#1. Perform the below given activities:

# a. Take a sample data set of your choice

#b. Apply random forest, logistic regression using Spark R

#c. Predict for new dataset

lib=c("bigmemory", "readr", "Hmisc", "dplyr", "MASS", "ggplot2", "lattice", "caret", "rpart",

"randomForest", "rpart.plot","lattice", "rattle", "data.table","RColorBrewer", "reshape2",

"InformationValue","stringr", "VIF", "Information", "Amelia", "gdata", "party","car",

"lubridate","zoo", "sqldf", "fuzzyjoin", "party", "mice", "tseries", "timeSeries","forecast")

sapply(lib, require, character.only=TRUE, quietly=TRUE)

# Integration with spark with R

install.packages("sparklyr")

library(sparklyr)

spark\_install(version = "2.3.1")

install.packages("Rtools")

install.packages("devtools")

devtools::install\_github("rstudio/sparklyr")

install.packages("Rtools")

setwd("C:/Users/Jagannath/Documents/assignment data acadgild/assignmnet 21-24")

loanTrain <- read.csv("C:/Users/Jagannath/Documents/assignment data acadgild/assignmnet 21-24/loan.csv", na.strings = c(""," ","NA"))

loanTrain<-loan{}

## Check for duplicates

loanTrain<-loanTrain[!duplicated(loanTrain),]

############################## Missing Values #########################

## Visualize Na terms

install.packages("Amelia")

library(Amelia)

missmap(loanTrain)

sapply(loanTrain,function(x) sum(is.na(x)))

#### Impute mean/median/mode

library(ggplot2)

ggplot(loanTrain, aes(1, LoanAmount)) + geom\_boxplot()

hist(loanTrain$LoanAmount)

# Impute by Median

loanTrain$LoanAmount[is.na(loanTrain$LoanAmount)]<-median(loanTrain$LoanAmount, na.rm = T)

loanTrain$Self\_Employed[is.na(loanTrain$Self\_Employed)]<-mode(loanTrain$Self\_Employed, na.rm = T)

# Mode function

mode <- function(x){t <- as.data.frame(table(loanTrain$Credit\_History))

return(as.character(t$Var1[which.max(t$Freq)]))}

loanTrain$Credit\_History[is.na(loanTrain$Credit\_History)]<-mode(loanTrain$Credit\_History)

## Impute using package imputeMissings

library(imputeMissings)

l<-impute(loanTrain, method = "median/mode")

## Mice Package

library(mice)

d<-loanTrain[,c(2:12)]

imputed\_Data <- mice(d, m=5, maxit = 50, method = 'pmm', seed = 500)

#######outlier treatment#########

library(ggplot2)

## Capping

boxplot(l$LoanAmount)

qnt <- quantile(l$LoanAmount, 0.75, na.rm = T)

caps <- quantile(l$LoanAmount, 0.95, na.rm = T)

H <- 1.5 \* IQR(l$LoanAmount, na.rm = T)

l$LoanAmount[l$LoanAmount > (qnt + H)] <- caps

## CoapplicantIncome

boxplot(l$CoapplicantIncome)

ggplot(l, aes(1,CoapplicantIncome)) + geom\_boxplot(outlier.colour = "red", outlier.shape = 2)

qnt <- quantile(l$CoapplicantIncome, 0.75, na.rm = T)

caps <- quantile(l$CoapplicantIncome, 0.95, na.rm = T)

H <- 1.5 \* IQR(l$CoapplicantIncome, na.rm = T)

l$CoapplicantIncome[l$CoapplicantIncome > (qnt + H)] <- caps

### Applicant Income

ggplot(l, aes(1,ApplicantIncome)) + geom\_boxplot(outlier.colour = "red",

outlier.shape = 2)

qnt <- quantile(l$ApplicantIncome, 0.75, na.rm = T)

caps <- quantile(l$ApplicantIncome, 0.95, na.rm = T)

H <- 1.5 \* IQR(l$ApplicantIncome, na.rm = T)

l$ApplicantIncome[l$ApplicantIncome > (qnt + H)] <- caps

#### Bivariate Analysis

## Continuous Variable

contVars<-c("ApplicantIncome","CoapplicantIncome","LoanAmount",

"Loan\_Amount\_Term")

cont\_df<-l[,names(l) %in% contVars]

## Scatter plot

pairs(cont\_df)

library(corrplot)

corrplot(cor(cont\_df), type = "full", "ellipse")

#

ggplot(l, aes(Property\_Area, ApplicantIncome)) + geom\_boxplot(fill = "steelblue")

ggplot(l, aes(Gender, ApplicantIncome)) + geom\_boxplot(fill = "steelblue")

ggplot(l, aes(Dependents, ApplicantIncome)) + geom\_boxplot(fill = "steelblue")

### Data Modelling

# hot encoding categorical variables

l$Gender<- ifelse(l$Gender == "Female",0,1)

l$ Married<- ifelse(l$Married == "No",0,1)

l$Education <- ifelse(l$Education == "Not Graduate",0,1)

l$Self\_Employed <- ifelse(l$Self\_Employed == "No",0,1)

l$Loan\_Status <- ifelse(l$Loan\_Status == "N",0,1)

l$Loan\_Amount\_Term<-as.numeric(l$Loan\_Amount\_Term)

#many column are numeric that should be factor so coonverting them to factor

col\_list <- c("Gender","Married","Dependents","Education","Self\_Employed","Credit\_History","Loan\_Status")

l[col\_list] <- lapply(l[col\_list], factor)

# stripping + sign from dependents 3+ categorydf$Dependents

l$Dependents<- substr(l$Dependents, 1, 1)

l$Dependents <- as.factor(l$Dependents)# converting to a factor

# creating train and test data

library(caret)

index <- createDataPartition(l$Loan\_Status,p = .75,list = F) # creating partion based on Loanststaus

train <- l[index,] # creatingtrain data

test <- l[-index,]# creating test data

# removing Loan ID from train and test data

train <- subset(train,select= -c(Loan\_ID))

test <- subset(test,select = -c(Loan\_ID))

## Logistic Regression

str(train)

logistic<-glm(Loan\_Status~., family = "binomial", data = train)

summary(logistic)

# prediction

prediction <- predict(logistic,newdata=test,type='response')

prediction <- ifelse(prediction > 0.5,1,0)

prediction

# accuarcy check

train$Loan\_Status <- as.factor(train$Loan\_Status)

# gradient boosting

control <- trainControl(method = 'repeatedcv',

number = 5,

repeats = 3,

search = 'grid')

help("RandomForest-class")

# randam forest

library(pmml)

library(rpart)

loanDtree<-rpart(Loan\_Status~.,data = train)

loanDtree

pmml(loanDtree)

saveXML(pmml(loanDtree), file="loanDtree\_in\_pmml.xml")

library(sparklyr)

Ldtree<-spark\_connect(master="local")

loanDtree<-copy\_to(Ldtree, loanDtree)