Calla Code

Final Project

install.packages("tidyverse")

install.packages("sqldf")

install.packages("dplyr")

Install.packages(“scales”)

Install.packages(“ggplot2”)

Install.packages(“caret”)

library(tidyverse)

library(sqldf)

library(dplyr)

library(scales)

library(ggplot2)

library(scales)

# Load data and assign to new variables

app\_record <-read.csv("/Users/gozi/Downloads/application\_record.csv")7

credit\_record <-read.csv("/Users/gozi/Downloads/credit\_record.csv")

cr <- credit\_record

ar <- app\_record

# Add an age and employment lenth column to the application record dataframe and transform column to integer

ar$AGE <- ar$DAYS\_BIRTH / 365

ar$AGE <- lapply(ar$AGE, as.integer)

ar$emp\_length = ar$DAYS\_EMPLOYED/365

ar$emp\_length <- lapply(ar$DAYS\_EMPLOYED, as.integer)

# Converts the cr status column into a factor and orders it. It then creates a simple vector of the status levels, plus a numerical vector that will then be matched with the status levels:

cr$STATUS <- factor(cr$STATUS, levels=c("X", "C", "0", "1", "2", "3", "4", "5"))

status\_factor\_level <- c("X", "C", "0", "1", "2", "3", "4", "5")

status\_level <- c(0, 0, 1, 2, 3, 4, 5, 6)

cr$status\_level <- status\_level[match(cr$STATUS, status\_factor\_level)]

glimpse(ar)

#merge the files into one

Both <- merge(ar, cr, by="ID")

#Data exploration

ar %>% count(ID)

table(ar['CODE\_GENDER'])

glimpse(cr)

cr\_history <- cr %>% group\_by(ID) %>% summarize(n\_obs=n())

hist(cr\_history$n\_obs)

mean(cr\_history$n\_obs)

# View the months a balance was carried and the average monthhs a balance was carried

cr\_history <- cr %>% group\_by(ID) %>% summarize(n\_obs=n())

hist(cr\_history$n\_obs)

mean(cr\_history$n\_obs)

#View data on cr based on status

status\_hist <- cr %>% group\_by(STATUS) %>% summarize(n\_obs=n())

status\_hist

#Comparing housing with income

ar %>%

group\_by( NAME\_HOUSING\_TYPE) %>%

summarise(AVG\_INCOME =mean(AMT\_INCOME\_TOTAL))

#Comparing number of children with income

ar %>%

group\_by(CNT\_CHILDREN) %>%

summarise(mean(AMT\_INCOME\_TOTAL))

#plot to show number of children with income level

ar %>%

ggplot(aes(x = CNT\_CHILDREN , y= AMT\_INCOME\_TOTAL)) +

geom\_point(color = "blue")

#Graphs to show average months balance based on number of children

no\_kids <- kids0 %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

no\_kids

hist(no\_kids$avgMonthsBalance, main="Avg Months Balance of Applicants with 0 kids", xlab="Avg Months Balance", col="pink")

kids1 <- ar\_cr %>%

filter(CNT\_CHILDREN == (1:5))

kids1

kids1\_5 <- kids1 %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

kids1\_5

hist(kids1\_5$avgMonthsBalance, main="Avg Months Balance of Applicants with 1 -5 kids", xlab="Avg Months Balance", col="pink")

kids6 <- ar\_cr %>%

filter(CNT\_CHILDREN == (6:10))

kids6

kids6\_10 <- kids6 %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

kids6\_10

hist(kids6\_10$avgMonthsBalance, main="Avg Months Balance of Applicants with 6 - 10 kids", xlab="Avg Months Balance", col="pink")

kids12 <- ar\_cr %>%

filter(CNT\_CHILDREN > (10))

kids12

kids12 <- kids12 %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

kids12

hist(kids12$avgMonthsBalance, main="Avg Months Balance of Applicants with more than 10 kids", xlab="Avg Months Balance", col="pink")

ar\_cr %>%

ggplot(aes(x= NAME\_INCOME\_TYPE, y = mean(MONTHS\_BALANCE, fill = CODE\_GENDER)) +

scale\_y\_reverse() +

geom\_bar(stat = "identity") +

xlab("Income Type ") +

ylab("Months") +

ggtitle("Income vs Months Balance is Carried")

# bar graph of breakdown of dataset by gender, first created Individual ID subset to create the graph

IndividualIDS <- Both %>%

distinct(ID, CODE\_GENDER)

ggplot(IndividualIDS, aes(x=CODE\_GENDER, fill=CODE\_GENDER)) + geom\_bar() + scale\_fill\_manual("Gender", values = c("F" = "pink", "M" = "blue")) + ggtitle("Number of Females and Males in the Dataset") + geom\_text(stat='count', aes(label=..count..))

# histogram displaying total number of records of each status

ggplot(Both, aes(x=STATUS,y = ..count..), stat = "count") + geom\_bar(aes(fill= STATUS)) + ggtitle("Number of Records for Each Status") + xlab("Status") + geom\_text(aes(label = ..count..), stat = "count", vjust=-0.5) + scale\_fill\_brewer(palette = "Pastel1")

# creating the histogram that displays the average credit history length of the dataset

hist(AvgCreditHistory$avgMonthsBalance, main = "Average Credit History for All Applicants", xlab = "Average Months Balance", col = "royalblue3")

# creating bar graph comparing the months balance for each credit status by education level to see what level of education people with good vs bad credit have

Both %>% ggplot() + geom\_bar(aes(x= NAME\_EDUCATION\_TYPE, y= MONTHS\_BALANCE, fill= STATUS), stat ="identity", position = "dodge") + scale\_y\_reverse() + scale\_fill\_brewer(palette = "PRGn") + ggtitle("Avg Months Balance for Each Credit Status by Education Level") + xlab("Education Level") + ylab("Avg Months Balance")

#looking into average credit of each status group, first create matrix for each then graphs

StatusC <- Both %>%

filter(STATUS == "C")

crC <- StatusC %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

hist(crC$avgMonthsBalance, main="Avg Months Balance of Status C Applicants", xlab="Avg Months Balance", col="green")

StatusX <- Both %>%

filter(STATUS == "X")

cr5 <- StatusX %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

hist(cr5$avgMonthsBalance, main="Avg Months Balance of Status X Applicants", xlab="Avg Months Balance", col="tomato")

Status0 <- Both %>%

filter(STATUS == "0")

cr5 <- Status5 %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

hist(cr0$avgMonthsBalance, main="Avg Months Balance of Status 0 Applicants", xlab="Avg Months Balance", col="tomato")

Status1 <- Both %>%

filter(STATUS == "1")

cr1 <- Status1 %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

hist(cr1$avgMonthsBalance, main="Avg Months Balance of Status 1 Applicants", xlab="Avg Months Balance", col="tomato")

Status5 <- Both %>%

filter(STATUS == "1")

cr1 <- Status1 %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

hist(cr1$avgMonthsBalance, main="Avg Months Balance of Status 1 Applicants", xlab="Avg Months Balance", col="tomato")

Status2 <- Both %>%

filter(STATUS == "5")

cr2 <- Status2 %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

hist(cr2$avgMonthsBalance, main="Avg Months Balance of Status 2 Applicants", xlab="Avg Months Balance", col="tomato")

Status3 <- Both %>%

filter(STATUS == "3")

cr3 <- Status3 %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

hist(cr3$avgMonthsBalance, main="Avg Months Balance of Status 3 Applicants", xlab="Avg

Months Balance", col="tomato")

Status4 <- Both %>%

filter(STATUS == "3")

cr4 <- Status4 %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

hist(cr4$avgMonthsBalance, main="Avg Months Balance of Status 4 Applicants", xlab="Avg Months Balance", col="tomato")

Status5 <- Both %>%

filter(STATUS == "5")

cr5 <- Status5 %>%

group\_by(ID) %>%

select(MONTHS\_BALANCE) %>%

summarise(avgMonthsBalance = mean(MONTHS\_BALANCE))

hist(cr5$avgMonthsBalance, main="Avg Months Balance of Status 5 Applicants", xlab="Avg Months Balance", col="tomato")

#boxplot of ages of the status

ggplot(Both, aes(group = STATUS, x = STATUS, y = AGE, fill = STATUS)) + geom\_boxplot() + coord\_flip() + ggtitle("Distribution of Age Grouped by Status ") + scale\_fill\_brewer(palette = "Pastel1")

#looked at boxplot of Employment time

ggplot(Both, aes(group = STATUS, x = STATUS, y = EmploymentTime, fill = STATUS)) + geom\_boxplot() + coord\_flip() + ggtitle("Distribution of Employment Grouped by Status ") + scale\_fill\_brewer(palette = "Pastel1")

#graph looked bad, so split into employed and unemployed

Employed <- Both %>%

filter(emp\_length > 0)

Unemployed <- Both %>%

filter(emp\_length <= 0)

#the unemployed graph looked messed up, appears everyone who was a pensioner got same time so only included the employed group boxplot

ggplot(Employed, aes(group = STATUS, x = STATUS, y = EmploymentTime, fill = STATUS)) + geom\_boxplot() + coord\_flip() + ggtitle("Distribution of Employment Grouped by Status ") + scale\_fill\_brewer(palette = "Pastel1")

ggplot(Unemployed, aes(group = STATUS, x = STATUS, y = EmploymentTime, fill = STATUS)) + geom\_boxplot() + coord\_flip() + ggtitle("Distribution of Employment Grouped by Status ") + scale\_fill\_brewer(palette = "Pastel1")

#Exploring status and income by gender

Both %>%

group\_by(NAME\_INCOME\_TYPE) %>%

ggplot() + geom\_bar(aes(x= NAME\_EDUCATION\_TYPE, y= MONTHS\_BALANCE, fill= STATUS), stat ="identity", position = "dodge") + scale\_y\_reverse() + scale\_fill\_brewer(palette = "PRGn") + ggtitle("Avg Months Balance for Each Credit Status by Education Level") + xlab("Education Level") + ylab("Avg Months Balance")

#create good and bad applicant groups

GoodApplicants <- Both %>%

filter(STATUS %in% c("X", "C", "0"))

BadApplicants <- Both %>%

filter(STATUS %in% c("1", "2", "3", "4", "5"))

#look and see if number of children and months balance are correlated individually looking at the good/bad applicant groups

#both graphs really did not reveal anything so not including

GoodApplicants %>%

ggplot() + geom\_jitter(aes(x= CNT\_CHILDREN, y= MONTHS\_BALANCE, color = STATUS )) + scale\_y\_reverse()

BadApplicants %>%

ggplot() + geom\_jitter(aes(x= CNT\_CHILDREN, y= MONTHS\_BALANCE, color = STATUS )) + scale\_y\_reverse()

#Creates a summary table that shows what the worst level of delinquency each person reached, then generates a pie chart of the results:

highest\_dlq <- cr %>% group\_by(ID) %>% summarize(max\_dlq=max(status\_level))

highest\_dlq\_pie <- highest\_dlq %>% group\_by(max\_dlq) %>% summarize(count=n())

highest\_dlq\_pie$labels <- highest\_dlq\_pie$max\_dlq - 1

highest\_dlq\_pie$labels <- replace(highest\_dlq\_pie$labels, highest\_dlq\_pie$labels<0, "X/C")

highest\_dlq\_pie$labels <- factor(highest\_dlq\_pie$labels, levels=c("X/C", "0", "1", "2", "3", "4", "5"))

ggplot(highest\_dlq\_pie, aes(x="", y=count, fill=labels)) +

geom\_bar(stat="identity", width=1) +

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

scale\_fill\_manual(values=c("blue", "green", "yellow", "orange", "red", "purple", "black")) +

guides(fill=guide\_legend(title="Delinquency Status \n(Lower is Better)")) +

theme(axis.title.x=element\_blank(), axis.title.y=element\_blank()) +

ggtitle("Worst Delinquency Level Reached")

# Creates a data table with the percentages in each of the highest dlq buckets.

select(highest\_dlq\_pie, labels, count) %>% mutate(percent=scales::percent(round(count/sum(count), 3)))

# Creates a dataframe of where each person's worst delinquency level was, and where their last observation was.

status\_df <- sqldf("SELECT cr.ID, cr.MONTHS\_BALANCE, cr.STATUS, cr.status\_level, highest\_dlq.max\_dlq

FROM cr

INNER JOIN highest\_dlq ON cr.ID = highest\_dlq.ID

WHERE MONTHS\_BALANCE = 0")

status\_df <- status\_df %>% group\_by(max\_dlq, STATUS) %>% summarize(Observations=n())

# Change the labels for the x-axis from numeric to their actual names

status\_df$xlabels <- status\_df$max\_dlq - 1

status\_df$xlabels <- replace(status\_df$xlabels, status\_df$xlabels<0, "X/C")

status\_df$xlabels <- factor(status\_df$xlabels, levels=c("X/C", "0", "1", "2", "3", "4", "5"))

#Make the stacked bar chart

ggplot(status\_df) +

aes(fill=STATUS, y=Observations, x=xlabels) +

geom\_col(position="fill", stat="identity") +

scale\_fill\_manual(values=c("light blue", "blue", "green", "yellow", "orange", "red", "purple", "black")) +

ylab("Percentage of Cohort") +

guides(fill=guide\_legend(title="Final Status \n(Lower is Better)")) +

xlab("(<-- Better) Worst level of Delinquency Reached (Worse -->)") +

ggtitle("Breakdown of Final Observation Based on \nHighest Delinquency Level Reached")

# Creates a table of "good" payers (worst dlq = X, C, or 0) and one of "bad" payers (worst delinquency of 1-5)

```{r}

good\_payers <- sqldf("SELECT ar.ID, ar.AGE, ar.NAME\_HOUSING\_TYPE, ar. AMT\_INCOME\_TOTAL, ar.NAME\_EDUCATION\_TYPE, ar.DAYS\_EMPLOYED, ar.FLAG\_OWN\_CAR, ar.FLAG\_OWN\_REALTY, ar.NAME\_FAMILY\_STATUS, ar.CODE\_GENDER

FROM ar

INNER JOIN highest\_dlq ON ar.ID = highest\_dlq.ID

WHERE highest\_dlq.max\_dlq IN (0, 1)")

bad\_payers <- sqldf("SELECT ar.ID, ar.AGE, ar.NAME\_HOUSING\_TYPE, ar. AMT\_INCOME\_TOTAL, ar.NAME\_EDUCATION\_TYPE, ar.DAYS\_EMPLOYED, ar.FLAG\_OWN\_CAR, ar.FLAG\_OWN\_REALTY, ar.NAME\_FAMILY\_STATUS, ar.CODE\_GENDER

FROM ar

INNER JOIN highest\_dlq ON ar.ID = highest\_dlq.ID

WHERE highest\_dlq.max\_dlq IN (2, 3, 4, 5, 6)")

# Compares the good\_payers and bad\_payers dataframes

good\_payers %>% group\_by(NAME\_EDUCATION\_TYPE) %>% summarize(count=n()) %>%

ggplot(aes(x="", y=count, fill=NAME\_EDUCATION\_TYPE)) +

geom\_bar(stat="identity", width=1) +

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

ggtitle("Education Breakdown of 'Good' Payers")

bad\_payers %>% group\_by(NAME\_EDUCATION\_TYPE) %>% summarize(count=n()) %>%

ggplot(aes(x="", y=count, fill=NAME\_EDUCATION\_TYPE)) +

geom\_bar(stat="identity", width=1) +

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

ggtitle("Education Breakdown of 'Bad' Payers")

df <- good\_payers %>%

group\_by(FLAG\_OWN\_REALTY, FLAG\_OWN\_CAR) %>%

summarize(count=n())

df$FLAG\_OWN\_REALTY[df$FLAG\_OWN\_REALTY=="N"]<-"No realty"

df$FLAG\_OWN\_REALTY[df$FLAG\_OWN\_REALTY=="Y"]<-"Realty"

df$FLAG\_OWN\_CAR[df$FLAG\_OWN\_CAR=="N"]<-"No car"

df$FLAG\_OWN\_CAR[df$FLAG\_OWN\_CAR=="Y"]<-"Car"

df <- df %>% mutate(Ownership=paste(FLAG\_OWN\_REALTY, FLAG\_OWN\_CAR, sep = " / "))

df %>% ggplot(aes(x="", y=count, fill=Ownership)) +

geom\_bar(stat="identity", width=1) +

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

ggtitle("Ownership Breakdown of 'Good' Payers")

df <- bad\_payers %>%

group\_by(FLAG\_OWN\_REALTY, FLAG\_OWN\_CAR) %>%

summarize(count=n())

df$FLAG\_OWN\_REALTY[df$FLAG\_OWN\_REALTY=="N"]<-"No realty"

df$FLAG\_OWN\_REALTY[df$FLAG\_OWN\_REALTY=="Y"]<-"Realty"

df$FLAG\_OWN\_CAR[df$FLAG\_OWN\_CAR=="N"]<-"No car"

df$FLAG\_OWN\_CAR[df$FLAG\_OWN\_CAR=="Y"]<-"Car"

df <- df %>% mutate(Ownership=paste(FLAG\_OWN\_REALTY, FLAG\_OWN\_CAR, sep = " / "))

df %>% ggplot(aes(x="", y=count, fill=Ownership)) +

geom\_bar(stat="identity", width=1) +

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

ggtitle("Ownership Breakdown of 'Bad' Payers")

# Building the prediction models

modelDF <- select(ar, FLAG\_OWN\_CAR, FLAG\_OWN\_REALTY, AMT\_INCOME\_TOTAL,

NAME\_EDUCATION\_TYPE, NAME\_INCOME\_TYPE, AGE,

CNT\_FAM\_MEMBERS, CODE\_GENDER, goodBad)

# Changes all character variables to factors, plus the dependent variable "goodBad"

modelDF <- filter(modelDF, is.na(modelDF$goodBad)==FALSE)

modelDF$goodBad <- as.factor(modelDF$goodBad)

modelDF$FLAG\_OWN\_CAR <- as.factor(modelDF$FLAG\_OWN\_CAR)

modelDF$FLAG\_OWN\_REALTY <- as.factor(modelDF$FLAG\_OWN\_REALTY)

modelDF$NAME\_EDUCATION\_TYPE <- as.factor(modelDF$NAME\_EDUCATION\_TYPE)

modelDF$NAME\_INCOME\_TYPE <- as.factor(modelDF$NAME\_INCOME\_TYPE)

modelDF$CODE\_GENDER <- as.factor(modelDF$CODE\_GENDER)

set.seed(123)

trainList <- createDataPartition(y=modelDF$goodBad, p=0.70, list=FALSE)

trainSet <- modelDF[trainList, ]

testSet <- modelDF[-trainList, ]

svmModel <- svm(goodBad ~ ., data=trainSet, type="C-classification", kernel="radial", C=1.0, gamma=0.1)

pred\_svm <- predict(svmModel, newdata=testSet, type="class")

confusionMatrix(pred\_svm, testSet$goodBad)

rpartModel <- train(goodBad ~ ., data=trainSet, method="rpart", preProc=c("center","scale"))

pred\_rpart <- predict(rpartModel, newdata=testSet, type="raw")

confusionMatrix(pred\_rpart, testSet$goodBad)

Filter out the "bad candidates" who have STATUS 2, 3, 4, and 5.

```{r}

df\_dlq <- df %>% filter(STATUS == 2 | STATUS == 3 | STATUS == 4 | STATUS == 5)

head(df\_dlq)

```

Find columns with data that are not needed

```{r}

# Everyone had a mobile phone so this data does not help us.

unique(df\_dlq$FLAG\_MOBIL)

df\_dlq <- df\_dlq %>% subset(select = -FLAG\_MOBIL)

head(df\_dlq)

```

Fix up DAYS\_BIRTH and DAYS\_EMPOYED numbers

```{r}

# Make Days birth positive and turn it into years to get age

df\_dlq$DAYS\_BIRTH <- abs(df\_dlq$DAYS\_BIRTH)

df\_dlq$DAYS\_BIRTH <- df\_dlq$DAYS\_BIRTH/365

# Unemployed people have DAYS\_EMPLOYED set to 365243

df\_dlq %>% filter(DAYS\_EMPLOYED > 0)

nrow(df\_dlq %>% filter(DAYS\_EMPLOYED > 0)) # 525 people are unemployed

# Set unemployed to 0

df\_dlq$DAYS\_EMPLOYED[df\_dlq$DAYS\_EMPLOYED > 0 ] <- 0

nrow(df\_dlq %>% filter(DAYS\_EMPLOYED == 0)) # make sure 525 people have value 0

# Get employed people's days to positive using absolute value

df\_dlq$DAYS\_EMPLOYED <- abs(df\_dlq$DAYS\_EMPLOYED)

# Get Days Employed into Years Employed

df\_dlq$DAYS\_EMPLOYED <- df\_dlq$DAYS\_EMPLOYED/365

```

Histogram of Age of people with bad credit

```{r}

df\_dlq %>%

ggplot() + geom\_histogram(aes(x=DAYS\_BIRTH), color="black", fill="grey") +

xlab("Age") +

ggtitle("Age of People with Bad Credit")

df\_dlq %>%

ggplot() +

geom\_boxplot(aes(y=DAYS\_BIRTH), color = "black", fill="white")

# stat\_summary(fun="mean", geom="point", color="red")

```

Histogram of People of bad credit and how many years they were employed

```{r}

df\_dlq %>%

ggplot() + geom\_histogram(aes(x=DAYS\_EMPLOYED), color="black", fill="grey") +

xlab("Years\_Employed") +

ggtitle("Years Employed of People with Bad Credit")

```

Histogram of Number of Children People Have

```{r}

df\_dlq %>%

ggplot() + geom\_histogram(aes(x=CNT\_CHILDREN), color="black", fill="grey", bins=5) +

ggtitle("Number of Children People with Bad Credit Have")

```

Boxplot of Income

```{r}

df\_dlq %>%

ggplot() + geom\_boxplot(aes(y=AMT\_INCOME\_TOTAL))

```

Bar Chart for Owning Car and Property

```{r}

head(df\_dlq)

carRealty <- cbind(df\_dlq$FLAG\_OWN\_CAR, df\_dlq$FLAG\_OWN\_REALTY)

head(carRealty)

#yesNo <- c("Y", "N")

#yesOrNo <- c(yesNo, yesNo)

#df\_dlq %>%

# ggplot() + geom\_bar(aes(x=yesOrNo, fill=carRealty),stat="identity", color="black", position="dodge")

# geom\_bar(aes(x=FLAG\_WORK\_PHONE), color="black", fill="pink", position="dodge") +

# geom\_bar(aes(x=FLAG\_PHONE), color="black", fill="green", position="dodge") +

# geom\_bar(aes(x=FLAG\_EMAIL), color="black", fill="purple", position="dodge")

```

Bar Chart for Gender

```{r}

head(df\_dlq)

df\_dlq %>%

ggplot() +

geom\_bar(aes(x=CODE\_GENDER), color="black", fill="grey") +

ggtitle("Bar Chart of the Gender of People with Bad Credit")

```

Bar Chart for Occupation Type

```{r}

df\_dlq %>%

ggplot() +

geom\_bar(aes(x=OCCUPATION\_TYPE), color="black", fill="grey") +

theme(axis.text.x = element\_text(angle=90, hjust=1))

```

Bar Chart for Housing Type

```{r}

head(df\_dlq)

df\_dlq %>%

ggplot() +

geom\_bar(aes(x=NAME\_HOUSING\_TYPE), color="black", fill="pink") +

theme(axis.text.x = element\_text(angle=90, hjust=1))

```

Bar Chart for Marital Status

```{r}

head(df\_dlq)

df\_dlq %>%

ggplot() +

geom\_bar(aes(x=NAME\_FAMILY\_STATUS), color="black", fill="cyan")

```

Bar Chart for Education Type

```{r}

head(df\_dlq)

df\_dlq %>%

ggplot() +

geom\_bar(aes(x=NAME\_EDUCATION\_TYPE), color="black", fill="blue")

```

Bar Chart for Income Type

```{r}

head(df\_dlq)

df\_dlq %>%

ggplot() +

geom\_bar(aes(x=NAME\_INCOME\_TYPE), color="black", fill="green3")

```

Sana’s model code:

prediction\_df <- data.frame(gender = as.factor(ar\_cr$CODE\_GENDER),

income = ar\_cr$AMT\_INCOME\_TOTAL,

age = ar\_cr$AGE,

Employment = ar\_cr$emp\_length,

candidate\_state = as.factor(ar\_cr$candidate))

prediction\_df

set.seed(50)

training1 <- createDataPartition(y=prediction\_df$candidate\_state,p=.70,list=FALSE)

train <- prediction\_df[training1,]

test <- prediction\_df[-training1,]

dim(train)

dim(test)

model.rpart <- rpart(candidate\_state ~ ., data = train,

method = "class")

rpart.plot(model.rpart)

rpart <- predict(model.rpart, newdata = test,type = "class")

confusion\_rpart <- confusionMatrix(rpart, test$candidate\_state)

confusion\_rpart

confusion\_rpart$overall[1]