```
setwd("/Users/kashmirayadav/Desktop/BASDM IIT-Roorkee")

#### Importing the Data files

View(credit_record)
View(application_record)
app <- read.csv("Asignment 3/application_record.csv",
na.strings="")
cred<- read.csv("Asignment 3/credit_record.csv", na.strings =
"")
head(app,5)
head(cred,5)

#### Handling Missing Values and Duplicates
#### Checking For Missing Values and finding column-wise count
of missing values

sapply(cred,function(var){sum(is.na(var))})
sapply(app,function(var){sum(is.na(var))})

#### Finding missing values

app$OCCUPATION_TYPE<-NULL

#### Checking for duplicates in ID column and remove if any

sum(duplicated(app$ID))
app_nodupli<-app[!duplicated(app$ID,fromLast = T),]
nrow(app_nodupli)

sum(duplicated(cred$ID))
cred_nodupli<-cred[!duplicated(cred$ID,fromLast=T),]
nrow(cred_nodupli)

#### Feature Analysis
#### Checking percentage of IDs that are common between the
two tables

length(union(app_nodupli$ID,cred_nodupli$ID))

length(intersect(app_nodupli$ID,cred_nodupli$ID))/length(union
(app_nodupli$ID,cred_nodupli
                                                    $ID))*100

#### Visualizing the distribution of income types

table(app_nodupli$NAME_INCOME_TYPE)
pie(table(app_nodupli$NAME_INCOME_TYPE), col =
c("yellow","pink","purple","green","orange"))
barplot(table(app_nodupli$NAME_INCOME_TYPE),col=c("yellow","pi
nk","purple","green","orange"))
```

```
#### Finding the age in years and studying the age
distribution

Age<--(app_nodupli$DAYS_BIRTH/365)
hist(Age,col="pink")

plot(density(Age))

#### Studying the gender distribution in applicants' data

gender<-table(app_nodupli$CODE_GENDER)
print(gender)
Gender<-round(100*gender/sum(gender),1)
pie(gender,labels=Gender, col=c("pink","blue"))
legend("topright",c('Female' ,'Male'),fill=c("pink","blue"))

#### Studying the average annual income across different
education levels

income_edu<-
aggregate(app_nodupli$AMT_INCOME_TOTAL,by=list(app_nodupli$NAM
E_EDUCATION_TYPE),FUN=mean)
row.names(income_edu)<-income_edu$Group.1
income_edu$Group.1<-NULL
barplot(t(as.matrix(income_edu)))

#### Finding the count of applicants from different education
levels.
####  Also, showcasing how many of those own a car

table(app_nodupli$NAME_EDUCATION_TYPE)
table(app_nodupli$FLAG_OWN_CAR)

Own_car <- table(app_nodupli$NAME_EDUCATION_TYPE,
app_nodupli$FLAG_OWN_CAR)
print(Own_car)

#### Creating a stacked column chart to visualize the above
data

install.packages("data.table")
library(data.table)
install.packages("reshape2")
library(reshape2)

barplot(Own_car, beside = TRUE,
       legend.text = rownames(Own_car),
       col = c("Violet", "Purple"),
       main = "Education Level vs Car Ownership",
       xlab = "Education Level",
```

```
       ylab = "Count",
       args.legend = list(x = "topright", title = "Owns Car",
trace = TRUE))

#### Feature Creation
#### Convert "STATUS" to a binary variable called 'target',
where 0 is for good
#### applicants (where STATUS is either 'C', 'X' or '0') and 1
is for bad
#### applicants (where STATUS is either 1,2,3,4,5)

cred$Target<-ifelse(cred$STATUS %in% c('C','X','0'),0,1)
table(cred$Target)

#### cred#### Merging the two datasets to get the common
records

cred_aggre<-aggregate(cred$Target,list(cred$ID),sum)
head(cred_aggre)

cred_aggre$x<-ifelse(cred_aggre$x==0,0,1)
head(cred_aggre)
colnames(cred_aggre)<-c('ID','Target')

merged_data<-merge(app_nodupli,cred_aggre,by='ID',all=FALSE)
head(merged_data)

#### studying the data distribution in terms of the target

counts <- table(cred$Target)
percentages <- counts / sum(counts) * 100
pie(counts, labels = paste0(c('Good', 'Bad'), "\n", counts, "
(", round(percentages, 1), "%)"), col = c('green', 'red'),
main = 'Distribution of Good and Bad Applicants')

#### Data Preparation and Modeling
#### Converting categorical column into factors

catcols<-names(merged_data)[grepl('FLAG',names(merged_data))]
merged_data[catcols]<-lapply(merged_data[catcols],as.factor)
charcols<-names(which(sapply(merged_data,class)=='Character'))
str(merged_data)
merged_data[charcols] <- lapply(merged_data[charcols],
as.factor)

dropcols <- c(which(sapply(merged_data,function (x)
length(levels(x))== 1) ),
              which(names(merged_data)=="ID"))

levels(test_data$Target) <- levels(train_data$Target)
merged_data$Target <- as.factor(merged_data$Target)
```

```
merged_data <- na.omit(merged_data)

newdata <- merged_data[,-dropcols]
head(newdata)


#### Using this cleaned data to build a classification model

install.packages('caret')
library(caret)

#### Splitting the data into training and testing sets

set.seed(120)
train_index<-createDataPartition(newdata$Target,p=0.7,
list=FALSE)
train_data <- newdata[train_index, ]
test_data <- newdata[-train_index, ]
levels(test_data$Target) <- levels(train_data$Target)

set.seed(100)
trctrl <- trainControl(method = "cv",
                       number = 10,
                       savePredictions=TRUE) # 10 fold cross
validation

glm_fit <- train(Target ~ ., data = train_data, method =
"glm",
                 family = "binomial",
                 trControl=trctrl)

cv_mod <- glm_fit$finalModel

#### Predictions on test set

pred <- predict(glm_fit)
class <- predict(glm_fit,test_data, type = 'raw')
prob <- predict(glm_fit,test_data, type = 'prob')

#### Confusion Matrix on test set

confusionMatrix(class, test_data$Target, positive = '1')

#### ROC curve

library(pROC)

# Assuming roc_data contains the desired ROC curve
roc_data <- roc(test_data$Target, prob[, 2], positive = '1')

# Plot the ROC curve with AUC score
```

```
plot(roc_data, col = "blue", main = paste0('AUC Score = ',
round(auc(roc_data), 4)), xlim = c(0, 1))

# Create a data frame for plotting TPR vs FPR
fpr_tpr <- data.frame(TPR = roc_data$sensitivities, FPR = 1 -
roc_data$specificities)

# Plot TPR vs FPR
plot(fpr_tpr$FPR, fpr_tpr$TPR, lwd = 3, type = 'l', col =
'steelblue',
     main = paste0('AUC Score = ', round(auc(roc_data), 4)),
     xlab = '1 - Specificity (FPR)', ylab = 'Sensitivity
(TPR)', xlim = c(0, 1))

# Add a diagonal line
abline(0, 1, col = 'orange', lwd = 2)
```