#IST 687 Project

#Cory Wellbrock, Jean Thompson, Trevor Kahl

#https://archive.ics.uci.edu/ml/datasets/Adult

#How can we determien msot effectively if someone makes over 50,000.

#What are indicators that we can use to best predict this? Education, relationship, native country, etc?

#If I am a certain job title can we predict a base of pay?

#Formula to make prediction

#cleaning the dataset

#basic functions: mean, mode, regression

#histograms

#ggplot2

#world heat map

#randomForest

#https://zoom.us/rec/play/uJYqdOn9p203EoXG4gSDBvR\_W465eP-s1CMf\_qEEnxy8B3RXYQfwYLQbZ-KoPEFXi2D2f3zoxd\_wzttc

####################################################################

#################### Load Data and Name Columns ####################

####################################################################

#URL to dataset

url <-"https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"

#Loading data without a header (dataset doesn't contain a header)

adultData <- read.csv(url, header = FALSE)

nrow(adultData)

#-- 32561

colnames(adultData)

head(adultData)

#Renaming the columns

colnames(adultData) <- c("age", "workclass", "fnlwgt", "education", "education-num", "marital-status",

"occupation", "relationship", "race", "sex", "capital-gain", "capital-loss",

"hours-per-week", "native-country", "income")

colnames(adultData)

head(adultData)

str(adultData)

####################################################################

####################### Cleaning the dataset #######################

####################################################################

#Removing 'capital-gain' and 'capital-loss'

adultData <- adultData[,-11:-12]

colnames(adultData)

#Columns with NAs

colnames(adultData)[colSums(is.na(adultData)) > 0]

#-- Zero initially

#Finding question marks/periods and making NA

idx <- adultData == " ?"

is.na(adultData) <- idx

#Columns with NAs

colnames(adultData)[colSums(is.na(adultData)) > 0]

#-- [1] "workclass" "occupation" "native-country"

#Number of rows before

nrow(adultData)

#-- 32561

#Count of total NAs

sum(is.na(adultData))

#-- 4262

#Omit the NAs

adultData <- na.omit(adultData)

#Count of NAs after

sum(is.na(adultData))

#-- 0

#New number of rows

nrow(adultData)

#-- 30162

#Structure before trimming

str(adultData)

#Backup data

adultData\_bak <- adultData

#Restore data

adultData <- adultData\_bak

#Trimming leading Whitespace

adultData[,1] <- trimws(adultData[,1])

adultData[,2] <- trimws(adultData[,2])

adultData[,3] <- trimws(adultData[,3])

adultData[,4] <- trimws(adultData[,4])

adultData[,5] <- trimws(adultData[,5])

adultData[,6] <- trimws(adultData[,6])

adultData[,7] <- trimws(adultData[,7])

adultData[,8] <- trimws(adultData[,8])

adultData[,9] <- trimws(adultData[,9])

adultData[,10] <- trimws(adultData[,10])

adultData[,11] <- trimws(adultData[,11])

adultData[,12] <- trimws(adultData[,12])

adultData[,13] <- trimws(adultData[,13])

#Resetting structure

adultData[,1] <- as.integer(adultData[,1])

adultData[,2] <- as.factor(adultData[,2])

adultData[,3] <- as.integer(adultData[,3])

adultData[,4] <- as.factor(adultData[,4])

adultData[,5] <- as.integer(adultData[,5])

adultData[,6] <- as.factor(adultData[,6])

adultData[,7] <- as.factor(adultData[,7])

adultData[,8] <- as.factor(adultData[,8])

adultData[,9] <- as.factor(adultData[,9])

adultData[,10] <- as.factor(adultData[,10])

adultData[,11] <- as.integer(adultData[,11])

adultData[,12] <- as.factor(adultData[,12])

adultData[,13] <- as.factor(adultData[,13])

adultData4 <- adultData

str(adultData)

#Finding unique counts per column

apply(adultData, 2, function(x) length(unique(x)))

####################################################################

############################ Bar Charts ############################

####################################################################

library(ggplot2)

#Count by race

barRace <- ggplot(adultData, aes(x = race)) + geom\_bar(color="black", fill="white") +

ggtitle("Race Bar Chart") + theme(axis.text.x = element\_text(angle = 45, hjust=1))

barRace

#Count by race, color fill by education

barRaceEdu <- ggplot(adultData, aes(x = race)) + geom\_bar(aes(fill=education)) +

ggtitle("Race/Education Bar Chart") + theme(axis.text.x = element\_text(angle = 45, hjust=1))

barRaceEdu

#Count by race, color fill by workclass

barRaceWork <- ggplot(adultData, aes(x = race)) + geom\_bar(aes(fill=workclass)) +

ggtitle("Race/Workclass Bar Chart") + theme(axis.text.x = element\_text(angle = 45, hjust=1))

barRaceWork

#Count by race, color fill by Occupation

barRaceOcc <- ggplot(adultData, aes(x = race)) + geom\_bar(aes(fill=occupation)) +

ggtitle("Race/Occupation Bar Chart") + theme(axis.text.x = element\_text(angle = 45, hjust=1))

barRaceOcc

library(gridExtra)

grid.arrange(barRaceEdu, barRaceWork, barRaceOcc, ncol = 3)

#Count by age

barAge <- ggplot(adultData, aes(x = age)) + geom\_bar(color="black", fill="white") +

ggtitle("Age Bar Chart") + theme(axis.text.x = element\_text(angle = 90, hjust=1))

barAge

#Count by workclass

barWorkclass <- ggplot(adultData, aes(x = workclass)) + geom\_bar(color="black", fill="white") +

ggtitle("Workclass Bar Chart") + theme(axis.text.x = element\_text(angle = 45, hjust=1))

barWorkclass

#Count by education

barEdu <- ggplot(adultData, aes(x = education)) + geom\_bar(color="black", fill="white") +

ggtitle("Education Bar Chart") + theme(axis.text.x = element\_text(angle = 45, hjust=1))

barEdu

#Count by occupation

barOcc <- ggplot(adultData, aes(x = occupation)) + geom\_bar(color="black", fill="white") +

ggtitle("Occupation Bar Chart") + theme(axis.text.x = element\_text(angle = 45, hjust=1))

barOcc

#Count by native country

barNCountry <- ggplot(adultData, aes(x = `native-country`)) + geom\_bar(color="black", fill="white") +

ggtitle("Native-Country Bar Chart") + theme(axis.text.x = element\_text(angle = 45, hjust=1))

barNCountry

#Count by native country excluding US

barNCountry1 <- ggplot(subset(adultData, `native-country` != "United-States"), aes(x = `native-country`)) + geom\_bar(color="black", fill="white") +

ggtitle("Native-Country (Excluding US) Bar Chart") + theme(axis.text.x = element\_text(angle = 45, hjust=1))

barNCountry1

####################################################################

############################ Pie Charts ############################

####################################################################

#Pie chart of education

pieEdu <- ggplot(adultData, aes(x = "", fill = factor(education))) +

geom\_bar(width = 1) +

theme(axis.line = element\_blank(),

plot.title = element\_text(hjust=0.5)) +

labs(fill="education",

x=NULL,

y=NULL,

title="Pie Chart of Education",

caption="Source: adultData") +

coord\_polar(theta = "y", start=0)

pieEdu

pieEdu + facet\_wrap(~ income)

#Pie chart of workclass

pieWork <- ggplot(adultData, aes(x = "", fill = factor(workclass))) +

geom\_bar(width = 1) +

theme(axis.line = element\_blank(),

plot.title = element\_text(hjust=0.5)) +

labs(fill="workclass",

x=NULL,

y=NULL,

title="Pie Chart of Workclass",

caption="Source: adultData") +

coord\_polar(theta = "y", start=0)

pieWork

pieWork + facet\_wrap(~ education)

#Pie chart of Race

pieRace <- ggplot(adultData, aes(x = "", fill = factor(race))) +

geom\_bar(width = 1) +

theme(axis.line = element\_blank(),

plot.title = element\_text(hjust=0.5)) +

labs(fill="race",

x=NULL,

y=NULL,

title="Pie Chart of Race",

caption="Source: adultData") +

coord\_polar(theta = "y", start=0)

pieRace

#Pie chart of Occupation

pieOcc <- ggplot(adultData, aes(x = "", fill = factor(occupation))) +

geom\_bar(width = 1) +

theme(axis.line = element\_blank(),

plot.title = element\_text(hjust=0.5)) +

labs(fill="occupation",

x=NULL,

y=NULL,

title="Pie Chart of Occupation",

caption="Source: adultData") +

coord\_polar(theta = "y", start=0)

PieOcc

pieOcc + facet\_wrap(~ education)

grid.arrange(pieEdu, pieWork, pieOcc, ncol = 3)

############################################################################

########################### Build model with KSVM ##########################

############################################################################

library(kernlab)

#Creating a random index of the dataset

randIndex <- sample(1:dim(adultData)[1])

head(randIndex)

#Creating cutpoint

cutPoint <- floor(2 \* dim(adultData)[1]/3)

cutPoint

#Creating train dataset

trainData <- adultData[randIndex[1:cutPoint],]

nrow(trainData)

#Creating test dataset

testData <- adultData[randIndex[(cutPoint + 1):dim(adultData)[1]],]

nrow(testData)

#Validating lengths

nrow(adultData) == nrow(trainData) + nrow(testData)

# 1) Build a model using KSVM to predict income

outputKSVM <- ksvm(income ~., data = trainData, kernel = "rbfdot", kpar = "automatic", C = 50,

cross = 3, prob.model = TRUE)

# 2) Test model on testing dataset and compute percent of income correctly predicted

predKSVM <- predict(outputKSVM, testData, type = "votes")

#Combining Original and Prediction

compTableKSVM <- data.frame(testData[,13], predKSVM[1,])

View(predKSVM)

#Rename columns

colnames(compTableKSVM) <- c("income", "predKSVM")

head(compTableKSVM)

#Creating confusion matrix

(cfmtrx <- table(compTableKSVM))

#Compute the percentage correct

(pCor <- ((cfmtrx[1,2] + cfmtrx[2,1]) /

(cfmtrx[1,1] + cfmtrx[1,2] + cfmtrx[2,1] + cfmtrx[2,2])

\* 100))

# 75.84046%

#Storing the error difference as a column

compTableKSVM$Correct <- ifelse(compTableKSVM$predKSVM == 1, "Correct", "Incorrect")

#Creating New Dataframe

dfKSVM <- data.frame(testData, compTableKSVM$predKSVM, compTableKSVM$Correct)

#Rename columns

colnames(dfKSVM)

colnames(dfKSVM) <- c("age", "workclass", "fnlwgt", "education", "education-num", "marital-status",

"occupation", "relationship", "race", "sex", "hours-per-week", "native-country",

"income", "predKSVM", "Correct")

head(dfKSVM)

#Building model without using all variables

# 1) Build a model using KSVM1 to predict income

outputKSVM1 <- ksvm(income ~ education + occupation + workclass, data = trainData, kernel = "rbfdot",

kpar = "automatic", C = 50, cross = 3, prob.model = TRUE)

# 2) Test model on testing dataset and compute percent of income correctly predicted

predKSVM1 <- predict(outputKSVM1, testData, type = "votes")

#Combining Original and Prediction

compTableKSVM1 <- data.frame(testData[,13], predKSVM1[1,])

View(predKSVM1)

#Rename columns

colnames(compTableKSVM1) <- c("income", "predKSVM1")

head(compTableKSVM1)

#Creating confusion matrix

(cfmtrx1 <- table(compTableKSVM1))

#Compute the percentage correct

(pCor1 <- ((cfmtrx1[1,2] + cfmtrx1[2,1]) /

(cfmtrx1[1,1] + cfmtrx1[1,2] + cfmtrx1[2,1] + cfmtrx1[2,2])

\* 100))

# 78.44639%

###JT###

#useful binary variable

adultData$relationship.num <- recode(adultData$relationship,'Wife' = 1,'Own-child' = 2,'Husband' =3,'Not-in-family' = 4,'Other-relative' = 5,'Unmarried' = 6)

ggplot(data = melt(adultData), mapping = aes(x = value)) + geom\_histogram(bins = 20) + facet\_wrap(~variable, scales = 'free\_x')

adultData$income.num <-ifelse(adultData$income=='>50K',1,0)

adultData$sex.num <- ifelse(adultData$sex =='Male',1,0) #made male =1 female0

head(adultData)

adultData3 <- as.factor(adultData$income)

View(adultData)

#basic histogram

numvar = adultData$age

adultHistogram <- ggplot(adultData) + aes(x=as.numeric(numvar)) + geom\_histogram(binwidth=1,color="black")

adultHistogram

adultHistDens <- ggplot(adulData) + aes(x=as.numeric(numvar))+geom\_density()

adultHistDens

#density outline of the histogram shows that there are two modes in the age spectrum, but looking closer,

#there is one true mode

skewness(numvar)

#this equals 0.5586919, showing skewness to the right

#histogram of age by income group

ggplot(adultData) + aes(x=as.numeric(age), group=income, fill=income) +

geom\_histogram(binwidth=1, color='black')

#hist of age by gender group

ggplot(adultData) + aes(x=as.numeric(age), group=sex, fill=sex) + geom\_histogram(binwidth=1, color='black')

#histogram for hours per week

hist(adultData$`hours-per-week`, main='Hours per Week Histogram', xlab="Hours per Week")

hoursperweekhist <- ggplot(adultData) +

#######workclass edit for grouping

summary(adultData$workclass)

levels(adultData$workclass)[1] <- 'Unknown'

# combine into Government job

adultData$workclass <- gsub('^Federal-gov', 'Government', adultData$workclass)

adultData$workclass <- gsub('^Local-gov', 'Government', adultData$workclass)

adultData$workclass <- gsub('^State-gov', 'Government', adultData$workclass)

# combine into Self-Employed job

adultData$workclass <- gsub('^Self-emp-inc', 'Self-Employed', adultData$workclass)

adultData$workclass <- gsub('^Self-emp-not-inc', 'Self-Employed', adultData$workclass)

# combine into Other/Unknown

adultData$workclass <- gsub('^Never-worked', 'Other', adultData$workclass)

adultData$workclass <- gsub('^Without-pay', 'Other', adultData$workclass)

adultData$workclass <- gsub('^Other', 'Other/Unknown', adultData$workclass)

adultData$workclass <- gsub('^Unknown', 'Other/Unknown', adultData$workclass)

adultData$workclass <- as.factor(adultData$workclass)

summary(adultData$workclass)

#now there are 4 categories for workclass, including Government, Private, Self-Employed, and Other/Unknown.

#this will help simplify our visuals for workclass and income influences

#now i am going to explore the relationship between industry and income

#with some bar charts comparing our four new categories for workclass

#first we will do the barplot by categories for each income group.

# get the counts by industry and income group and format into a separate table

count <- table(adultData[adultData$workclass == 'Government',]$income)["<=50K"]

count <- c(count, table(adultData[adultData$workclass == 'Government',]$income)[">50K"])

count <- c(count, table(adultData[adultData$workclass == 'Other/Unknown',]$income)["<=50K"])

count <- c(count, table(adultData[adultData$workclass == 'Other/Unknown',]$income)[">50K"])

count <- c(count, table(adultData[adultData$workclass == 'Private',]$income)["<=50K"])

count <- c(count, table(adultData[adultData$workclass == 'Private',]$income)[">50K"])

count <- c(count, table(adultData[adultData$workclass == 'Self-Employed',]$income)["<=50K"])

count <- c(count, table(adultData[adultData$workclass == 'Self-Employed',]$income)[">50K"])

count <- as.numeric(count)

#create a dataframe for this table

adultindustry <- rep(levels(adultData$workclass), each = 2)

adultincome <- rep(c('<=50K', '>50K'), 4)

newdf <- data.frame(adultindustry, adultincome, count)

newdf

#results combined with count

# adultindustry adultincome count

#1 Other/Unknown <=50K 0

#2 Other/Unknown >50K 0

#3 Other/Unknown/Unknown <=50K 2393

#4 Other/Unknown/Unknown >50K 953

#5 Private <=50K 17410

#6 Private >50K 4876

#7 Self-Employed <=50K 2259

#8 Self-Employed >50K 1314

#add the plyr package for this new percentage from the count column

library(plyr)

#calculate the percentages for each section

newdf <- ddply(newdf, .(adultindustry), transform, percent = count/sum(count) \* 100)

#format the labels and calculate their positions with our percentages we created

newdf <- ddply(newdf, .(adultindustry), transform, pos = (cumsum(count) - 0.5 \* count))

newdf$label <- paste0(sprintf("%.0f", newdf$percent), "%")

#new bar plot of counts by each industry with in their income groups

ggplot(newdf, aes(x = adultindustry, y = count, fill = adultincome)) +

geom\_bar(stat = "identity") +

geom\_text(aes(y = pos, label = label), size = 2) +

ggtitle('Income by Industry')

####based on this vis, those who are self-employed have the highest tendency

#representation of number of people with income over 50

#education and age are used as explanatory variables

adultData2 = adultData %>%

group\_by(education,age)%>%

summarise(

over50k = sum(income.num, na.rm=TRUE)/n()

)

num1=adultData2$age

num2=adultData2$over50k

num1

num2

educationAge <- ggplot(adultData2, aes(num1,num2))

educationAge + geom\_point(aes(color=education))

#lm for education and income, then also do regression analysis

sexHours <- lm(adultData$sex.num ~ adultData$`hours-per-week`)

predict(sexHours, adultData, type = "response")

summary(sexHours)

educationIncome <- lm(adultData$income.num ~ adultData$`education-num`)

predict(educationIncome, adultData, type = "response")

summary(educationIncome)

#lm sex ed hours

incomeSexEdHours <- lm(income.num ~ `education-num` + sex.num +`hours-per-week`,data = adultData)

predict(incomeSexEdHours, adultData, type = "response")

summary(incomeSexEdHours)

incomeAgeHoursEd <- lm(adultData$income.num ~ adultData$age + adultData$`hours-per-week` + adultData$`education-num` + adultData$relationship.num)

predict(incomeAgeHoursEd, adultData, type = "response")

summary(incomeAgeHoursEd)

library(tidyverse)

library(rworldmap)

#world map

#https://blog.learningtree.com/how-to-display-data-on-a-world-map-in-r/

adultData$`native-country` <- recode(adultData$`native-country`,'United-States' = 'USA','England' = 'GBR','Columbia' = 'COL', 'Canada' = 'CAN','Jamaica' = 'JAM', 'Italy' = 'ITA', 'Germany' = 'DEU', 'China' = 'CHN', 'Cuba' = 'CUB', 'Honduras' = 'HMD', 'India' = 'IND', 'Iran' = 'IRN', 'Mexico' = 'MEX', 'Philippines' = 'PHL', 'Poland' = 'POL', 'Puerto-Rico' = 'PRI', 'Japan' = 'JPN', 'Taiwan' = 'TWN', 'Ireland' = 'IRL', 'France'= 'FRA', 'Hungary' = 'HUN', 'Ecuador' = 'ECU', 'Guatemala' = 'GTM', 'Dominican-Republic' = 'DOM', 'Peru' = 'PER' ,'Yugoslavia' = 'MKD', 'Trinadad&Tobago'= 'TTO', 'El-Salador' = 'SLV', 'Haiti' = 'HTI', 'Thailand' = 'THA' )

adultData$`native-country`

mapped\_data <- joinCountryData2Map(adultData, joinCode = "ISO3", nameJoinColumn = "native-country")

par(mai=c(0,0,0.2,0),xaxs="i",yaxs="i")

mapCountryData(mapped\_data, nameColumnToPlot = "education", catMethod = "categorical", colourPalette = "terrain")

View(adultData)

view(adultData3)

#using radom Forest to guess income based on all categories

randomForAdult <- randomForest(adultData3[,-13], adultData3[,13])

randomForAdult

importance(randomForAdult)