accuracy : 0.7036   
##   
## 'Positive' Class : NO   
##

Evaluate Stacking on testing set

pred\_stack\_test = predict(stack, Chicago\_crime\_test, type = "raw")  
confusionMatrix(pred\_stack\_test,Chicago\_crime\_test$Arrest)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction NO YES  
## NO 61674 9036  
## YES 1180 6763  
##   
## Accuracy : 0.8701   
## 95% CI : (0.8677, 0.8725)  
## No Information Rate : 0.7991   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5029   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9812   
## Specificity : 0.4281   
## Pos Pred Value : 0.8722   
## Neg Pred Value : 0.8514   
## Prevalence : 0.7991   
## Detection Rate : 0.7841   
## Detection Prevalence : 0.8990   
## Balanced Accuracy : 0.7046   
##   
## 'Positive' Class : NO   
##