Valerie M. Jones

IST 707 Final Project

16 June 2021

**Introduction**

On October 1, 2020, the US Army has implemented a new combat ready physical fitness test, that was gender neutral, occupational specialty specific and similar to events soldiers’ face in combat conditions. The Army named this new physical fitness test the Army Combat Fitness Test (ACFT). The Army conducted a Baseline Soldier Physical Readiness Requirements Study (BSPRRS) to identify which fitness events most predict performance on tests of commonly performed physically demanding military tasks. It was concluded in the study that there were several gaps in the test participants and conflicting findings in the results. While the analysis was 80% predictive in the physical demands of specific Military Occupational Specialties, the BSPRRS identified the Leg tuck event was not the most predictive variable (Washington Post). Despite this analysis, the leg tuck was still included in the evaluation as one of the six events. There is currently congressional senate regarding whether the leg tuck is an effective assessment of physical readiness as it has a high failure rate amongst females. As a result the ACFT is suspended as a test of record until further analysis can be conducted on its equality. Soldiers are now required to take a diagnostic ACFT which will be used for analysis.

Concerns have been raised that this new 6 event test is discriminatory against certain military occupational specialties that are not regularly exposed to the physical demands of this new test, gender and soldiers who have physical limitations that would not exclude them from deploying but limit their ability to complete the required events.

Effective 1 April 2021 the Army transitioned to the ACFT 3.0, as part of the constant evolution of the ACFT. This updated ACFT establishes a tiered system based upon performance by gender. (HQDA EXORD 144-21)

Once a soldier has both achieved a score greater than 360 points (baseline passing) and exceeded the statistical average score for their gender, the soldier will be proportionally tiered based on their relative position above the average score. (HQDA EXORD 144-21) All units will record the ACFT data/scores in the Defense Training Management System (as diagnostic tests) which will serve as the source for data analysis. (HQDA EXORD 144-21)

The purpose of this project is to attempt to predict the number of Soldiers entering the three tiers established in ACFT 3.0 using various data mining techniques.

**About the Data**

Actual ACFT data is not available to mirror the ongoing study. In accordance with Headquarters Department of the Army, Executive Order 144-21 Army Physical Fitness Training and Testing for FY21-22:

*The Defense training Management System (DTMS) display and reports functionality has been modified effective 26 February 2021, so that it only provides aggregated unit test information which cannot be traced back to a soldier*. *The storing of ACFT data in locally maintained databases or spreadsheets is prohibited during the transition period and any files already in use must be entered into DTMS and then deleted. Commanders and unit leaders will not have access to individual soldier ACFT data and will only use the DTMS display and reports functionality to assess their organization's overall physical fitness*.

For the purposes of this project, historical Army Physical Fitness Test (APFT) results from DTMS will be utilized to create the tier system using the total numerical scores of the Push-Up event, Sit-Up event and 2mi run as indicators of tier performance. Because the primary platform of use is RStudio Cloud, all PII and unit designation information were removed prior to the data being uploaded.

Pre-processing

* A unique identifier was applied to the data to prevent duplication errors during model generation.
* Omit records of soldiers on physical limitation profiles who are unable to participate in all standard events (push-up, sit-up and 2-mile run)
* Convert 2 mile run result to integer replacing ‘:’ with decimal to determine the mean
* Random samples of 75 soldiers were extracted from each of the four company’s datasets creating a total population of 300 soldiers once combined

Tiers

Tier 1: Soldiers who score 90% or above in all three events

Tier 2: Soldiers who score between 240 and 280 out of 300 total points but score less than 90% in one or more events

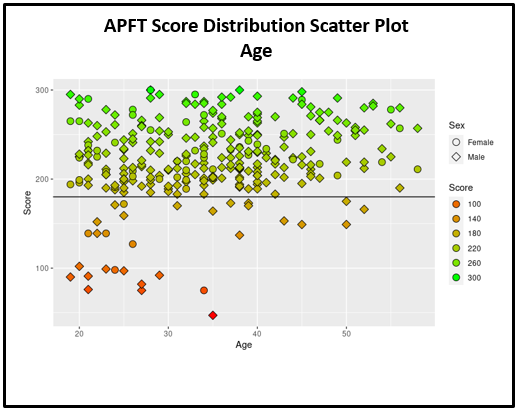
Tier 3: Soldiers who score 239 and below

Tier 4: Soldier fails fitness test

**Analysis**

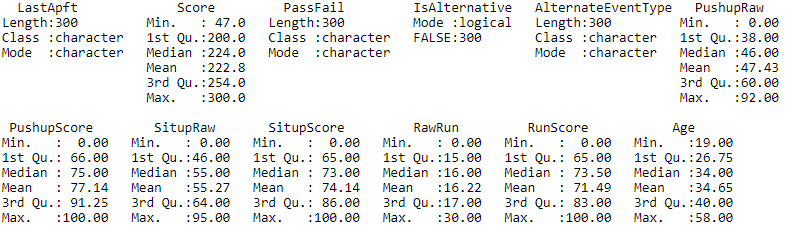
General Statistics

To provide an overview of the population that will be assigned to tiers a scatter plot and dot plot were used to illustrate the distribution of Soldiers’ APFT scores categorized by Age and Gender. The reference line depicts the minimum passing score of 180 soldiers must achieve.

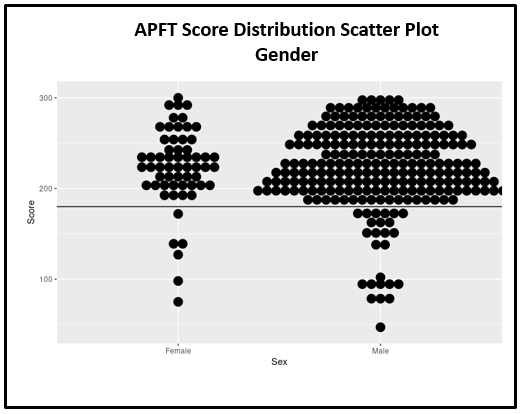


The above scatter plot illustrates the distribution of scores by the soldiers’ age. 88 percent of soldiers in the sample can achieve a passing score on the APFT. The scatterplot compares the number of male soldiers who pass the APFT versus the number of females. The mean score of the entire population is 222 and the median age is 34. With the previous APFT, Age and gender determined the scoring levels for each of the events. On the above scatterplot, the scores are evenly distributed for the various age groups, as a result age will not be considered a factor in this study.

The summary statistics below outlines specific statistics for each variable:



The dot plot below further illustrates soldiers’ scores in relation to the minimum passing score.



In addition to showing the scores in relation to the minimum passing score, this plot specifically identifies how gender is not equally weighted in the population. Females account for 20 percent of the sample where males account for 80 percent. This was also identified in the BPRRS study; females were underrepresented in the study[[1]](#footnote-1). For the purposes of both studies, males and females will be separated into different data sets with their own tiers. For the predictive models, males will be used as the training set and females will be used as the test set.

Association Rule Mining

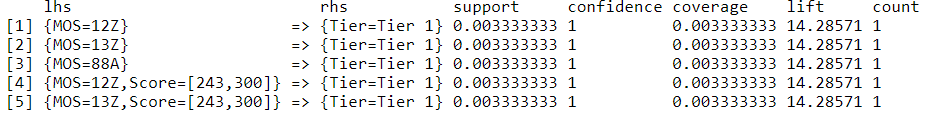
*What if we could predict the tier assignment based upon the soldiers’ Military Occupational Specialty?* The ACFT is focused toward setting standards based on the Soldiers’ Military Occupational Specialty (MOS). The ACFT standards are listed under three categories, Moderate (Gold), Significant (Gray), and Heavy (black).

The black category is for soldiers with “heavy” physically demanding positions. The majority of Combat Arms (Infantry, Field Artillery, Scouts etc.) And Aviation specialties fall within this category.

The gray category is for soldiers with “significant” physically demanding positions. The majority of Combat Support and Combat Service Support specialties fall within this category

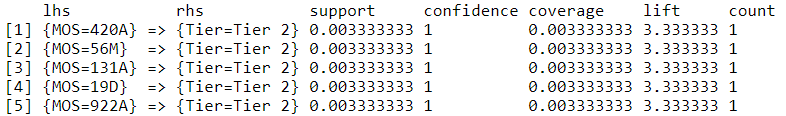
The gold category is for soldiers with “significant” physically demanding positions. Combat Support, Combat Service Support, and senior rank positions fall within this category

*Tier 1: Soldiers who score 90% or above in all three events*



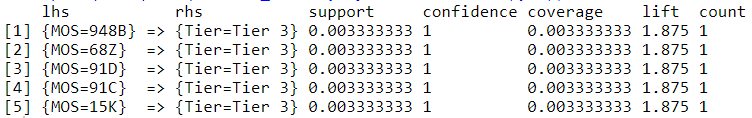
The Association Rule mining applied to Tier 1 shows 100% confidence that Engineer and Filed Artillery specialties will score in the Tier 1 category.

*Tier 2:* *Soldiers who score between 240 and 280 out of 300 total points but score less than 90% in one or more events*



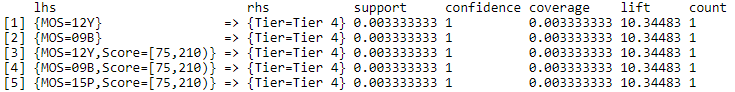
The Association Rule mining applied to Tier 2 shows 100% confidence that the majority of warrant officer specialties will score in the Tier 2 category.

*Tier 3:* *Soldiers who score 239 and below*



The Association Rule mining applied to Tier 3 shows 100% confidence that the majority of specialties considered under the “Gold” ACFT categorization will score in the Tier 3 category.

*Tier 4: Soldiers who fail the physical fitness test*

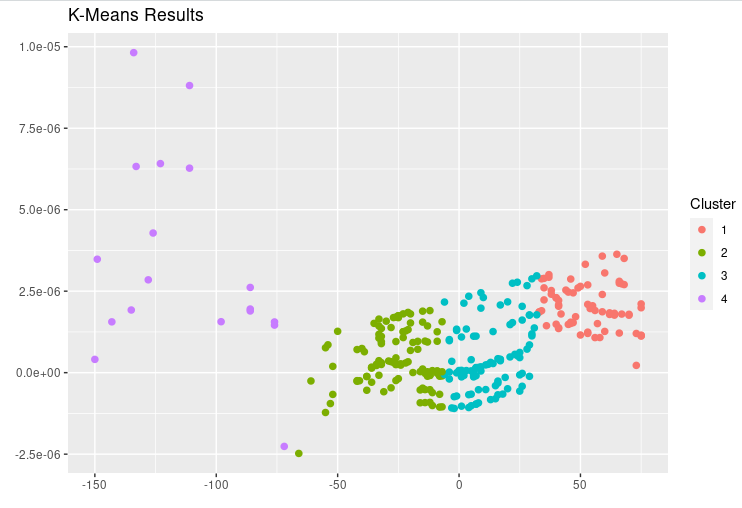


The Association Rule mining applied to Tier 4 shows a 100% confidence training pipeline Soldiers (09B) will be score into this tier, it is not conclusive as there is a mix of other Military Occupational Specialties.

While the association rule mining had 100% confidence in the placement of tiers, the total counts were limited to individual soldiers of the specific unit which is not an accurate predictor of identifying tiers based on Military Occupational Specialty. Tier predictions should not be concluded from this analysis as it is incomplete.

Clustering

K-Means –Results



The resulting clusters from this model are evenly distributed in comparison to the tiers in with respect to intra-cluster distance and in terms of inter-cluster distance. Cluster 4 is a more dispersed which can be attributed to the gaps in the different events. The main reason for these outliers is a result of Soldiers not completing all events. This typically occurs when a soldier fails an event and chooses not to participate in the proceeding event(s) as they have already failed the fitness test. This model best predicts the assignment of tiers.

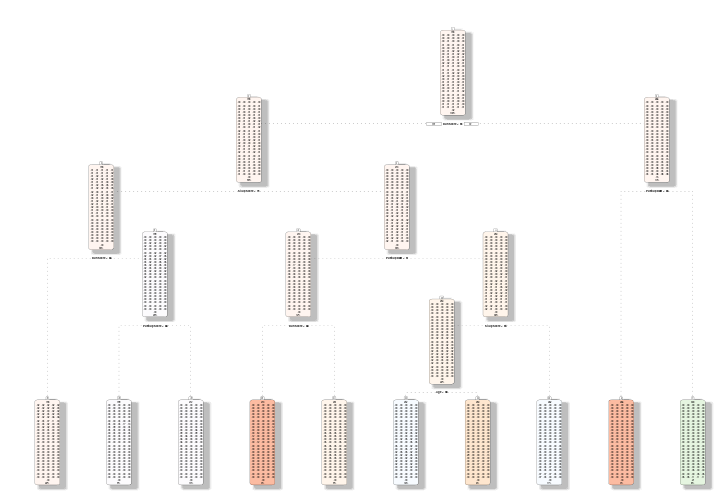
k Nearest Neighbor (kNN) Model

This model was significantly less accurate than the k-means clustering Model, but still had very strong prediction results. It achieved a 15% accuracy when predicting the male training data tiers and a 19% accuracy when predicting the female test data tiers.

Naives Bayes

Unfortunately, this model was not as successful as the k-means result as it was only able to accurately identify the male training data 40 percent of the time with the training data and returned a null value with the female test data.

Decision Tree



The decision tree model did not accurately predict tiers. This model used 14 nodes to decide the most important scores in determining the tiers. This model was only able to correctly predict the train data tiers with a 2 percent accuracy and the test data tiers with 5 percent accuracy.

Even with additional pruning accuracy rates did not increase. One hypothesis is the inability to normalize the data given its length and required values for accurate prediction for the study.

**Conclusion**

The US Army has implemented a new combat ready physical fitness test that was gender neutral and similar to events soldiers’ face in combat conditions. Concerns have been raised that this new 6 event test is discriminatory against certain military occupational specialties that are not regularly exposed to the physical demands of this new test, gender and soldiers who have physical limitations that would not exclude them from deploying but limit their ability to complete the required events.

Effective 1 April 2021 the Army transitioned to the ACFT 3.0, as part of the constant evolution of the ACFT. This updated ACFT establishes a tiered system based upon performance by gender. (HQDA EXORD 144-21)

The purpose of this project is to attempt to predict the number of Soldiers entering the three tiers established in ACFT 3.0 using various data mining techniques.

Various models were utilized in attempt to predict the number of soldiers entering a one of four tiers. Due to multiple variables with significant values, only the K-means clustering method provided decent accuracy in classifying the individual tiers. Its clusters were approximate to the criteria of the tiers so this model is would be useful in preliminary classification efforts. The Naïve Bayes model also had a moderately high accuracy rate for predicting the tiers. But this only applied to the training data.

In order to effectively predict tiers, more subsets of the data would need to be produced and stronger parameters to limit the population of each model. There were too many competing values for the decision tree to work which is why the accuracy rates were so low. Population size, despite scaling attempts, hindered the accuracy rates of the different models.

**Reflection**

This topic interested me in that I was curious to see how the current Army Combat Fitness test study could accurately classify the tiers based on soldiers’ individual performance.

It would have been more effective had I utilized the different events as subsets to predict the tiers. Using the total score as a factor created too many observations within the prediction models and too broad of a confusion matrix which is why the matrices were omitted from the final report.

There were too many competing values for the decision tree model to work which is why the accuracy rates were so low. In hindsight I should have limited the data to the final score instead of including the individual events.

The use of RStudio cloud also made this task more cumbersome as it has low memory and you have to frequently reinstall packages as the system refreshes. It is a good tool for minor projects, like the earlier assignments, but a project this detailed requires dedicated use of the desktop software. It met my need to be mobile.

**References**

Baseline Soldier Physical Readiness Requirements Study, A Peer Review, University of Iowa, 1 April 2020

Headquarters Department of the Army, Executive Order 144-21 Army Physical Fitness Training (APFT) and Testing For FY21-22. (HQDA EXORD 144-21)

U.S. Army Public Health Command, Public Health Report, Development of a New Army Standardized Physical Readiness Test January 2012 through December 2013, June 2015

National Defense Authorization Act for Fiscal Year 2021

Army Field Manual 7-22 (Holistic Health and Fitness), 1 October 2020

20-09 - Army Combat Fitness Test (Version 2), United States Army, Combined Arms Center, 18 February 2020.

**Code**

#install packages

install.packages("tm")

install.packages("slam")

install.packages("quanteda")

install.packages('proxy')

install.packages("tidytext")

install.packages("factoextra")

install.packages("mclust")

install.packages("useful")

install.packages('arules')

install.packages('arulesViz')

install.packages("tidyverse")

install.packages("e1071")

install.packages("reader")

install.packages("rpart")

install.packages("rattle")

install.packages("rpart.plot")

install.packages("RColorBrewer")

install.packages("Cairo")

install.packages("naivebayes")

#Load Libraries

library(stats)

library(dplyr)

library(ggplot2)

library(ggfortify)

library(readr)

library(arules)

library(arulesViz)

library(useful)

library(tm)

library(stringr)

library(slam)

library(quanteda)

library(SnowballC)

library(arules)

library(proxy)

library(cluster)

library(stringi)

library(proxy)

library(Matrix)

library(tidytext)

library(plyr)

library(ggplot2)

library(factoextra)

library(mclust)

library(reader)

library(e1071)

library(rpart)

library(rattle)

library(rpart.plot)

library(RColorBrewer)

library(Cairo)

library(naivebayes)

##########################Construct Data Set##############################################

#Load Project data from csv

setwd("//cloud//project//cloud\_data")

library(readr)

rdO <- read\_csv("Cloud\_Data/HSC\_HHBN.csv")

rdH <- read\_csv("Cloud\_Data/OPS\_HHBN.csv")

rdI <- read\_csv("Cloud\_Data/IS\_HHBN.csv")

rdS <- read\_csv("Cloud\_Data/SIG\_HHBN.csv")

#Omit physical limitation profiles

rdO<-rdO %>% filter(rdO$IsAlternative=="FALSE")

rdH<-rdH %>% filter(rdH$IsAlternative=="FALSE")

rdI<-rdI %>% filter(rdI$IsAlternative=="FALSE")

rdS<-rdS %>% filter(rdS$IsAlternative=="FALSE")

str(rdO)

view(rdO)

####Sample each data set, complete records only removing null records from Last APFT Date######

spl\_rdO<-sample\_n(rdO[complete.cases(rdO$LastApft),], size= 75)

spl\_rdH<-sample\_n(rdH[complete.cases(rdH$LastApft),], size= 75)

spl\_rdI<-sample\_n(rdI[complete.cases(rdI$LastApft),], size= 75)

spl\_rdS<-sample\_n(rdS[complete.cases(rdS$LastApft),], size= 75)

view(spl\_rdO)

#combine data sets

project\_rd<- rbind(spl\_rdO, spl\_rdH, spl\_rdI, spl\_rdS)

view(project\_rd)

######Data Clean#####

#since R cannot analyze Time effectively, I've turned run times into integers

project\_rd$RawRun<-str\_replace(project\_rd$RawRun,":",".") #replace run colon with decimal

project\_rd$RawRun<-str\_replace(project\_rd$RawRun,":00"," ") #replace milliseconds with null

project\_rd$RawRun <- as.integer(project\_rd$RawRun)#convert to numeric value

project\_rd$Sex<-str\_replace(project\_rd$Sex,"F","Female")

project\_rd$Sex<-str\_replace(project\_rd$Sex,"M","Male")

pd<-project\_rd

pd$AgeRange<-cut(pd$Age,breaks=c(17,21,26,31,36,41,46,51,60),labels=c("17-21","22-26","27-31","32-36",

"37-41","42-46","47-51","52+"))

##Create Tiers

#Based on predetermined factors:

#Tier 1: Soldiers who score 90 or above in all three events

#Tier 2: Soldiers who score between 240-280 (score less than 90 in one or more events)

#Tier 3: Soldiers who score 239 and below

#Tier 4: Soldier fails test

pd$Tier<- ifelse(pd$PushupScore>89 & pd$SitupScore>89 & pd$RunScore > 89 ,"Tier 1",

ifelse(pd$PushupScore<90 & pd$Score >239 | pd$SitupScore<90 & pd$Score >239 | pd$RunScore<90 & pd$Score >239, "Tier 2",

ifelse(pd$Score<240 & pd$PassFail=="Pass","Tier 3", "Tier 4"

)

))

view(pd)

summary(pd)

#Calculate mean

pd %>% summarize(Avg = mean(Score)) #total

pd %>% summarize(Avg = mean(Age)) #total

pd %>% filter(Sex=="Male") %>% summarize(Avg = mean(Age)) #male

pd %>% filter(Sex=="Male")%>% summarize(Avg = mean(Score)) #male

pd %>% filter(Sex=="Female") %>% summarize(Avg = mean(Age)) #female

pd %>% filter(Sex=="Female")%>% summarize(Avg = mean(Score)) #female

#####Visuals####

view(pd)

g<-ggplot(pd,aes(x=Age,y=Score,shape=Sex,fill=Score, title="APFT Score Distribution Scatterplot"))+geom\_point(size=3.5)+geom\_hline(yintercept = 180)+scale\_shape\_manual(values=c(21,23))+scale\_fill\_gradient(low="red",high="green",breaks=seq(100,300, by=40),guide=guide\_legend(override.aes=list(shape=21)))

g

##dot plot

dg<-ggplot(pd,aes(x=Sex,y=Score))+geom\_dotplot(binaxis="y",binwidth=10,stackdir="center")+geom\_hline(yintercept = 180)

dg

##Histogram of total scores

hist(pd$Score)

##Tiers Pie Chart

tiers <-table(pd$Tier)

tiers

pie(tiers)

#####Association Rule Mining

install.packages("arules")

install.packages("arulesViz")

library("arules")

library("arulesViz")

#Create dataframe

str(pd2)

apd <-pd2[,c("MOS","Score","Tier")]

# Fitting model

# Training Apriori on the dataset

set.seed = 250 # Setting seed

associa\_rules1 <- apriori(data=apd, parameter=list (supp=0.001,conf = 0.08), appearance = list (rhs="Tier=Tier 1"))

associa\_rules2 <- apriori(data=apd, parameter=list (supp=0.001,conf = 0.08), appearance = list (rhs="Tier=Tier 2"))

associa\_rules3 <- apriori(data=apd, parameter=list (supp=0.001,conf = 0.08), appearance = list (rhs="Tier=Tier 3"))

associa\_rules4 <- apriori(data=apd, parameter=list (supp=0.001,conf = 0.08), appearance = list (rhs="Tier=Tier 4"))

# Plot,

plot(apd, topN = 10)

# Visualizing the results

inspect(head(sort(associa\_rules1, by = "confidence"), 5))

inspect(head(sort(associa\_rules2, by = "confidence"), 5))

inspect(head(sort(associa\_rules3, by = "confidence"), 5))

inspect(head(sort(associa\_rules4, by = "confidence"), 5))

plot(associa\_rules1, method = "graph",

measure = "confidence", shading = "lift")

pd2<-pd[c(1,5,14,18,19,20,21,22,23,24,25,26,27)]

view(pd2)

###MALES####

pd\_male<-pd2[c(1,2,3,4,5,6,7,8,9,10,13)]

pd\_male<-pd2 %>% filter(pd2$Sex=="Male")

view(pd\_male)

mtiers<-table(pd\_male$Tier)

tiers

pie(mtiers)

###FEMALES####

pd\_female<-pd2[c(1,2,3,4,5,6,7,8,9,10,13)]

pd\_female<-pd2 %>% filter(pd2$Sex=="Female")

view(pd\_female)

ftiers<-table(pd\_female$Tier)

ftiers

pie(ftiers)

#######Normalize apd (ineffective for each dataframe)

#Create Matrix

#apdmatrix<-as.matrix(apd)

#Normalize fedfilesmatrix

#ffn1<-apply(apdmatrix,1,function(i)round(i/sum(i),3))

#apdnorm<-t(ffn1)

#verify normalization

#(apdmatrix[1:11,1:10])

#(apdnorm[1:11,1:10])

#Kmeans

allkmeans<-kmeans(apd[,2],4)

(allkmeans$cluster)

kmeanstable<-table(rownames(apd),allkmeans$cluster)

kmeanstable

plot(allkmeans,apd,title="K-Means Results",xlab="",ylab="")

str(apd)

#Kmeans Males

male\_kmeans<-kmeans(pd\_male,3)

(male\_kmeans$cluster)

kmeansamaletable<-table(rownames(pd\_male),male\_kmeans$cluster)

kmeansamaletable

plot(male\_kmeans,pd\_male,title="K-Means - Male Results",xlab="",ylab="")

#Kmeans Female

female\_kmeans<-kmeans(pd\_female,3)

(female\_kmeans$cluster)

kmeansafemaletable<-table(rownames(pd\_female),female\_kmeans$cluster)

kmeansafemaletable

plot(female\_kmeans,pd\_female,title="K-Means - Female Results",xlab="",ylab="")

str(pd\_female)

str(pd\_male)

#create dataframes excluding NA columns

traindigitdf<-subset(pd\_male, select = -c(1,2,11,12,13))

testdigitdf<-subset(pd\_female, select = -c(1,2,11,12,13))

#convert the data to numeric values and factors

traindigitdf<-data.frame(sapply(traindigitdf, as.numeric))

testdigitdf<-data.frame(sapply(testdigitdf, as.numeric))

traindigitdf$Score<-as.factor(traindigitdf$Score)

noheadertrn<-data.frame(traindigitdf[,-1])

str(data.frame(testdigitdf[,]))

#knn train

ksize <- round(sqrt(nrow(traindigitdf)))

ksize

#Predict Train Data

knn\_trn <- class::knn(train=traindigitdf[-1], test =traindigitdf[-1], cl=traindigitdf$Score, k = ksize, prob=FALSE)

print(knn\_trn)

crazytrain\_knn <- (table(knn\_trn, traindigitdf$Score))

crazytrain\_knn\_chk <- sum(diag(crazytrain\_knn))/nrow(traindigitdf)

crazytrain\_knn

crazytrain\_knn\_chk

#0.15

#knn test

ksize2 <- round(sqrt(nrow(testdigitdf)))

ksize2

#Predict Test Data

knn\_tst <- class::knn(train=testdigitdf[-1], test =testdigitdf[-1], cl=testdigitdf$Score, k = ksize2, prob=FALSE)

print(knn\_tst)

crazytst\_knn <- (table(knn\_tst, testdigitdf$Score))

crazytst\_knn\_chk <- sum(diag(crazytst\_knn))/nrow(testdigitdf)

crazytst\_knn

crazytst\_knn\_chk

summary(pd\_male)

view(pd\_male)

library(rpart)

#Decision Tree

train\_dt<-rpart(traindigitdf$Score~., data=traindigitdf, method="class")

library(rpart.plot)

library(rattle)

#plot decision tree

fancyRpartPlot(train\_dt)

# Visualizing the results

inspect(sort(associa\_rules, by = 'lift')[1:10])

plot(associa\_rules, method = "graph",

measure = "confidence", shading = "lift")

#create dataframes excluding NA columns

traindigitdf<-subset(pd\_male, select = -c(1,2,11,12))

testdigitdf<-subset(pd\_female, select = -c(1,2,11,12))

traindigitdf<-data.frame(sapply(traindigitdf, as.numeric))

testdigitdf<-data.frame(sapply(testdigitdf, as.numeric))

traindigitdf$Score<-as.factor(traindigitdf$Score)

noheadertrn<-data.frame(traindigitdf[,-1])

str(data.frame(testdigitdf[,]))

#knn

ksize <- round(sqrt(nrow(traindigitdf)))

ksize

#Predict Train Data

knn\_trn <- class::knn(train=traindigitdf[-1], test =traindigitdf[-1], cl=traindigitdf$Score, k = ksize, prob=FALSE)

print(knn\_trn)

crazytrain\_knn <- (table(knn\_trn, traindigitdf$Score))

crazytrain\_knn\_chk <- sum(diag(crazytrain\_knn))/nrow(traindigitdf)

crazytrain\_knn

crazytrain\_knn\_chk

#0.1949153

summary(pd\_male)

view(pd\_male)

#decision tree

train\_dt<-rpart(traindigitdf$Score~., data=traindigitdf, method="class")

summary(train\_dt)

fancyRpartPlot(train\_dt)

#prune

train\_dt<- prune(train\_dt, cp=.016)

fancyRpartPlot(train\_dt)

summary(train\_dt)

#predict train data

predtrn\_dt<- predict(train\_dt, noheadertrn, type ="class")

#confusion matrix

crazytrain\_dt<- table(predtrn\_dt, traindigitdf$Score)

crazytrain\_dt

#validate prediction accuracy

sum(diag(crazytrain\_dt))/nrow(traindigitdf)

#predict test data

predtst\_dt<- predict(train\_dt, testdigitdf, type = "class")

print(predtst\_dt)

#confusion matrix

crazytst\_dt<- table(predtst\_dt, testdigitdf$Score)

crazytst\_dt

#validate prediction accuracy

sum(diag(crazytst\_dt))/nrow(testdigitdf)

library(naivebayes)

library(e1071)

## Naive Bayes Model

traindigit\_nb<- naiveBayes(traindigitdf$Score~., data=traindigitdf, na.action=na.pass)

table(traindigit\_nb)

length(traindigit\_nb)

#predict train data

predtrn\_nb<-predict(traindigit\_nb, noheadertrn)

predtrn\_nb

#confusion matrix

crazytrain\_nb<- table(traindigitdf$Score, predtrn\_nb)

crazytrain\_nb

#validate prediction accuracy

sum(diag(crazytrain\_nb))/nrow(traindigitdf)

#predict test data

predtst\_nb<-predict(traindigit\_nb,testdigitdf)

table(predtst\_nb)

#confusion matrix

crazytst\_nb<- table(testdigitdf$Score, predtst\_nb)

crazytst\_nb

#validate prediction accuracy

sum(diag(crazytst\_nb))/nrow(testdigitdf)

1. (84% men, 16% women) Baseline Soldier Physical Readiness Requirements Study, University of Iowa, 1 April 2020 [↑](#footnote-ref-1)