Capstone Champaign IL Jail Data

Terry McLaughlin

6/17/2020

Contents

1 Overview

1.1 Introduction

This document discusses my analysis of a Champaign Illinois dataset regarding arrest information. The dataset was provided by the State of Illinois. This is the last of two projects required for the Capstone class, the ninth and final course HarvardX Data Science Program.

In this project, I applied skills learned throughout the HarvardX Data Science course series to create models using a real-world dataset I downloaded from the State of Illinois. You can obtain this dataset at: https://data.illinois.gov/dataset/jail-booking-data. I wanted to find a real-world local dataset to work with. Additionally, I wanted to find a dataset that was not too large so as to limit the type of models my machine could create.

I will try to predict whether a person arrested is over 30 or under 30 depending on other variables in the dataset. I will examine the accuracy of the various models using a train and test dataset to determine the best model. Lastly, I will compare the results of the best models to the validation dataset to see the final accuracy of the model.

1.2 Dataset Summary

The dataset provided has the following dimensions.

Description	Value
Rows	67,923
Columns	38
Rows after cleaning dataset	63,982
Columns after cleaning the dataset	14 (3 of which were transformed)

Each row of data represents a booking along with an associated charge. For example, if a person was arrested for DUI, running a red light, and resisting arrest that would be 3 lines in the dataset.

2 Method and Analysis

Steps to prepare project

1. Load Libraries: Load libraries needed for project
2. Data Loading: Load initial dataset using code provided in course materials
3. Transform Dataset (in the Approach section of this document): Add and remove columns. Remove rows with null values for use in analysis

2.1 Project Setup

2.1.1 Load Libraries

The libraries listed below are needed for my project.

```
########################################
# Load libraries needed for project
########################################
library(dslabs)
library(tidyverse)
library(lubridate)
library(caret)
library(dplyr)
library(kableExtra)
library(formattable)
library(ggrepel)
library(randomForest)
library(Amelia)
library(ggplot2)
library(productplots)
library(ggmosaic)
```

2.1.2 Data Loading

The following code was used to load the dataset into my project.

```
##############################################################################################
# Download dataset
##############################################################################################
if(!require(tidyverse))
      install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret))
      install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table))
      install.packages("data.table", repos = "http://cran.us.r-project.org")

# Champaign Jail Dataset
# https://data.illinois.gov/dataset/jail-booking-data
url <- "https://data.illinois.gov/dataset/820f5916-069f-41e8-afc2-815a26770bea/resource/dfcad952-67cc-4
jaildata <- read_csv(url)
download.file(url, "chm_jail_data.csv")
tempfile()
tmp_filename <- tempfile()
download.file(url, tmp_filename)
jaildata_tbl <- read_csv(tmp_filename)
file.remove(tmp_filename)
```

```
#read file in locally (to speed up testing)
#jaildata_tbl <- read_csv("C:/Users/terry.mclaughlin/OneDrive/Documents/Capstone-2nd-Project/chm_jail_d
```

2.2 Approach / Overview

For this project, I tried several different machine algorithms presented in the course materials. My dataset was heavily categorized, so this largely determined the models I was able to use. Additionally, I determined the best model by using accuracy.

My models were divided into 3 categories and are as follows:

- Generative Models
  - Naive Bayes
  - QDA
  - LDA
  - GLM
- Ensembles
  - Combine all generative models
  - Only combine models over 70
- Tress and Forests
  - Classification Tree
  - Random Forest

I split the dataset into a train and validation set. Then I further split the train set into a train and test set. Finally, I used the models results I trained against the test set, and compared it to the validation set to ensure overall accuracy.

2.3 Data Cleanup and Transformation

There are several columns in the raw data file that we won't need. Also, we need to add a few columns necessary to normalize the data. The columns added were: (1) Over30 - indicates if the age is <=30 or >30, (2) AgeDecade - indicates decade of the age, and (3) Crime_Severity - indicates a weight of the various buckets of crime. For example, a traffic offense is a 1 whereas a violent crime would be a 8. The category other was given a 9, as terrorism, fugitive from justice, and other types of crimes were in this bucket. The other category is subjective.

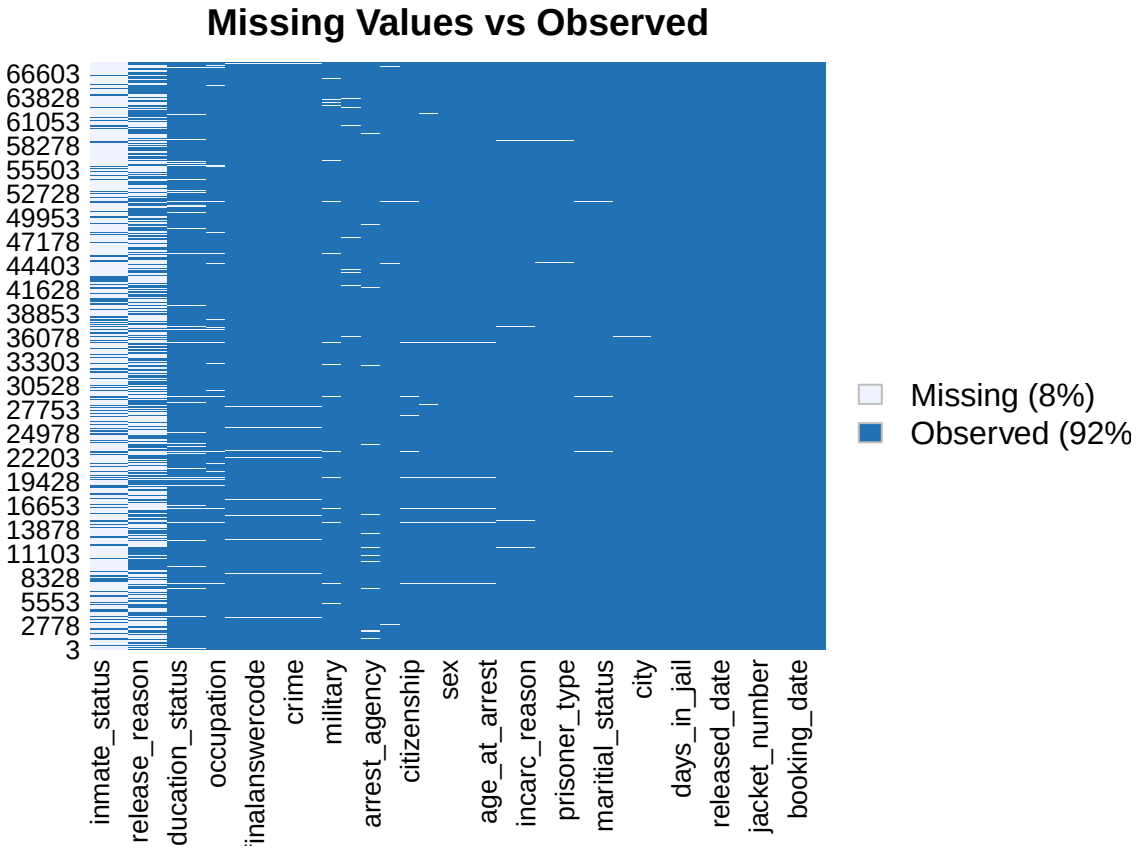
The code is below to cleanup and transform the raw data for use in our models.

```
##############################################################################################
# Clean the dataset - prepare for machine learning algorithm
##############################################################################################
#first convert jaildata to a dataframe
chmJailDataRaw <- as.data.frame(jaildata_tbl)

#first - pull out columns we want and add column for over 30,  booking year,
#and crime severity
chmJailData <- chmJailDataRaw %>% select(c(booking_date,
                                           jacket_number, age_at_arrest,
                                           marital_status, race, sex,
                                           military, prisoner_type, crime,
```

```
                                          superhighlevel)) %>%
         mutate(booking_year = as.integer(str_sub(chmJailDataRaw$booking_number,1,4)),
         over30 = factor(ifelse(age_at_arrest <= 30, "<=30", ">30")) ,
         crime_severity = ifelse(superhighlevel == "Other", 10, 5) )


#get rid of null rows for and remove the 7 rows that are prior to 2012
chmJailData <- chmJailData %>% filter(!is.na(age_at_arrest) &
                                        !is.na(superhighlevel) &
                                        !is.na(marital_status) &
                                        !is.na(race) &
                                        !is.na(sex) &
                                        !is.na(military) &
                                        !is.na(crime) &
                                        !is.na(prisoner_type)
                                      & booking_year >= 2012)


######################Block to add age category by decade   ##################
#start at 10 and end at 90 as 16 is youngest and 85 is oldest
ageDecade <- c(paste(seq(10, 85, by = 10), seq(10 + 10 - 1, 90 - 1, by = 10),
                     sep = "-"))
ageDecade

class(chmJailData$age_at_arrest)
chmJailData['age_group']  <- cut(chmJailData$age_at_arrest,
                                 breaks = c(seq(10,85, by = 10), Inf),
                                 labels = ageDecade, right = FALSE)


########Block to change the crime_severity  to an integer value  ############

########Block to change the crime_severity  to an integer value  ############
gsr <- function(Source, Search, Replace)
{if (length(Search) != length(Replace))
    stop("Search and Replace Must Have Equal Number of Items\n")

  Changed <- as.character(Source)

  for (i in 1:length(Search))
  {cat("Replacing: ", Search[i], " With: ", Replace[i], "\n")
    Changed <- replace(Changed, Changed == Search[i], Replace[i])}
  cat("\n")
  Changed}

chmJailData$crime_severity <- gsr(chmJailData$superhighlevel,
                                  c("Domestic Violence","Drug",
                                    "DUI","Other", "Property",
                                    "Public Order","Sex",
                                    "Traffic","Violent"),
                                  c("7", "5", "4", "9", "2",
                                    "3", "6", "1", "8"))
```

```r
############end block on crime severity ####################

#Change certain columns to factors or numeric - needed for machine learning model
chmJailData$marital_status <- as.factor(chmJailData$marital_status)
chmJailData$race <- as.factor(chmJailData$race)
chmJailData$sex <- as.factor(chmJailData$sex)
chmJailData$military <- as.factor(chmJailData$military)
chmJailData$superhighlevel <- as.factor(chmJailData$superhighlevel)
chmJailData$prisoner_type <- as.factor(chmJailData$prisoner_type)
chmJailData$jacket_number <- as.factor(chmJailData$jacket_number)
chmJailData$crime_severity <- as.numeric(chmJailData$crime_severity)

#############################################################################################
# Examine the dataset - General Statistics
#############################################################################################
#Pull out key info
##add as.character to control decimal points displayed
chmJailData_summary <- chmJailData %>%
  summarize(Number_Of_Rows = as.character(nrow(chmJailData)),
            Number_Of_Columns = as.character(ncol(chmJailData)),
            No_PeopleArrested = as.character(n_distinct(jacket_number)),
            No_Different_Crimes = as.character(n_distinct(crime)),
            Minimum_age = min(age_at_arrest,na.rm = TRUE),
            Maximum_age = max(age_at_arrest,na.rm = TRUE),
            Average_age = round(mean(age_at_arrest,na.rm = TRUE),2),
            First_Year_Of_Data = as.character(min(booking_year)),
            Last_Year_Of_Data = as.character(max(booking_year)),
            Percent_Males = round(sum(chmJailData$sex=='Male',na.rm = TRUE)/n()*100, 2),
            Percent_Females = round(sum(chmJailData$sex=='Female',na.rm = TRUE)/n()*100, 2))

#transpose data for better readablilty
transposeSummary <- data.frame(t(chmJailData_summary))
```

The dataset had numerous columns that I did not need. It also had missing values in columns I did need. First, using the “Amelia” package we can look at the data and determine the extent of missing values. The good news is 92% of the data provided had values in the fields we need for our models.



Let’s look at the header of our cleaned dataset that we will use to create our models.

Transformed Data Header

```
##   booking_date jacket_number age_at_arrest marital_status     race  sex
## 1     1/1/2012         22914            51       Divorced    White Male
## 2     1/1/2012         22914            51       Divorced    White Male
## 3     1/1/2012         22914            51       Divorced    White Male
## 4     1/1/2012       1024225            32         Single Hispanic Male
## 5     1/1/2012       1024225            32         Single Hispanic Male
## 6     1/1/2012       1024226            20         Single    White Male
##   military          prisoner_type
## 1     None Misdemeanor Arraignment
## 2     None Misdemeanor Arraignment
## 3     None Misdemeanor Arraignment
## 4     None      Felony Arraignment
## 5     None      Felony Arraignment
## 6     None Misdemeanor Arraignment
##                                          crime superhighlevel booking_year
## 1          DRIVING UNDER THE INFLUENCE OF ALCOHOL            DUI         2012
## 2                         OTHER TRAFFIC OFFENSES        Traffic         2012
## 3 RESISTING,OBSTRUCTING,OR DISARMING A POLICE OFC   Public Order         2012
## 4                               DOMESTIC BATTERY        Violent         2012
```

```
## 5                              AGGRAVATED BATTERY        Violent         2012
## 6 RESISTING,OBSTRUCTING,OR DISARMING A POLICE OFC   Public Order         2012
##   over30 crime_severity age_group
## 1    >30              4     50-59
## 2    >30              1     50-59
## 3    >30              3     50-59
## 4    >30              8     30-39
## 5    >30              8     30-39
## 6   <=30              3     20-29
```

Now we will view summary statistics of the cleaned data we are going to use in our modeling.

Summary Statistics

	Value
Number_Of_Rows	63982
Number_Of_Columns	14
No_PeopleArrested	18816
No_Different_Crimes	197
Minimum_age	16
Maximum_age	85
Average_age	30.7
First_Year_Of_Data	2012
Last_Year_Of_Data	2018
Percent_Males	76.92
Percent_Females	23.08

At this point we need to split the dataset into a training set and validation set. Then further split the training set into a test and train set. Here is the code.

```
#############################################################################################
# Split the dataset to create train set, test set, and a validation set
#############################################################################################
#first let's define predictor and oucome

# Validation set will be 20% of the data
set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(chmJailData$over30, times = 1, p = 0.2, list = FALSE)
jail_train <- chmJailData[-test_index,]
validation <- chmJailData[test_index,]

#Split further for a test set
#now y is the train data age
y<- jail_train$over30
#split again to obtain a test set so we can test various models -
#    need to see if we change the seed
set.seed(10, sample.kind="Rounding")
test_indexsplit <- createDataPartition(jail_train$over30, times = 1, p = 0.2, list = FALSE)
training <- jail_train[-test_indexsplit,] #train
test <- jail_train[test_indexsplit,] #test
```

3 Dataset General Statistics and Overview

Prior to building models, we need to examine and familiarize ourselves with the data. First we will view overall summary charts, then drill down into charts displaying demographic information.

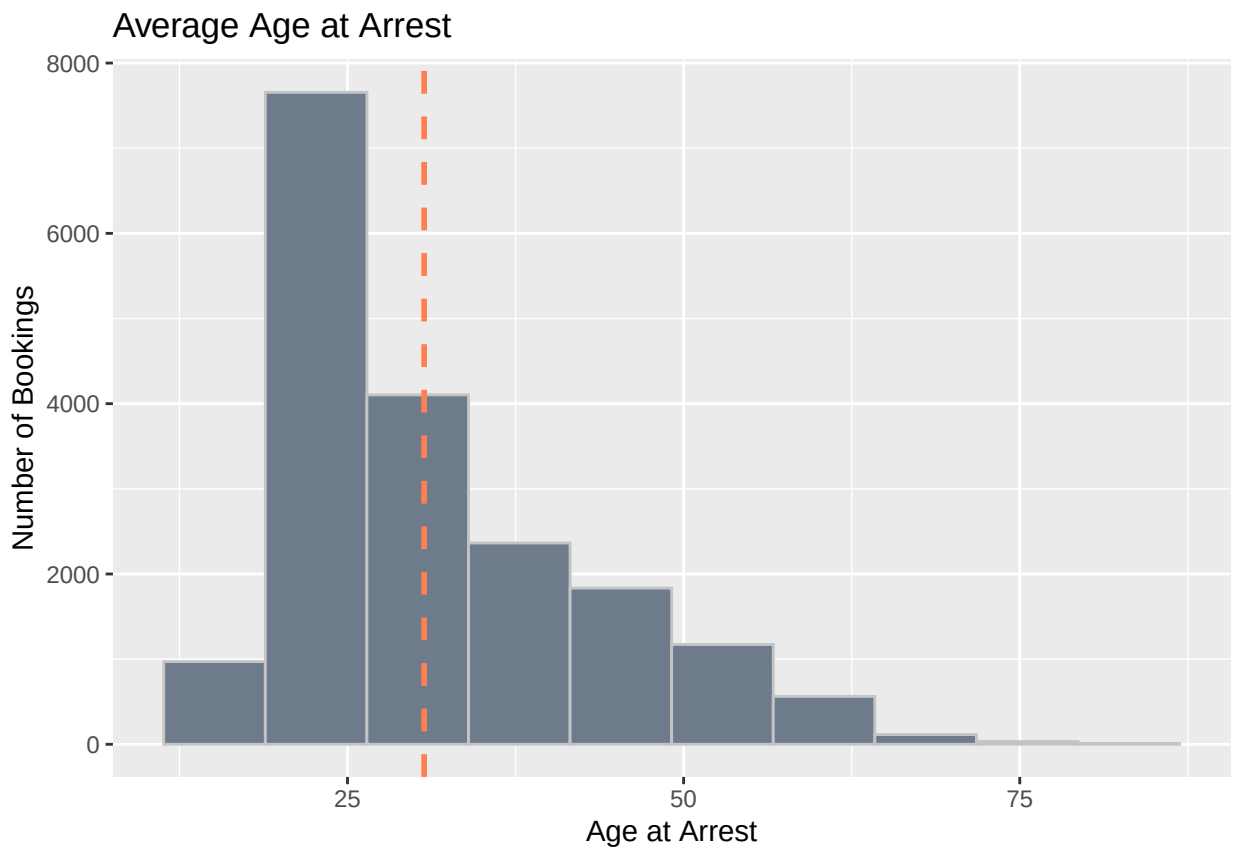
3.1 Dataset Overview / Visualization

The first thing we need to do is gain a high level understanding of our data.

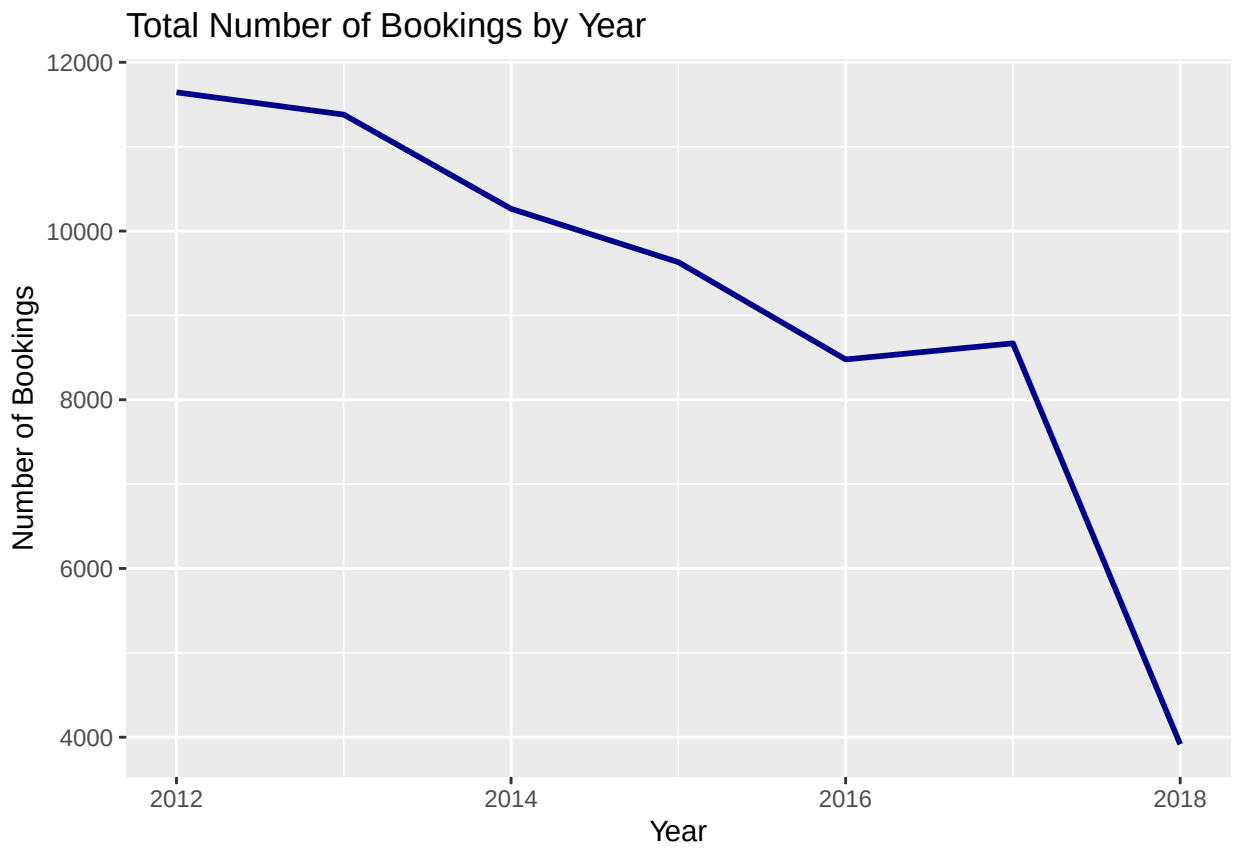
Summary Series of Charts

- The first chart is a histogram of the age at arrest with a vertical line drawn on the average age.
- The second chart is a simple line chart of number of arrest per year. You can see a trend downward since 2012. The data is very low for 2018, which tells me 2018 is not a full year of data.
- The last chart in this series is the number of crimes by crime type. I am not surprise to see traffic at the top of this listing.

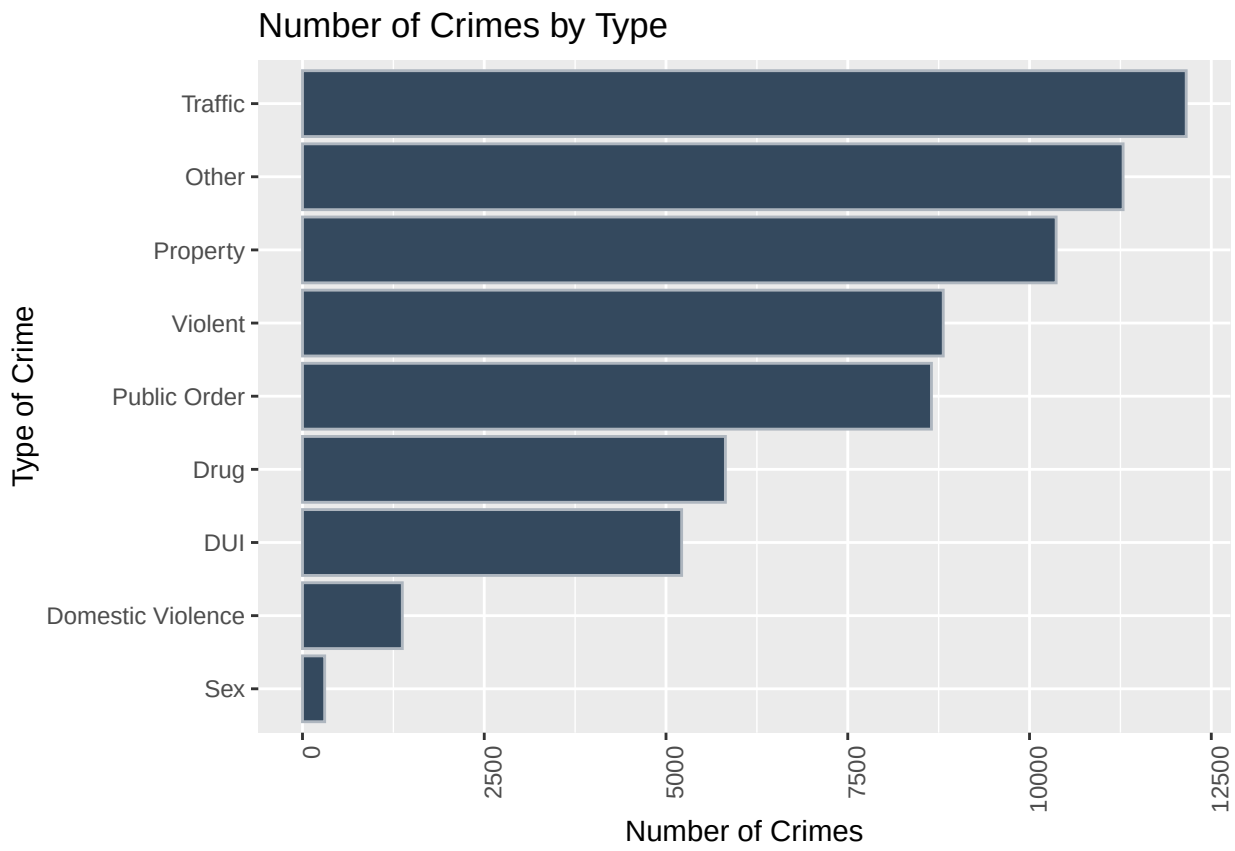
```
##############################################################################################
# SUMMARY CHARTS
##############################################################################################
#Age Distribution Histogram
chmJailData %>%
  group_by(jacket_number) %>%
  summarize(avg = mean(age_at_arrest)) %>%
  ggplot(aes(avg)) +
  geom_histogram(bins = 10, color = "gray77", fill = "lightsteelblue4") +
  geom_vline(xintercept=30.7, linetype="dashed", color = "coral",size=1) +
  ggtitle("Average Age at Arrest") +
  xlab("Age at Arrest") +
  ylab("Number of Bookings")
```




```
# crimes per year
chmJailData %>%
  group_by(booking_year) %>%
  summarize(Total=n())  %>%
  ggplot(aes(x=booking_year, y=Total)) +
  geom_line(color = "blue4", size=1) +
  ggtitle("Total Number of Bookings by Year") +
  xlab("Year")+
  ylab("Number of Bookings")
```




```
# Crimes by type
chmJailData %>%
  group_by(superhighlevel)%>%
  summarize(Total = n()) %>%
  ggplot(aes(x= reorder(superhighlevel, Total), y = Total)) +
  geom_bar(stat = "identity", color = "#aeb6bf", fill = "#34495e") +
  ggtitle("Number of Crimes by Type") +
  xlab("Type of Crime")+
  ylab("Number of Crimes") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  coord_flip()
```

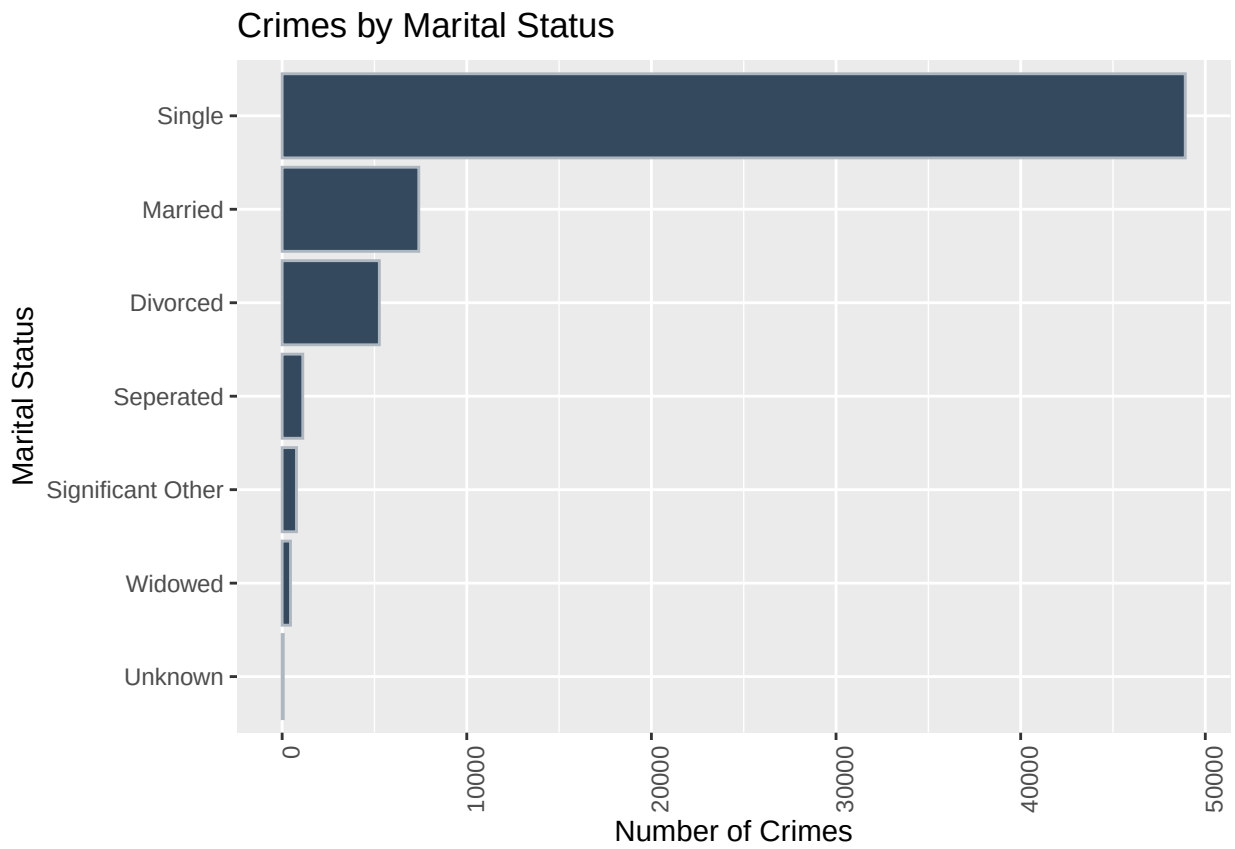


Marital Status Series of Charts

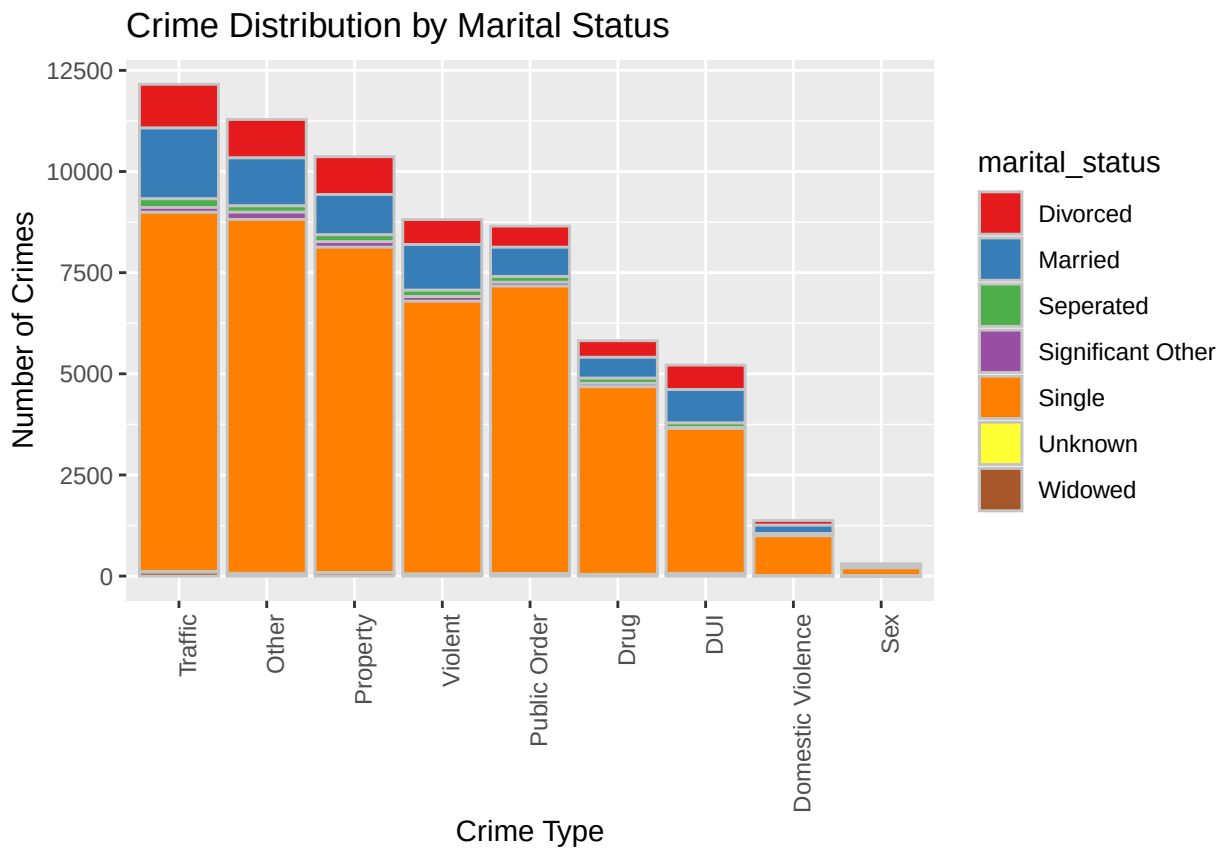
- The first chart in this series displays the total number of crimes by marital status.
- The next chart displays the distribution by type of crime and marital status.

There are no surprises in these charts. Given that the highest age group is in the 20's, it is not surprising that single people have a greater proportion of arrests.

```
##############################################################################
# MARITAL STATUS
##############################################################################
#Crimes by Marital Status
chmJailData %>%
  group_by(marital_status)%>%
  summarize(Total = n()) %>%
  ggplot(aes(x= reorder(marital_status, Total), y = Total)) +
  geom_bar(stat = "identity", color = "#aeb6bf", fill = "#34495e") +
  ggtitle("Crimes by Marital Status") +
  xlab("Marital Status")+
  ylab("Number of Crimes") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  coord_flip()
```



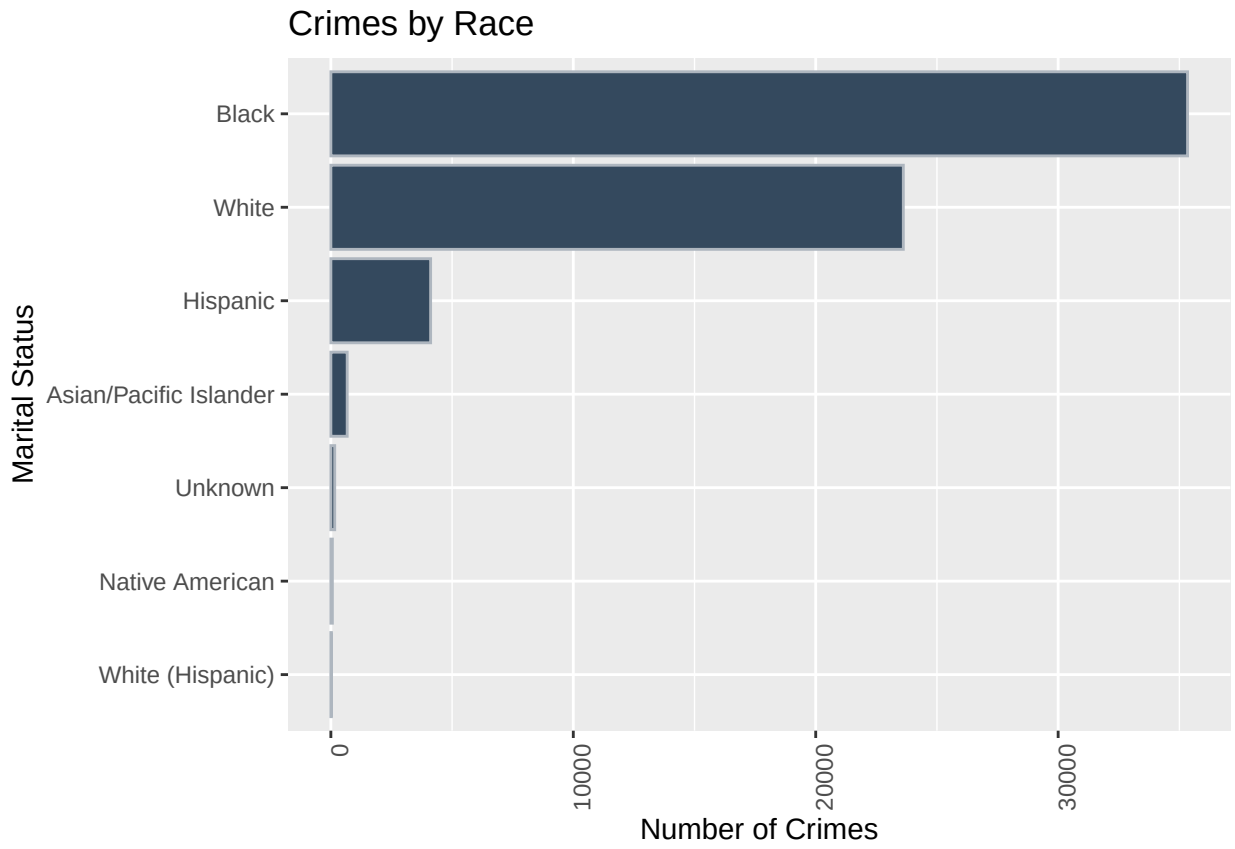

```
#Proportion by Marital Status
chmJailData %>%
  ggplot(aes(fct_infreq(superhighlevel))) +
  geom_bar(aes(fill=marital_status), color = "gray77") +
  scale_fill_brewer(palette = "Set1")+
  ggtitle("Crime Distribution by Marital Status") +
  xlab("Crime Type")+
  ylab("Number of Crimes") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



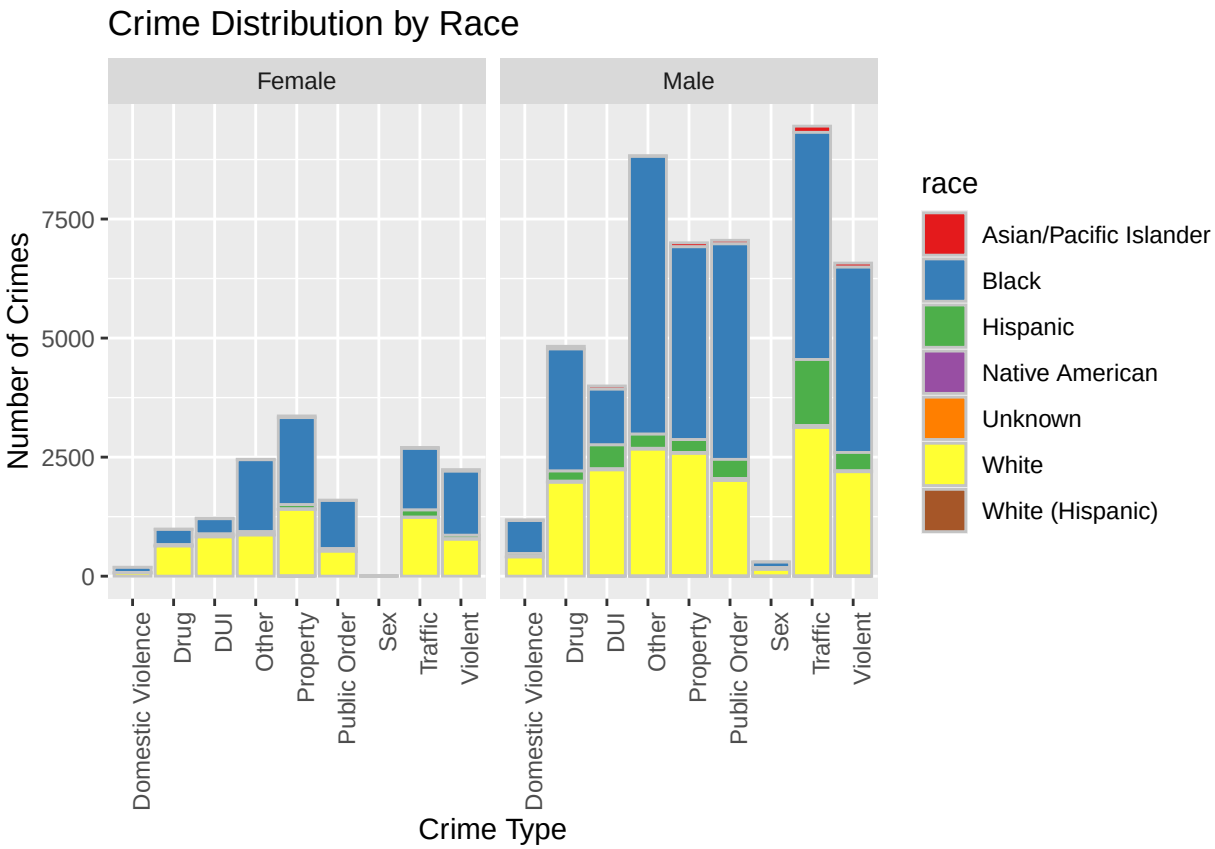
Race Series of Charts

- The first chart in this series displays the total number of crimes by race.
- The second chart displays the distribution by type of crime and race faceted by sex.
- The last chart in this series is a mosaic chart of crime and race faceted by whether or not the age is over 30.

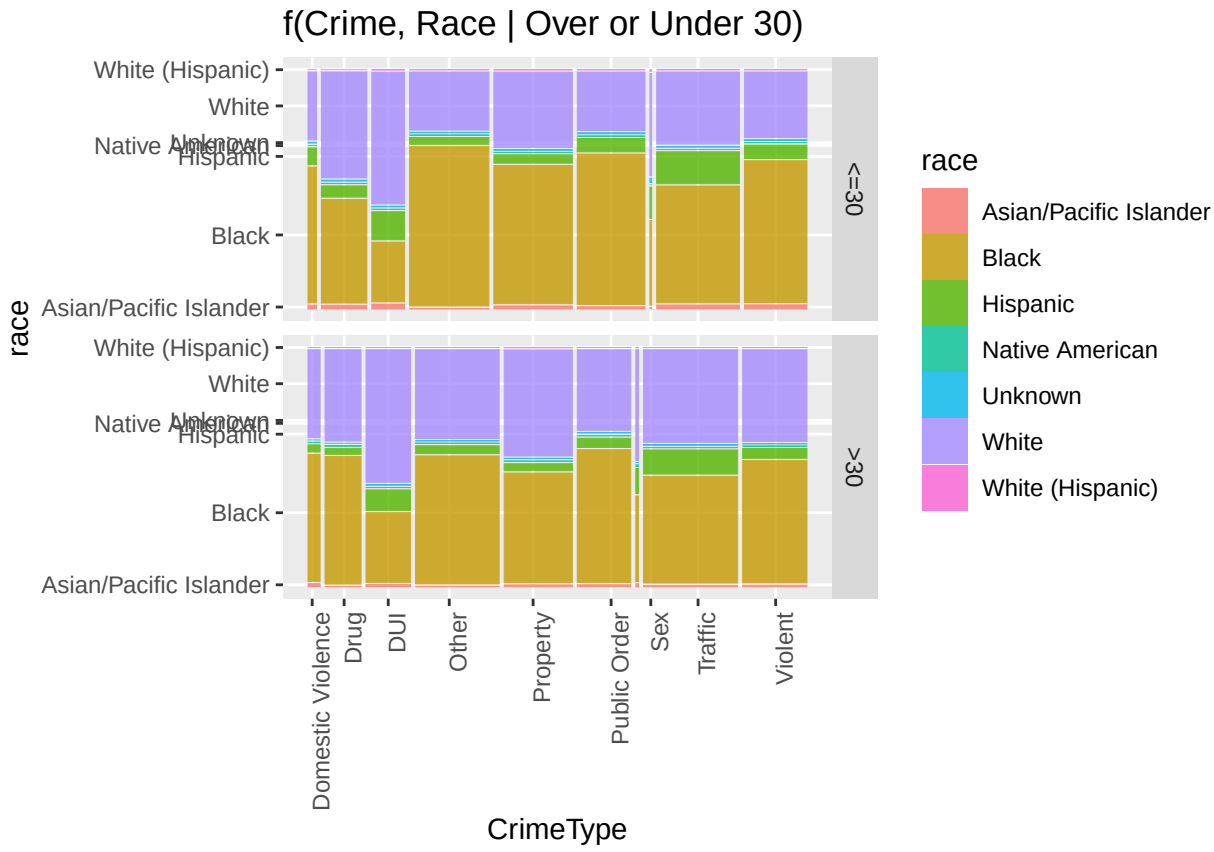
```
##############################################################################################
# RACE
##############################################################################################
#Crimes by Race
chmJailData %>%
  group_by(race)%>%
  summarize(Total = n()) %>%
  ggplot(aes(x= reorder(race, Total), y = Total)) +
  geom_bar(stat = "identity", color = "#aeb6bf", fill = "#34495e") +
  ggtitle("Crimes by Race") +
  xlab("Marital Status")+
  ylab("Number of Crimes") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  coord_flip()
```




```
#Crime Distribution by Race
chmJailData %>%
  ggplot(aes(superhighlevel)) +
  geom_bar(aes(fill=race), color = "gray77") +
  scale_fill_brewer(palette = "Set1")+
  ggtitle("Crime Distribution by Race")+
  xlab("Crime Type")+
  ylab("Number of Crimes") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  facet_grid(~sex)
```




```
#mosaic by race, crime type, faceted by <=30 and >30
chmJailData %>%
   mutate(CrimeType = superhighlevel)  %>%
  ggplot()+
  geom_mosaic(aes(x = product(race, CrimeType),
                  fill = race)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("f(Crime, Race | Over or Under 30)") +
  xlab("Crime Type") +
  facet_grid(over30~.)
```


f(Crime, Race | Over or Under 30)
White (Hispanic)
White
Native American
Unknown
Hispanic
Black
Asian/Pacific Islander
race
<=30
>30
Domestic Violence
Drug
DUI
Other
Property
Public Order
Sex
Traffic
Violent
CrimeType
race
Asian/Pacific Islander
Black
Hispanic
Native American
Unknown
White
White (Hispanic)

4 Results

As stated in the introduction, we will generate our models to determine if the age is <=30 or >30, given the predictors. The predictors used are crime severity, race, marital status, sex, military and booking year. I have also divided the type of models into three categories: Generative, Ensembles, and Trees and Forests.

4.1 Generative Models

These models take into account the joint distribution of y and the predictors x.

The code for all the models below is as follows where method is “naive_bayes”, “qda”, “lda”, or “glm”:

train_model <- train(over30 ~ crime_severity + race +
marital_status + sex + military +
booking_year,
method = “method”, data = training)

test_model_accuracy <- confusionMatrix(predict(train_model, test),
test$over30$)overall[“Accuracy”]

validation_accuracy <- confusionMatrix(predict(validation_accuracy, validation),
validation$over30$)overall[“Accuracy”]

4.1.1 Model 1 - Naive Bayes

Naive Bayes is the most general of the generative models. Naive Bayes is used for classifications datasets and is based on Bayes theorem.

As a refresher, below is Bayes Theorem:

$$P(A \mid B) = \frac{(P(B \mid A)P(A)}{P(B)}$$

The Accuracy obtained was 0.6718765 on the test set and 0.6699484 on the validation set. Let’s see if we can do better with our next generative model.

Method	Test Set Accuracy	Validation Set Accuracy
Model 1 - Naive Bayes	0.6718765	0.6699484

4.1.2 Model 2 - QDA

Per the course materials, Quadratic Discriminate Analysis, QDA, is a version of Naive Bayes in which we assume that the conditional probabilities for the predictors are multivariate normal. To train our model the code simply replaces the method variable for Naive Bayes to QDA. Let’s examine our results.

Variable Importance (using varImp())

```
## ROC curve variable importance
##
##                Importance
## marital_status   100.0000
## race              26.8715
## military          14.7783
## booking_year       8.2398
## sex                0.6826
## crime_severity     0.0000
```

The Accuracy obtained was 0.6890691 on the test set and 0.6845601 on the validation set. The accuracy is slightly better than Naive Bayes.

Method	Test Set Accuracy	Validation Set Accuracy
Model 1 - Naive Bayes	0.6718765	0.6699484
Model 2 - QDA	0.6890691	0.6845601

4.1.3 Model 3 - LDA

LDA provides a solution to the problem of having too many parameters. LDA assumes the correlation structure is the same for all classes, which reduces the number of parameters we need to estimate. Now we will run the LDA model.

The Accuracy obtained for LDA 0.7278500 on the test set and 0.7283951 on the validation set. This accuracy is much better than Naive Bayes and QDA.

Method	Test Set Accuracy	Validation Set Accuracy
Model 1 - Naive Bayes	0.6718765	0.6699484
Model 2 - QDA	0.6890691	0.6845601
Model 3 - LDA	0.7278500	0.7283951

4.1.4 Model 4 - GLM

The Generalized Linear Model (GLM) helps represent the dependent variable as a linear combination of independent variables. GLM is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution. Let's run the model.

The Accuracy obtained for the GLM model was 0.7300967 on the test set and 0.7276137 on the validation set. We improved on the LDA model.

Method	Test Set Accuracy	Validation Set Accuracy
Model 1 - Naive Bayes	0.6718765	0.6699484
Model 2 - QDA	0.6890691	0.6845601
Model 3 - LDA	0.7278500	0.7283951
Model 4 - GLM	0.7300967	0.7276137

4.2 Ensembles

Ensembles combine multiple machine learning algorithms into one model to improve predictions. The idea of an ensemble is similar to the idea of combining data from different models to obtain a better estimate of the true support, or true accuracy. For example, pollsters may combine results of more than one poll.

For the two ensemble models I replicated one of the class exercises on the Champaign Jail Dataset. First I combined all models, then I combined only the models that had an accuracy over 70%.

4.2.1 Model 5 - Ensemble - Naive Bayes, QDA, LDA, and GLM

The first of these models combines all four generative models we have ran thus far. The code for both ensemble models I am presenting looks as follows:

```
##############################################################################
# Ensembles using test set
##############################################################################
#now let's try ensemble to see if we can improve our accuracy
```

```r
#example taken from quiz in course materials section 6.1
#Ensembles combine multiple machine learning algorithms into one model to improve predictions

#train all the models
models <- c("naive_bayes", "qda", "lda", "glm")

fits <- lapply(models, function(model){
  train(over30 ~ crime_severity + race +
          marital_status + sex + military +
          booking_year,
          method = model, data = training)
})

names(fits) <- models

#generate matrix of predictions
pred <- sapply(fits, function(object)
  predict(object, newdata = test))

#compute Accuracy
acc <- colMeans(pred == test$over30)


#build ensemble to build prediction - over30 and under 30
over30Results <- rowMeans(pred == ">30")
y_hat <- ifelse(over30Results > 0.5, ">30", "<=30")


#calculation mean of accuracy
acc_hat <- sapply(fits, function(fit) min(fit$results$Accuracy))
ens_all_accuracy <- mean(acc_hat)

#only consider high accuracy
ind <- acc_hat >= 0.7
over30Reults2 <- rowMeans(pred[,ind] == ">30")
y_hat2 <- ifelse(over30Reults2>=0.5, ">30", "<=30")
ens_ldaglm_accuracy <- mean(y_hat2 == test$over30)
```

For Model 5, Ensemble - Naive Bayes, QDA, LDA, and GLM the results are 0.6905495 against the test set, and 0.6905027 against the validation set.

Method	Test Set Accuracy	Validation Set Accuracy
Model 1 - Naive Bayes	0.6718765	0.6699484
Model 2 - QDA	0.6890691	0.6845601
Model 3 - LDA	0.7278500	0.7283951
Model 4 - GLM	0.7300967	0.7276137
Model 5 - Ensemble with Naive Bayes, QDA, LDA, and GLM	0.6905495	0.6905027

4.2.2 Model 6 - Ensemble - GLM and LDA Only

Model 6 combine GLM and LDA models, which have our highest accuracies thus far. For Model 6, Ensemble - GLM and LDA Only the results are 0.7302921 against the test set, and 0.7284732 against the validation set. As we can see, ensembles do increase accuracy.

Method	Test Set Accuracy	Validation Set Accuracy
Model 1 - Naive Bayes	0.6718765	0.6699484
Model 2 - QDA	0.6890691	0.6845601
Model 3 - LDA	0.7278500	0.7283951
Model 4 - GLM	0.7300967	0.7276137
Model 5 - Ensemble with Naive Bayes, QDA, LDA, and GLM	0.6905495	0.6905027
Model 6 - Ensemble with LDA and GLM	0.7302921	0.7284732

4.3 Trees and Forests

The last two models are using trees and forests. Model 7 is a classification tree models and Model 8 uses the Random Forest Method.

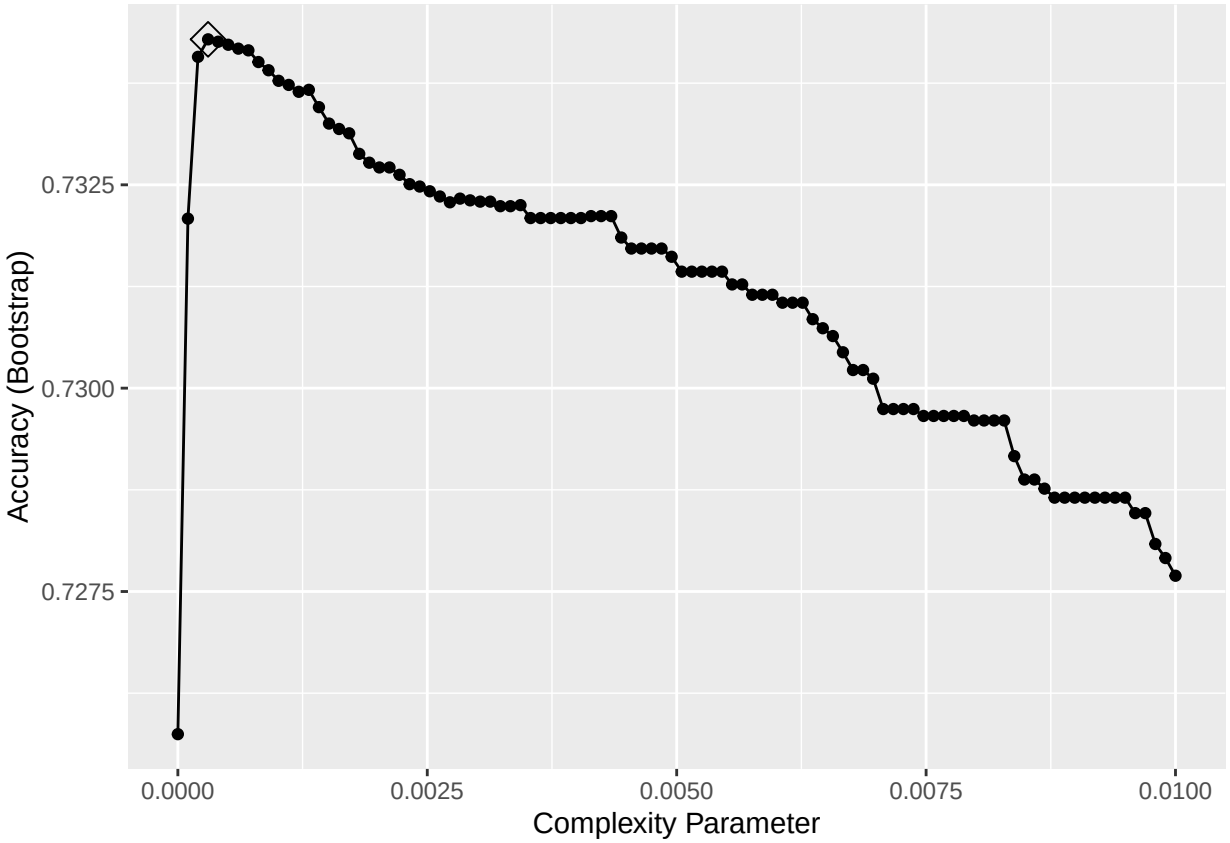
4.3.1 Model 7 - Classification Tree

Classification trees, or decision trees, are used in prediction problems where the outcome is categorical. The data in the dataset if largely categorical.

The code to run this model is displayed below.

```
##########################################################################
# Model 7 - Classification Tree (rPart)
##########################################################################

set.seed(30, sample.kind="Rounding")
train_rpart <- train(over30 ~ crime_severity + race +
                     marital_status + sex + military +
                     booking_year,
                     method = "rpart",
                     tuneGrid = data.frame(cp=seq(0,0.01, len=100)),
                     data = training)

rpart_accuracy <- confusionMatrix(predict(train_rpart, test), test$over30)$overall["Accuracy"]

#rPart on validation
val_rpart <- train(over30 ~ crime_severity + race +
                       marital_status + sex + military +
                       booking_year,
                     method = "rpart",
                     tuneGrid = data.frame(cp=seq(0,0.01, len=100)),
                     data = jail_train)
#ggplot(val_rpart, highlight = TRUE)
rpart_val_accuracy <- confusionMatrix(predict(val_rpart, validation),
                                      validation$over30)$overall["Accuracy"]

#add rows to a running table to summarize all model results
model_results <- bind_rows(model_results,
                           tibble(method="Model 7 - Classification Tree",
                                  TestAcc = rpart_accuracy,
                                  ValAcc = rpart_val_accuracy))
```

We can plot the tuning variables used in this model to see which value was chosen for optimization.



The Accuracy obtained for the Classification Tree model was 0.7306828 on the test set and 0.7301922 on the validation set. The accuracy is better than the generative and ensemble models.

Method	Test Set Accuracy	Validation Set Accuracy
Model 1 - Naive Bayes	0.6718765	0.6699484
Model 2 - QDA	0.6890691	0.6845601
Model 3 - LDA	0.7278500	0.7283951
Model 4 - GLM	0.7300967	0.7276137
Model 5 - Ensemble with Naive Bayes, QDA, LDA, and GLM	0.6905495	0.6905027
Model 6 - Ensemble with LDA and GLM	0.7302921	0.7284732
Model 7 - Classification Tree	0.7306828	0.7301922

4.3.2 Model 8 - Random Forest

According to the class materials, random forests address the shortcomings of decision trees. The goal is to improve prediction performance and reduce instability by averaging multiple decision trees (a forest of trees constructed with randomness).

The code to create the Random Forest is displayed below.

```
#############################################################################
# Model 8 - Random Forest
#############################################################################

set.seed(40, sample.kind="Rounding")
rf_train <- randomForest(over30 ~ crime_severity + race +
                         marital_status + sex + military +
```

```
                          booking_year,
                          data = training)
rf_accuracy <- confusionMatrix(predict(rf_train, test),
                               test$over30)$overall["Accuracy"]

#Test RF on validation set
set.seed(50, sample.kind="Rounding")
rf_validation <- randomForest(over30 ~ crime_severity + race +
                                marital_status + sex + military +
                                booking_year,
                                data = jail_train)
rf_val_accuracy <- confusionMatrix(predict(rf_validation, validation),
                                   validation$over30)$overall["Accuracy"]

#add rows to a running table to summarize all model results
model_results <- bind_rows(model_results,
                           tibble(method="Model 8 - Random Forest",
                                  TestAcc = rf_accuracy,
                                  ValAcc = rf_val_accuracy))
```

The Accuracy obtained for the Random Forest model was 0.7330273 on the test set and 0.7319112 on the validation set. The Random Forest model is our best fitting model.

Method	Test Set Accuracy	Validation Set Accuracy
Model 1 - Naive Bayes	0.6718765	0.6699484
Model 2 - QDA	0.6890691	0.6845601
Model 3 - LDA	0.7278500	0.7283951
Model 4 - GLM	0.7300967	0.7276137
Model 5 - Ensemble with Naive Bayes, QDA, LDA, and GLM	0.6905495	0.6905027
Model 6 - Ensemble with LDA and GLM	0.7302921	0.7284732
Model 7 - Classification Tree	0.7306828	0.7301922
Model 8 - Random Forest	0.7330273	0.7319112

5 Conclusion

The goal was to find the best fitting model that would predict if a person arrested was over 30 or under 30. The Random Forest was the model that predicted this with the highest accuracy of 73.2%.

I was a bit disappointed in the results of my analysis. At first I didn't obtain any results at all. Then I realized many of the columns had null values, so I removed them from the dataset. After that, I was only obtaining an accuracy around 60-63%. I thought I would obtain an accuracy in line with some of the course exercises that were around 85%. In fact, I was going to try the whole exercise over with a different dataset. However, I realized in the the real world, more than likely, you will not be able to select the dataset you use.

After my initial results in the 60s% I added an additional variable where I normalized the crime severity and was able to get an accuracy in the 70s%. If I had more time, I could probably normalize more of the data to obtain a higher accuracy.

I had high expectations when I started the Data Science Program at HarvardX. The class exceeded my expectations. I have a basic understanding of data modeling, and now understand what they mean. I understand how some models are derived and compared. Now, with my understanding of models, when I watch the election results or updates on Covid 19, I feel like I am a greater part of the conversation.