Week 6 Practice Problems

Josh Janda

2/27/2020

Week 6 Practice Problems

1. Using R in two separate code chunks, import the Chicago Food Inspections Data.

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## 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 ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tree)
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

```
library(dataPreparation)
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

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

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loading required package: progress

## dataPreparation 0.4.1

## Type dataPrepNews() to see new features/changes/bug fixes.
```

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

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

```
food = read_csv("https://uofi.box.com/shared/static/5637axblfhajotail80yw7j2s4r27hxd.csv",
    col_types = cols(Address = col_skip(),
        `Census Tracts` = col_skip(), City = col_skip(),
        `Community Areas` = col_skip(), `Historical Wards 2003-2015` = col_skip(),
        `Inspection Date` = col_date(format = "%m/%d/%Y"),
        Location = col_skip(), State = col_skip(),
        Wards = col_skip(), `Zip Codes` = col_skip()))
```

2. I want to do an analysis that focuses on the high and medium risk restaurants. Particularly, I am curious about the kind of violations restaurants get when they completely pass vs when they don't. The Problems (of the Investigative Cycle) are:

- Is it quantity or quality of the violations that determines a restaurant's passing?
- How do high risk restaurants compare to medium risk restaurants that completely pass vs those that fail?
- How well can we predict whether a restaurant will completely pass the inspection or not?

For starters, let's take a look at the data.

```
food = arrange(food, desc(`Inspection Date`)) # sorts data by most recent observations

colnames(food) = tolower(colnames(food))

head(food)
```

```
## # A tibble: 6 x 13
##   `inspection id` `dba name` `aka name` `license #` `facility type` risk
##             <dbl> <chr>      <chr>            <dbl> <chr>           <chr>
## 1         2290733 POLLERIA ~ POLLERIA ~     2428612 Poultry Slaugh~ Risk~
## 2         2290799 LA FAMILI~ LA FAMILI~     2665437 Grocery Store   Risk~
## 3         2290743 RED COACH~ RED COACH~       45131 Restaurant      Risk~
## 4         2290780 UVA KITCH~ UVA KITCH~     2647067 Restaurant      Risk~
## 5         2290770 TACO BELL  TACO BELL      2670614 Restaurant      Risk~
## 6         2290739 ARCHIES    ARCHIES        2636959 Restaurant      Risk~
## # ... with 7 more variables: zip <dbl>, `inspection date` <date>,
## #   `inspection type` <chr>, results <chr>, violations <chr>,
## #   latitude <dbl>, longitude <dbl>
```

Looking at the table above, we have a total of 13 variables. We have the business name, unofficial name, business licence #, type of business, risk level, zip code located in, inspection date, inspection type, inspection results, violations, latitude and longitude.

I am only focused on looking at businesses that are considered high and medium risk restaurants. I also only want to look at restaurants that Pass, Pass w/ Conditions, or completely fail. Lastly, I want to only select unique businesses, or the first entry of that business.

```
food_med_high_risk = food %>% filter(risk %in% c("Risk 2 (Medium)", "Risk 1 (High)"),
                                     results %in% c("Pass w/ Conditions", "Pass", "Fail"),
                                     `facility type` == "Restaurant")

food2 = distinct(food_med_high_risk, `license #`, .keep_all=TRUE) #keeps only first unique entry of lic

food2$totalviolations = str_count(food2$violations, "\\|") + 1 # sums total violations of business

food3 = food2 %>% filter(`license #` >= 1)

table(food3$results)
```

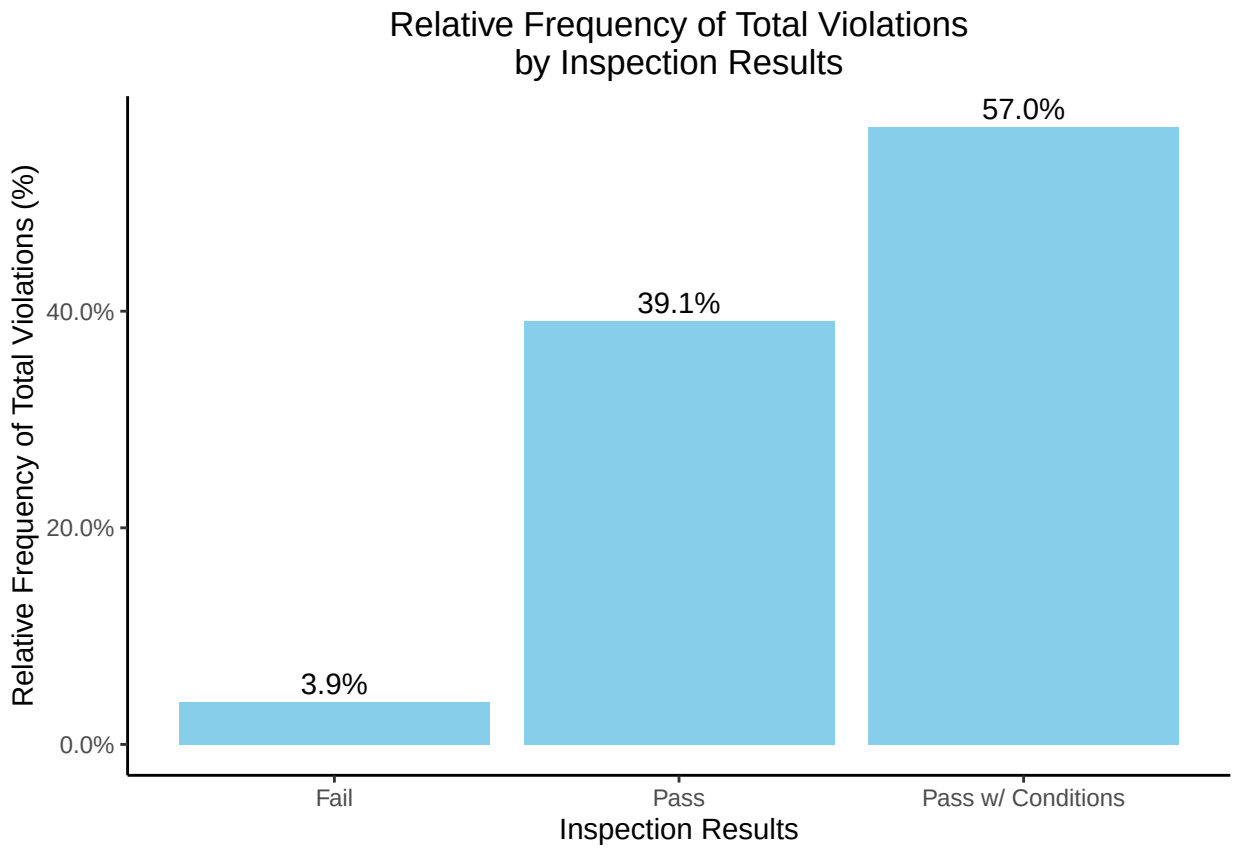
```
##
##               Fail               Pass Pass w/ Conditions
##                476              11652               5894
```

The table above gives us the number of restaurants that fail, pass, or pass w/ conditions. To answer the first question, I will utilize a grouped bar plot.

```
violations_per_pass_group = food3 %>%
  select(results, totalviolations) %>%
  drop_na(results, totalviolations) %>%
  group_by(results) %>% summarise(completeviolations = sum(totalviolations)) %>%
  mutate(proportion_violations = completeviolations / sum(completeviolations))

ggplot(data = violations_per_pass_group, aes(x = results, y = proportion_violations)) +
  geom_bar(stat = 'identity', fill = 'skyblue') +
  geom_text(aes(label = scales::percent(proportion_violations), y = proportion_violations),
            stat = 'identity', vjust = -.4) +
  labs(fill = "Risk Level", x = "Inspection Results",
       y = "Relative Frequency of Total Violations (%)",
       title = "Relative Frequency of Total Violations\nby Inspection Results") +
```

```
  theme(plot.title = element_text(hjust = 0.5),
        panel.background = element_rect(fill = "white"),
        axis.line = element_line(color = "black")) +
  scale_y_continuous(labels = scales::percent)
```



Looking at the plot above, we are able to see the relative frequency of the total violations each restaurant has received by each of ths inspection results. From the plot, we can see that 3.9% of the total amount of violations resulted in failures, 39.1% of the total violations resulted in a pass, and 57.0% of the total violations resulted in a pass w/ conditions. This leads me to believe that it is not the quantity of violations that result in a failure, but instead the quality of the violation. Many businesses are receiving numerous violations but passing inspection while a relatively small amount of total violations result in a failure.

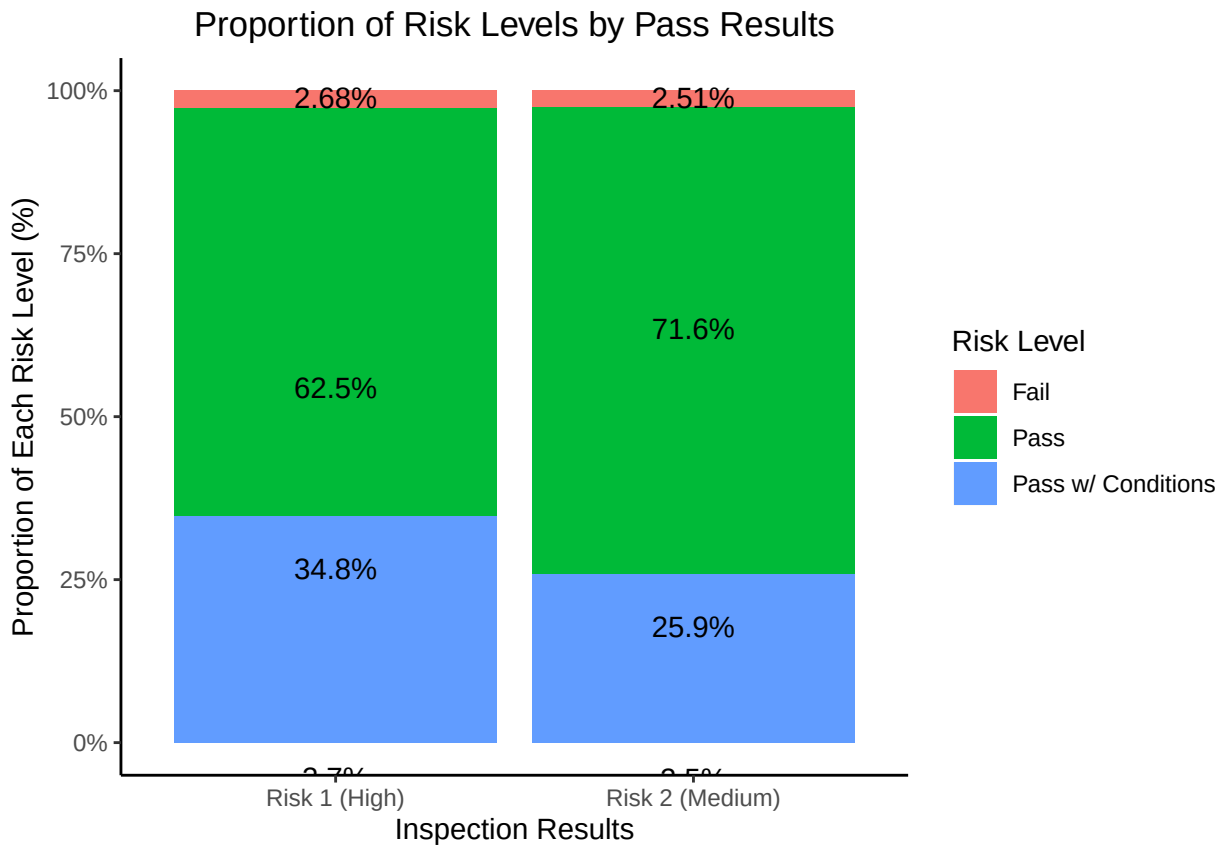
Next, I want to look at the question: How do high risk restaurants compare to medium risk restaurants that completely pass vs those that fail?

For this, I will utilize another barplot that looks at the relative frequency of passes and failures between risk levels of restaurants.

```
risk_vs_results = food3 %>%
  select(results, risk) %>%
  drop_na(results, risk) %>%
  group_by(risk) %>%
  count(results) %>%
  mutate(prop = n / sum(n))

ggplot(data = risk_vs_results, aes(x = risk, y = prop, fill = results)) +
```

```
  geom_bar(stat = 'identity') +
  geom_text(aes(label = scales::percent(prop), y = prop),
            stat = 'identity', vjust = 3) +
  labs(x = 'Inspection Results', y = 'Proportion of Each Risk Level (%)',
       fill = "Risk Level", title = 'Proportion of Risk Levels by Pass Results') +
  theme(plot.title = element_text(hjust = 0.5),
        panel.background = element_rect(fill = 'white'),
        axis.line = element_line(color = 'black')) +
  scale_y_continuous(labels = scales::percent) +
  annotate(geom = 'text', x = 'Risk 1 (High)', y = .99,
           label = scales::percent(risk_vs_results$prop[1])) +
  annotate(geom = 'text', x = 'Risk 2 (Medium)', y = .99,
           label = scales::percent(risk_vs_results$prop[4]))
```



Using the plot above, we are able to see the proportion of inspection results between each risk level. This let's us answer the question of how high risk restaurants compare to medium risk restaurants. What the plot tells us is that there is not much of a difference between medium and high risk restaurants.

For high risk restaurants:

- 34.8% of inspection results lead to a pass w/ conditions
- 63.5% of inspection results lead to a pass
- 2.68% of inspection results lead to a fail

For medium risk restaurants:

- 25.9% of inspection results lead to a pass w/ conditions
- 71.6% of inspection results lead to a pass
- 2.51% of inspection results lead to a fail

Overall, from this plot, we are able to understand that comparably high risk restaurants and medium risk restaurants have extremely similar fail rates and passing rates. Although, it should be noted that there pass w/ conditions rate are noticeably different where higher risk restaurants have a higher rate of passing with some sort of conditions.

For our last question, we want to answer: How well can we predict whether a restaurant will completely pass the inspection or not?

To answer this, I will be utilizing a classification tree.

```
set.seed(27)

tree_data = food3 %>%
  drop_na() %>%
  select(risk, longitude, latitude,
         totalviolations, results, inspection_type = `inspection type`) %>%
  mutate(results = as.factor(results),
         risk = as.factor(case_when(risk == 'Risk 1 (High)' ~ "High",
                           risk == 'Risk 2 (Medium)' ~ "Medium")))

tree_data$inspection_type = as.factor(tree_data$inspection_type)

trn_idx = createDataPartition(
  tree_data$results,
  p = .75,
  list = FALSE
) # partition data to create equal proportions of passes and failures in training and test set

tree_trn = tree_data[trn_idx,]
tree_tst = tree_data[-trn_idx,]

num_columns = colnames(tree_data %>% select_if(is.numeric)) # get numeric columns for scaling
scales = build_scales(tree_trn,
                      num_columns,
                      verbose = TRUE)
```

```
## [1] "build_scales: I will compute scale on  3 numeric columns."
## [1] "build_scales: it took me: 0s to compute scale for 3 numeric columns."
```

```
tree_trn_scaled = fastScale(tree_trn,
                            scales = scales,
                            verbose = TRUE)
```

```
## [1] "fastScale: I will scale 3 numeric columns."
## [1] "fastScale: it took me: 0s to scale 3 numeric columns."
```

```
tree_tst_scaled = fastScale(tree_tst,
                            scales = scales,
                            verbose = TRUE)
```

```
## [1] "fastScale: I will scale 3 numeric columns."
## [1] "fastScale: it took me: 0s to scale 3 numeric columns."
```

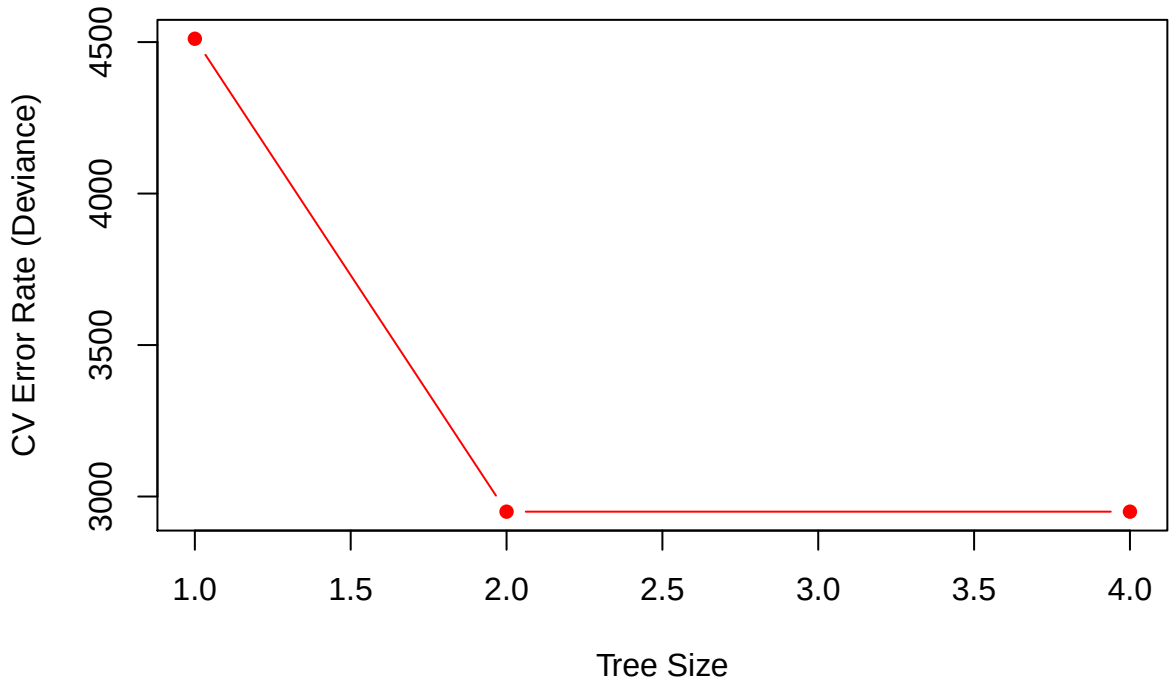
With the data split and scaled, I can now build the classification tree model. I will be using cross-validation in order to create a more robust/accurate model, the tree will be pruned using the misclassification rate.

```
set.seed(27)

tree_clf = tree(results ~ ., data = tree_trn_scaled)

cv_tree_clf = cv.tree(tree_clf,
                      FUN = prune.misclass,
                      K = 10)
```

```
plot(cv_tree_clf$size,
     cv_tree_clf$dev,
     type = 'b',
     pch = 16,
     col = 'red',
     xlab = 'Tree Size',
     ylab = 'CV Error Rate (Deviance)')
```

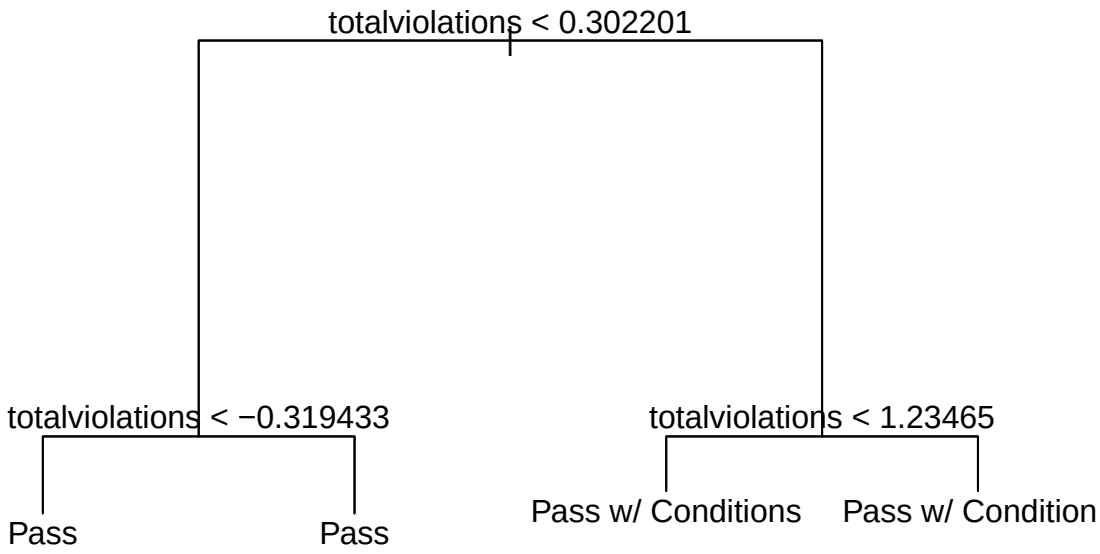


Looking at the plot above, we can see that the best tree sizes are 2 and 4. In order to choose a more complex model, I will choose the tree of size 4.

```
set.seed(27)

prune_cv_tree = prune.misclass(tree_clf, best = 4)

plot(prune_cv_tree)
text(prune_cv_tree, pretty = 0)
```



The tree is splitting on total violations, then on total violations again. Due to the data imbalance, our model is only classifying as pass or pass w/ conditions. There is no failure split.

Let's see how this model performs on the test data.

```
tree_pred = predict(prune_cv_tree,
                    tree_tst_scaled,
                    type = 'class')

tree_tst_acc = mean(tree_pred == tree_tst_scaled$results)
tree_tst_acc
```

```
## [1] 0.7011696
```

For our tree model, we get an accuracy rate of .7012 on the unseen test data. This is okay, but nowhere near great. Data imbalance is a huge factor causing this low accuracy rate, as well as the few amount of predictors that are useable.

Overall, I would say that we cannot predict whether a restaurant will pass inspection or not very well.