

American International University-Bangladesh

(AIUB)

Department of Computer Science and Engineering (CSE)

Faculty of Science & Technology (FST)

Spring 2022-23

Course: Introduction to Data Science

Section: B

Final Term Project

**Submitted by:**

**Name: Sakib-Ul-Ahsan**

**ID: 20-42978-1**

**Department: CSE**

**Submitted to:**

**Tohedul Islam**

**Assistant Professor**

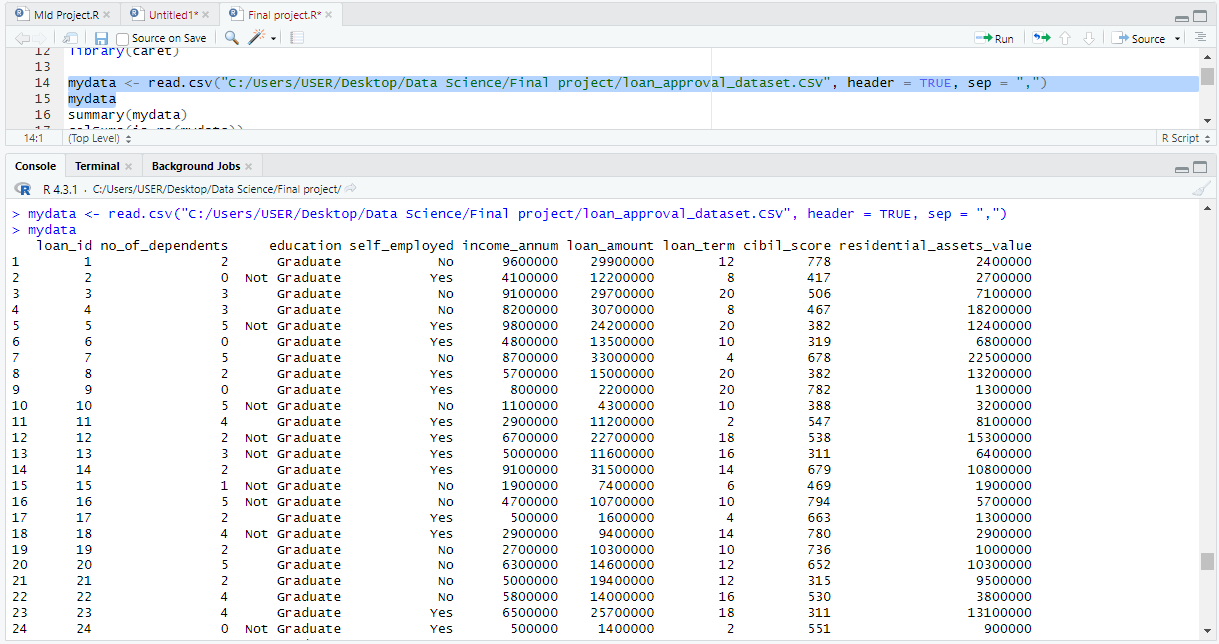
**Department of Computer Science**

**Dataset Description**: The dataset was collected from Kaggle. The name of the dataset is loan\_approval\_dataset. It was collected in csv format. It contains 13 attributes and 4269 instances. The attributes include loan\_id, no\_of\_dependents, education, self\_employed, loan\_amount, loan\_term, cibil\_score, residential\_assets\_value, commercial\_assets\_value, luxury\_assets\_value, bank\_asset\_value, loan\_status. The target attribute is named loan\_status. Which has only two values approved or rejected.

**Importing the dataset:**

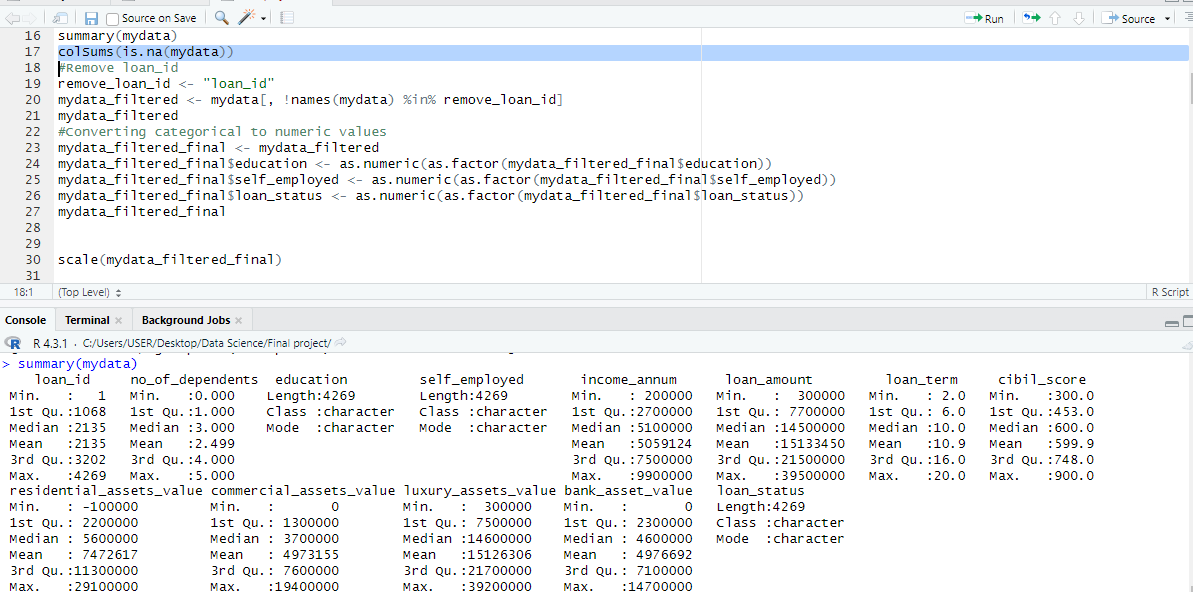
mydata <- read.csv("C:/Users/USER/Desktop/Data Science/Final project/loan\_approval\_dataset.CSV", header = TRUE, sep = ",")

mydata



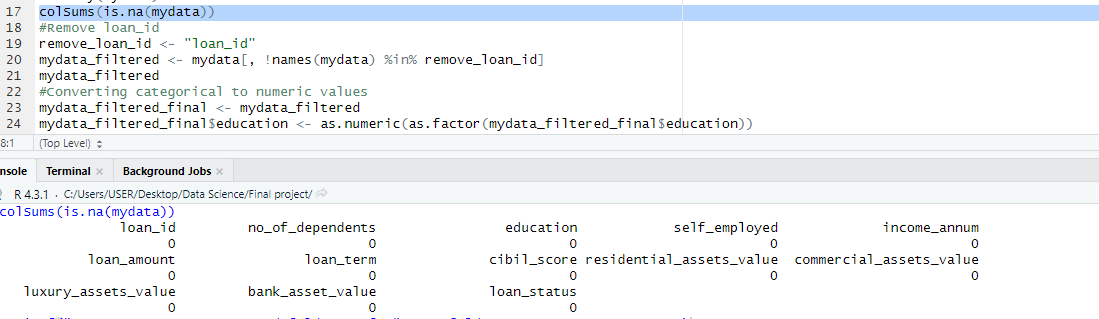
**Summary of all the attributes:**

summary(mydata)



**Look for missing values:**

colSums(is.na(mydata))



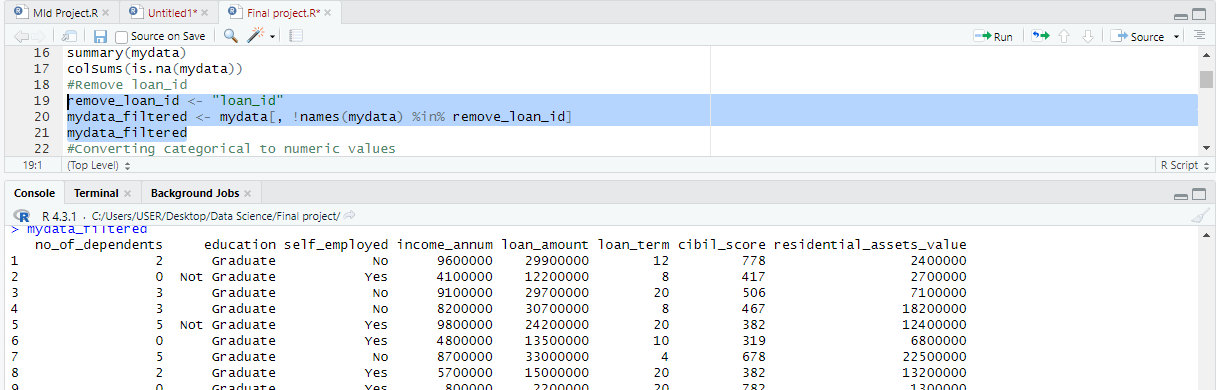
**Data Pre-processing:**

**Removing loan\_id attribute:**

remove\_loan\_id <- "loan\_id"

mydata\_filtered <- mydata[, !names(mydata) %in% remove\_loan\_id]

mydata\_filtered



**Converting categorical values to numeric values:**

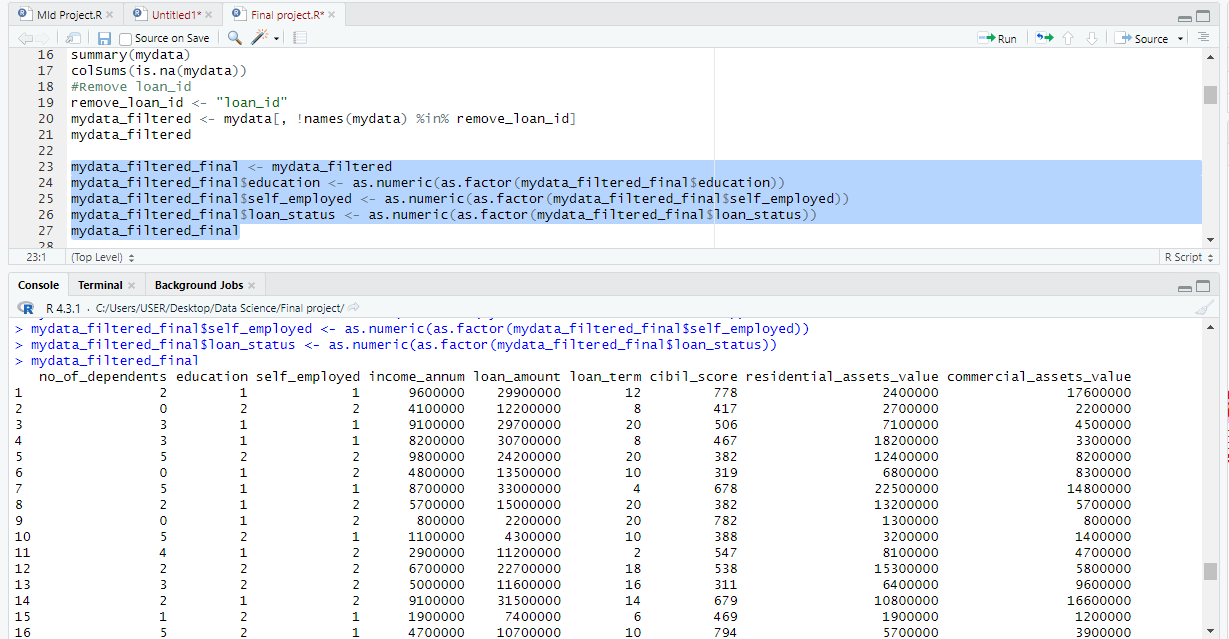
mydata\_filtered\_final <- mydata\_filtered

mydata\_filtered\_final$education <- as.numeric(as.factor(mydata\_filtered\_final$education))

mydata\_filtered\_final$self\_employed <- as.numeric(as.factor(mydata\_filtered\_final$self\_employed))

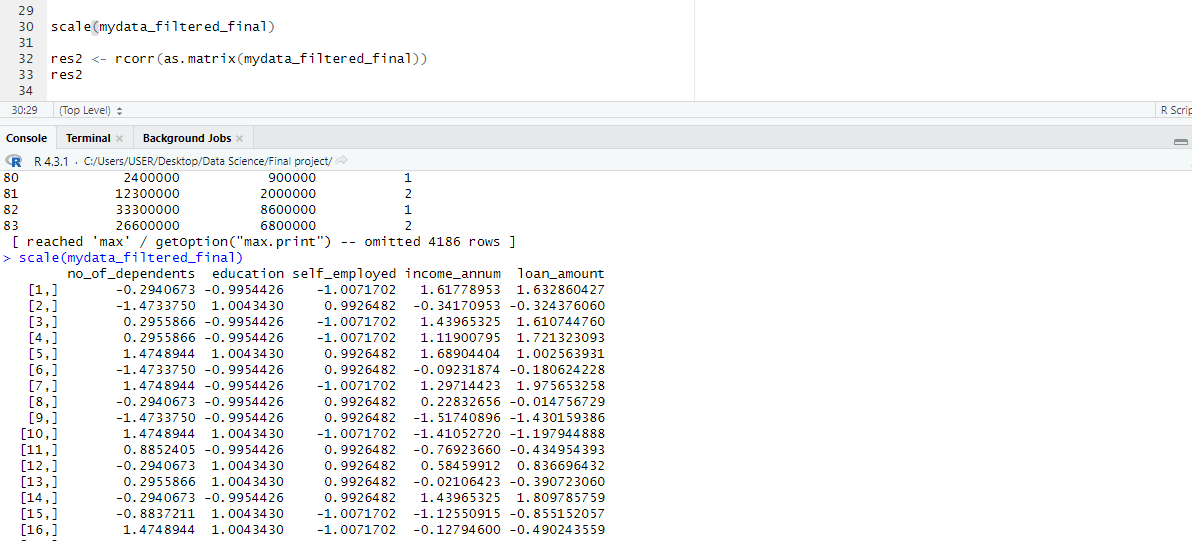
mydata\_filtered\_final$loan\_status <- as.numeric(as.factor(mydata\_filtered\_final$loan\_status))

mydata\_filtered\_final



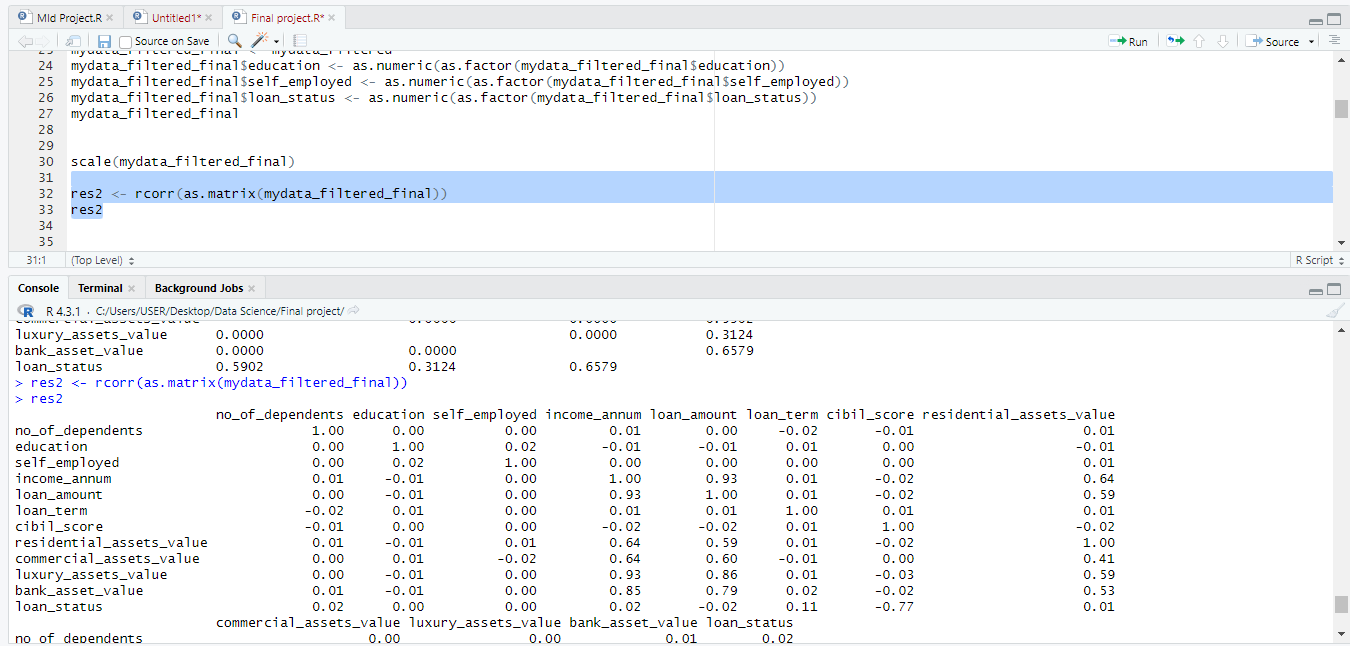
**Scalling:**

scale(mydata\_filtered\_final)



**Correlation Matrix:**

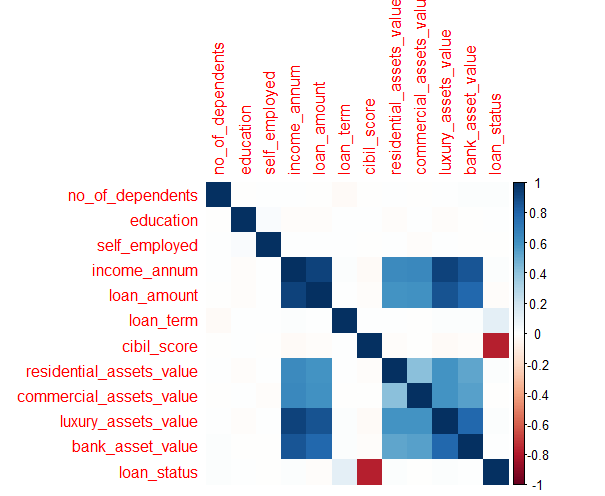
res2 <- rcorr(as.matrix(mydata\_filtered\_final))

res2

**Correlation Matrix with figure:**

correlation\_matrix <- cor(mydata\_filtered\_final[, sapply(mydata\_filtered\_final, is.numeric)])

corrplot(correlation\_matrix, method = "color")



**Splitting the data into 80/20 train/test ratio:**

set.seed(255)

split <- sample.split(mydata\_filtered\_final$loan\_status, SplitRatio = 0.8)

train <- subset(mydata\_filtered\_final, split == TRUE)

test <- subset(mydata\_filtered\_final, split == FALSE)

train\_scaled <- scale(train[, -13])

test\_scaled <- scale(test[, -13])

**Applying the KNN with the value of K as 10:**

k\_value <- 10

fold\_pred <- knn(

train = train\_scaled,

test = test\_scaled,

cl = train$loan\_status,

k = k\_value

)

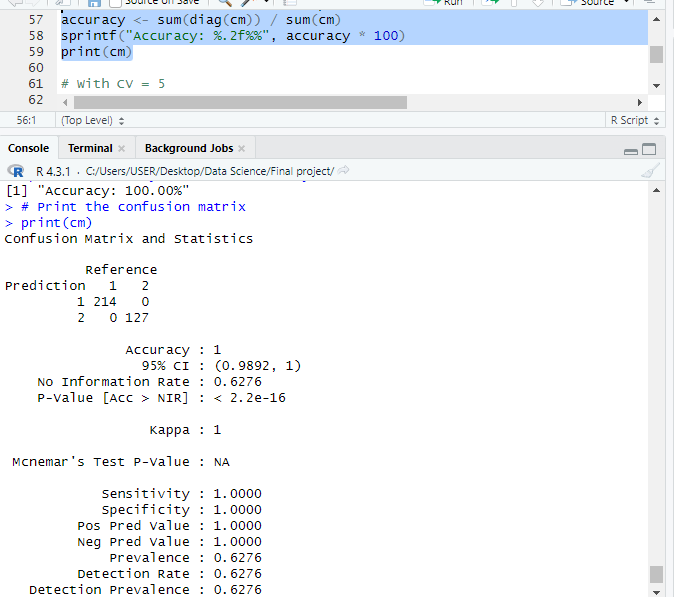
**Printing the accuracy achieved in KNN and also the required confusion matrix:**

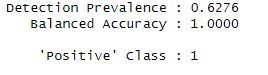
cm <- confusionMatrix(data=fold\_pred, reference=as.factor(fold\_val$loan\_status))

accuracy <- sum(diag(cm)) / sum(cm)

sprintf("Accuracy: %.2f%%", accuracy \* 100)

print(cm)





**Applying KNN with cross validation of 10 and finding the accuracy and confusion matrix:**

num\_folds <- 10

fold\_accuracies <- numeric(length = num\_folds

for (fold in 1:num\_folds) {

fold\_indices <- sample(1:nrow(train), size = floor(1 / num\_folds \* nrow(train)))

fold\_train <- train[-fold\_indices, ]

fold\_val <- train[fold\_indices, ]

fold\_train\_scaled <- scale(fold\_train[, -13])

fold\_val\_scaled <- scale(fold\_val[, -13])

fold\_pred <- knn(

train = fold\_train\_scaled,

test = fold\_val\_scaled,

cl = fold\_train$loan\_status,

k = k\_value

)

cm\_with\_cv <- confusionMatrix(data=fold\_pred, reference=as.factor(fold\_val$loan\_status)

print(cm\_with\_cv)

fold\_accuracy <- sum(diag(cm)) / sum(cm)

fold\_accuracies[fold] <- fold\_accuracy

}

average\_accuracy <- mean(fold\_accuracies)

sprintf("Average Accuracy across %d folds: %.2f%%", num\_folds, average\_accuracy \* 100)

cm\_with\_cv

