Summary Report: Microfinance Expansion Project

Vivek Singh, Assistant Professor, University of Missouri–St. Louis

Rushikesh Maheshwari, Graduate Student, University of South Florida

# Executive Summary:

This report aims to address the effective and profitable expansion of a Microfinance company in a given region by driving insights from data collected for the current operational locations pertaining to their profitability, demographics, and other factors such as social, economic and financial indicators. This report covers the literature review, data collection and cleansing process and predictive modelling on collected data and explains what all factors plays an important role in defining successful operation of a microfinance company.

# Literature Review:

A literature review surveys books, scholarly articles, and any other sources relevant to a particular issue, area of research, or theory, and by so doing, provides a description, summary, and critical evaluation of these works in relation to the research problem being investigated.

Microfinance industries are turning out to , as name suggest comprises of loans, insurance and saving offered to customers who are underprivileged and

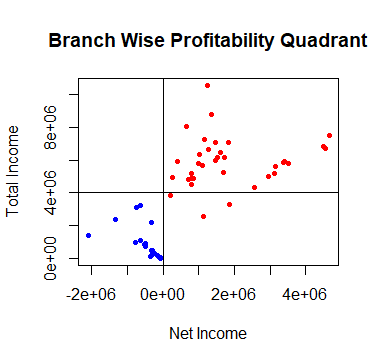
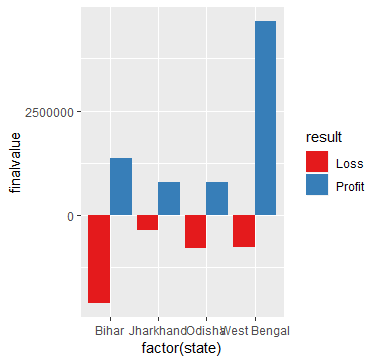
# **Data:**

## Data Exploration:

Branch name Current Status Predicted by model

ABC Profit Loss

BCA Profit Profit

p

## Data Point Selection:

The data for current operational branches of Janakalyan and their profitability was provided at start. After learning about the microfinance industry and that MFI provides loans and insurances to the people who lack traditional banking experiences or cannot benefit from the nationalized banks, I drilled into some demographic data in order to see if the demographic conditions plays an role in MFI business. Below were the data points collected from different sites to provide some **Demographic Information** about current branch locations.

|  |  |  |
| --- | --- | --- |
| **Data point** | **Description** | **Rational** |
| Population | Population of district | More population can drive more business opportunity |
| Increase in population | %increase in population of that location compared to previous year | Increase in population can help us with further possible expansion |
| Population density | population / km area | Gives us richness of population in particular area |
| Sex ratio | female to male ratio | Explains the diversity in population and possible degree of formation of self help groups by women |
| Literacy | % literacy in the region | The extent of literate population and hence opportunity for business |

We further wanted to collect the data explaining the economic and financial condition of that region.

So, we collected below data points.

|  |  |  |
| --- | --- | --- |
| **Data point** | **Description** | **Rational** |
| Unemployment Rate | % unemployment rate for a sample population at state level | Unemployment rate could state if there can be possibility that microfinance companies can help unemployed people help support their ventures if any |
| Growth by MFI | Count of SHGs in a region and amount saved in lakh Rs. | This data can provide us with existing count of SHGs and possible their possible profitability |
| GSDP (Gross State Domestic Product) | State-GDP | GSDP is an indicator of how the economy is performing in that state and could be a crucial factor for business expansion |

## **Data Collection:**

After finalizing that above data points are to be considered for analysis, I started looking for credible sources over the internet which can provide us with required data. Below are the online resources I referred to in order to gather the data.

<https://www.census2011.co.in/>

<https://unemploymentinindia.cmie.com/>

<https://data.gov.in/resources/region-wisestate-wise-progress-under-microfinance-savings-self-help-groups-shgs-banks>

<https://data.gov.in/major-indicator/state-wise-net-domestic-product-ndp-current-price>

<https://statisticstimes.com/economy/gdp-capita-of-indian-states.php>

I used RStudio to scrape the data from above websites in order to get the desired data. I used ‘rvest’ and ‘xml2’ packages for this script. The script is attached in [annexure (1.1)](#_1.1_Code_for)

## **Data Cleaning and Merging:**

After Web Scraping the data from above mentioned sources, I merged all these data points with the branch profitability list provided by Janakalyan. I used branch name, district names and state as common indicators amongst different data frames build from the scrapped data.

Below were the few observations and changes that were made during this process.

1. There were 3 branches for which profitability data was not available.

|  |  |  |  |
| --- | --- | --- | --- |
| 056 | **TIHIDI** | **BHADRAK** | **Odisha** |
| 057 | **BHADRAK** | **BHADRAK** | **Odisha** |
| 058 | **KUMARDHUBI** | **DHANBAD** | **Jharkhand** |

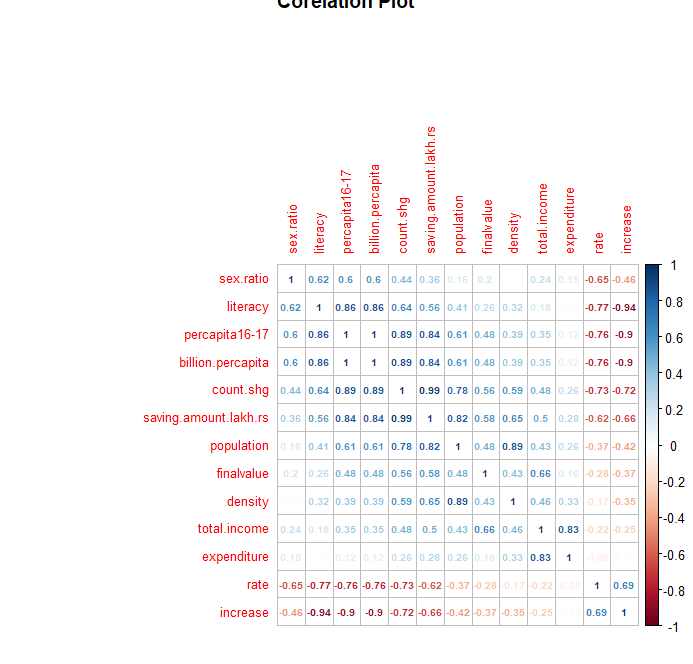
For these branches, all other data points were imputed as a median of rest of the line items present in the data set. All the unnecessary or duplicate columns and rows were deleted and

2. Based on the total investment and total expenditure values provided, I created a flag for each branch as “Profit” or “Loss”

The code for data cleaning and data merging is mentioned in [annexure (1.2)](#_1.2_Code_for)

## **Data Modelling:**

Plotted a correlation matrix to identify the correlation between Independent variables to be used to predict the profitability of the branches.



Based on above matrix, I decided to choose combination of **sex.ratio,literacy,count.shg and density** for our prediction model. The code for model and its output is posted below.

Based on above data, I ran a logistic regression with 10-folds cross validation method using **WEKA**. The results for the same were as below:

=== Run information ===

Instances: 58

Attributes: 6

sex.ratio

literacy

density

rate

count.shg

result

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8

Coefficients...

Class

Variable Profit

====================

sex.ratio 0.2112

literacy -0.0643

density 0.0403

rate -3.1702

count.shg -0

Intercept -191.4782

Odds Ratios...

Class

Variable Profit

====================

sex.ratio 1.2352

literacy 0.9377

density 1.0411

rate 0.042

count.shg 1

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 45 77.5862 %

Incorrectly Classified Instances 13 22.4138 %

Kappa statistic 0.5197

Mean absolute error 0.2213

Root mean squared error 0.3788

Relative absolute error 47.7034 %

Root relative squared error 78.7038 %

Total Number of Instances 58

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.811 0.286 0.833 0.811 0.822 0.520 0.845 0.898 Profit

0.714 0.189 0.682 0.714 0.698 0.520 0.845 0.738 Loss

Weighted Avg. 0.776 0.251 0.778 0.776 0.777 0.520 0.845 0.840

=== Confusion Matrix ===

a b <-- classified as

30 7 | a = Profit

6 15 | b = Loss

#Building the models  
  
m1 = glm(result~ density+sex.ratio+literacy,family=binomial(link = "logit"), data = finaldata)

m2 = glm(result~ density+sex.ratio+literacy+rate,family=binomial(link = "logit"), data = finaldata)

m3 = glm(result~ (density\*rate)+sex.ratio+literacy,family=binomial(link = "logit"), data = finaldata)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

m4 = glm(result~ density+sex.ratio+literacy+count.shg,family=binomial(link = "logit"), data = finaldata)

m5 = glm(result~ density+sex.ratio+literacy+rate+count.shg,family=binomial(link = "logit"), data = finaldata)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

library(stargazer)  
stargazer(m1,m2,m3,m4,m5,type = "text")  
## =======================================================================  
## Dependent variable:   
## -----------------------------------------------------  
## result   
## (1) (2) (3) (4) (5)   
## -----------------------------------------------------------------------  
## density 0.008\*\*\* 0.008\*\*\* -0.025 0.013\*\* 0.040   
## (0.002) (0.002) (0.029) (0.005) (0.043)   
##   
## sex.ratio 0.078\*\* 0.097\*\* 0.242 0.126\*\* 0.211   
## (0.033) (0.042) (0.165) (0.051) (0.175)   
##   
## literacy -0.006 0.023 -0.389 0.072 -0.064   
## (0.046) (0.058) (0.472) (0.105) (0.247)   
##   
## density:rate 0.006   
## (0.006)   
##   
## rate 0.282 -6.571 -3.170   
## (0.328) (6.486) (4.496)   
##   
## count.shg -0.00001 -0.00004   
## (0.00001) (0.0001)   
##   
## Constant -80.619\*\*\* -102.960\*\* -177.303\* -133.487\*\*\* -191.484   
## (30.417) (42.995) (100.903) (50.856) (139.354)  
##   
## -----------------------------------------------------------------------  
## Observations 58 58 58 58 58   
## Log Likelihood -18.384 -18.027 -15.955 -16.843 -16.162   
## Akaike Inf. Crit. 44.769 46.055 43.910 43.687 44.323   
## =======================================================================  
## Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

Not considering model m3 and m5 as it is providing us with fitted values of probabilities as either 1 or 0 which is too idealistic.

Further I built the confusion matrix and tested the model for results on the ‘test locations’ provided by Janakalyan.

#m1 Predict  
  
m1.predictedvalues <- as.factor(if\_else(fitted.values(m1) >0.5,"Profit","Loss"))  
confusionMatrix(m1.predictedvalues,finaldata$result)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Loss Profit  
## Loss 19 7  
## Profit 2 30  
##   
## Accuracy : 0.8448   
## 95% CI : (0.7258, 0.9265)  
## No Information Rate : 0.6379   
## P-Value [Acc > NIR] : 0.0004424   
##   
## Kappa : 0.6805   
## Mcnemar's Test P-Value : 0.1824224   
##   
## Sensitivity : 0.9048   
## Specificity : 0.8108   
## Pos Pred Value : 0.7308   
## Neg Pred Value : 0.9375   
## Prevalence : 0.3621   
## Detection Rate : 0.3276   
## Detection Prevalence : 0.4483   
## Balanced Accuracy : 0.8578   
##   
## 'Positive' Class : Loss   
##

m1.predict <- predict(m1,test\_data,type = "response");m1.predict

## 1 2 3 4 5   
## 0.02712272 0.02840296 0.12903717 0.04962254 0.27246825

m1.predict <- ifelse(m1.predict >0.5,"Profit","Loss");m1.predict

## 1 2 3 4 5   
## "Loss" "Loss" "Loss" "Loss" "Loss"

#m2 Predict  
  
m2.predictedvalues <- as.factor(if\_else(fitted.values(m2) >0.5,"Profit","Loss"))  
confusionMatrix(m2.predictedvalues,finaldata$result)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Loss Profit  
## Loss 19 6  
## Profit 2 31  
##   
## Accuracy : 0.8621   
## 95% CI : (0.7462, 0.9385)  
## No Information Rate : 0.6379   
## P-Value [Acc > NIR] : 0.0001331   
##   
## Kappa : 0.7132   
## Mcnemar's Test P-Value : 0.2888444   
##   
## Sensitivity : 0.9048   
## Specificity : 0.8378   
## Pos Pred Value : 0.7600   
## Neg Pred Value : 0.9394   
## Prevalence : 0.3621   
## Detection Rate : 0.3276   
## Detection Prevalence : 0.4310   
## Balanced Accuracy : 0.8713   
##   
## 'Positive' Class : Loss   
##

m2.predict <- predict(m2,test\_data,type = "response");m2.predict

## 1 2 3 4 5   
## 0.01240138 0.01229439 0.07430604 0.03492351 0.22473754

m2.predict <- ifelse(m2.predict >0.5,"Profit","Loss");m2.predict

## 1 2 3 4 5   
## "Loss" "Loss" "Loss" "Loss" "Loss"

#m4 Predict  
  
m4.predictedvalues <- as.factor(if\_else(fitted.values(m4) >0.5,"Profit","Loss"))  
confusionMatrix(m4.predictedvalues,finaldata$result)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Loss Profit  
## Loss 15 1  
## Profit 6 36  
##   
## Accuracy : 0.8793   
## 95% CI : (0.767, 0.9501)  
## No Information Rate : 0.6379   
## P-Value [Acc > NIR] : 3.506e-05   
##   
## Kappa : 0.7246   
## Mcnemar's Test P-Value : 0.1306   
##   
## Sensitivity : 0.7143   
## Specificity : 0.9730   
## Pos Pred Value : 0.9375   
## Neg Pred Value : 0.8571   
## Prevalence : 0.3621   
## Detection Rate : 0.2586   
## Detection Prevalence : 0.2759   
## Balanced Accuracy : 0.8436   
##   
## 'Positive' Class : Loss   
##

m4.predict <- predict(m4,test\_data,type = "response");m4.predict

## 1 2 3 4 5   
## 0.014001881 0.009260216 0.081254848 0.034000209 0.502052383

m4.predict <- ifelse(m4.predict >0.5,"Profit","Loss");m4.predict

## 1 2 3 4 5   
## "Loss" "Loss" "Loss" "Loss" "Profit"

For all the 3 models, all the test branches were predicted to be in “Loss”” except for the model m5 where branch “KARIMGANJ” which was predicted as a “Profit” by the model.

Annexure

# 1.1 Code for web-scraping and data collection

DataCollection\_WebScrapping.R

Rushikesh

Thu Aug 01 18:56:07 2019

library(rvest)

## Loading required package: xml2

library(xml2)  
library(openxlsx)  
library(rio)  
library(tidyr)  
  
  
setwd("F:/Vivek")  
  
###############################################################  
# Collecting Data for population, density, sex ratio, litracy #  
###############################################################  
  
links = c("https://www.census2011.co.in/census/state/districtlist/west+bengal.html",  
 "https://www.census2011.co.in/census/state/districtlist/bihar.html",  
 "https://www.census2011.co.in/census/state/districtlist/jharkhand.html",  
 "https://www.census2011.co.in/census/state/districtlist/orissa.html",  
 "https://www.census2011.co.in/census/state/districtlist/assam.html")  
  
for (i in 1:5)  
{  
 url <- links[i]  
 webpage <- read\_html(url)  
   
 #converting html to dataframe  
 state\_html <- html\_nodes(webpage,xpath='/html/body/div[1]/div/div[1]/table')  
 state\_data <-html\_table(state\_html, fill = TRUE)  
 typeof(state\_data)  
 state\_data <- as.data.frame(state\_data);state\_data  
   
 #Cleaning the dataframe  
 statename = c("westBengal","bihar","jharkhand","orissa","Assam")  
 state\_data <- state\_data[,-c(1,3,9:99)]  
 colnames(state\_data) <- tolower(make.names(colnames(state\_data)))   
 state\_data <- state\_data[-c(8,16),]  
 state\_data$state <- statename[i]  
 assign(paste("statedata\_",statename[i], sep = ""), state\_data)  
 rm(state\_data)  
}   
  
collateddata <- rbind(statedata\_westBengal,statedata\_jharkhand,statedata\_orissa,statedata\_bihar,statedata\_Assam)  
#write.xlsx(masterdata, "sampel\_scarped\_data.xlsx")  
  
#############################################  
  
#State wise Unemployment rate  
  
url <- "https://unemploymentinindia.cmie.com/"  
webpage <- read\_html(url)  
  
# Converting html to datafram  
tbls <- html\_nodes(webpage, "table")  
unemp\_html <- html\_nodes(tbls, xpath = '/html/body/table/tbody/tr/td/table/tbody/tr[2]/td/table/tbody/tr/td[2]/table[2]')  
unempdata <- html\_table(tbls,fill = TRUE)  
#unempdata <- as.data.frame(unempdata)  
  
unempdata <- unempdata[[16]]  
  
######################################  
  
#Growth under MFI and SHGs (2015)  
  
mfgrowth <- read.csv("rs\_session\_239\_AU1837\_1.1\_1.csv")  
#mfgrowth <- subset(mfgrowth, mfgrowth$REGION == "EASTERN REGION" )  
mfgrowth <- mfgrowth[-c(1,6),-c(1,2)]  
  
######################################  
  
#GSDP per capita Data  
  
url <- "https://statisticstimes.com/economy/gdp-capita-of-indian-states.php"  
webpage <- read\_html(url)  
tbls <- html\_nodes(webpage, "table")  
  
gsdp\_html <- html\_nodes(tbls, xpath ='//\*[@id="table\_id"]')  
gsdp\_data <- html\_table(gsdp\_html,fill = TRUE)  
gsdp\_data <- as.data.frame(gsdp\_data)  
  
#Cleaning  
  
gsdp\_data <- gsdp\_data[,-c(1,6,8:10)]  
  
#####################################  
  
#Writing the excel file  
  
# Create a blank workbook  
masterdata <- createWorkbook()  
  
# Add some sheets to the workbook  
addWorksheet(masterdata, "Demographic")  
addWorksheet(masterdata, "UnempData")  
addWorksheet(masterdata, "MFIGrowth")  
addWorksheet(masterdata, "GSDPPCData")  
  
# Write the data to the sheets  
writeData(masterdata, sheet = "Demographic", x = collateddata)  
writeData(masterdata, sheet = "UnempData", x = unempdata)  
writeData(masterdata, sheet = "MFIGrowth", x = mfgrowth)  
writeData(masterdata, sheet = "GSDPPCData", x = gsdp\_data)  
  
# Reorder worksheets  
worksheetOrder(masterdata) <- c(1,2,3,4)  
  
# Export the file  
#saveWorkbook(masterdata, "Masterdata.xlsx")

# 1.2 Code for Data merging and cleaning

**DataCleaning\_and\_Merging.R**

Rushikesh

Thu Aug 01 18:35:21 2019

###############################  
#Data Cleaning and Preparation#  
###############################  
  
library(openxlsx)  
library(rio)  
library(tidyr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

#Reading the Profit Loss data  
branch\_pl <- import("F:/Vivek/Branch Pofitability wo allocationsFY18-19 (003).xls", skip=2)  
branch\_pl <- branch\_pl[-c(56:58),]  
branch\_pl <- branch\_pl %>% separate("Branch",c("BranchCode","branch"), remove = TRUE) #To split the data on hyphen, space or other character  
str(branch\_pl)

## 'data.frame': 55 obs. of 17 variables:  
## $ BranchCode : chr "001" "002" "003" "004" ...  
## $ branch : chr "DUTTAPULIA" "HANSKHALI" "MATIKUMRA" "LAUHATI" ...  
## $ Disbursement : num 48385000 50067800 46877800 54197000 61663000 ...  
## $ Outstanding : num 30689334 31743931 32385961 28679093 34690797 ...  
## $ Interest Collection: num 5323078 5472744 6722031 6404707 6663294 ...  
## $ Reverse : num 256071 234283 328214 26664 34410 ...  
## $ Booked : num 213939 298481 26215 132217 144133 ...  
## $ Utkarsh BC : num 0 0 2357464 0 0 ...  
## $ Mas/Assignment : num 1187421 1228225 0 1109642 1342245 ...  
## $ Own Income : num 4094990 4310182 4064033 5402084 5432237 ...  
## $ P.Fees : num 455040 470677 0 425233 514371 ...  
## $ Utkarsh Commission : num 0 0 1098541 0 0 ...  
## $ MAS Commission : num 357937 370237 0 334491 404607 ...  
## $ Other Income : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Total Income : num 4907966 5151095 5162574 6161808 6351215 ...  
## $ Expenditure : num 4649638 4370939 2039140 4438315 5342270 ...  
## $ Profit/(Loss) : num 258328 780157 3123434 1723493 1008945 ...

branch\_pl <- branch\_pl[,-c(3:14)]  
names(branch\_pl)[5] <- "finalvalue"  
colnames(branch\_pl) <- tolower(make.names(colnames(branch\_pl)))  
branch\_pl$result <- ifelse(branch\_pl$finalvalue > 0,"Profit","Loss")  
  
#Reading demographic data  
demographic <- read.xlsx("Masterdata.xlsx",sheet = "Demographic")  
str(demographic)

## 'data.frame': 138 obs. of 7 variables:  
## $ district : chr "North Twenty Four Parganas" "South Twenty Four Parganas" "Barddhaman" "Murshidabad" ...  
## $ population: chr "10,009,781" "8,161,961" "7,717,563" "7,103,807" ...  
## $ increase : chr "12.04 %" "18.17 %" "11.92 %" "21.09 %" ...  
## $ sex.ratio : chr "955" "956" "945" "958" ...  
## $ literacy : chr "84.06 %" "77.51 %" "76.21 %" "66.59 %" ...  
## $ density : chr "2445" "819" "1099" "1334" ...  
## $ state : chr "westBengal" "westBengal" "westBengal" "westBengal" ...

demographic$increase <- as.numeric(gsub("\\%","",demographic$increase)) #removing % sign  
demographic$literacy <- as.numeric(gsub("\\%","",demographic$literacy)) #removing % sign  
demographic$population <- as.numeric(gsub("\\,","",demographic$population)) #removing comma in population data  
demographic <- transform(demographic, sex.ratio = as.numeric(sex.ratio), density = as.numeric(density))  
demographic[demographic=="westBengal"]<- "West Bengal"  
demographic[demographic=="bihar"]<- "Bihar"  
demographic[demographic=="Odissa"]<- "Odisha"  
demographic[demographic=="jharkhand"]<- "Jharkhand"  
  
#Reading Unemployment Data  
unemp <-read.xlsx("Masterdata.xlsx", sheet = "UnempData")  
str(unemp)

## 'data.frame': 28 obs. of 2 variables:  
## $ Unemployment.Rate.(%): chr "States (India)" "Andhra Pradesh" "Assam" "Bihar" ...  
## $ Unemployment.Rate.(%): chr "Jun 2019" "3.6" "5.5" "10.1" ...

names(unemp)[1]<- "state"  
names(unemp)[2]<- "rate"  
unemp <- unemp[-c(1,29),]  
unemp$rate <- as.numeric(unemp$rate)

## Warning: NAs introduced by coercion

#Reading MFI Growth Data  
mfigrowth <-read.xlsx("Masterdata.xlsx", sheet = "MFIGrowth")  
str(mfigrowth)

## 'data.frame': 33 obs. of 3 variables:  
## $ Region.State : chr "A & N Islands (UT)" "Andhra Pradesh" "Arunachal Pradesh" "Assam" ...  
## $ Total.No..of.SHGs: num 4998 884508 3351 292071 224469 ...  
## $ Savings.Amount : num 125 262950 256 9943 29667 ...

names(mfigrowth)[1]<- "state"  
names(mfigrowth)[2]<- "count.shg"  
names(mfigrowth)[3]<- "saving.amount.lakh.rs"  
  
#Reading GSDP Data  
gsdp <- read.xlsx("Masterdata.xlsx",sheet = "GSDPPCData")  
str(gsdp)

## 'data.frame': 35 obs. of 5 variables:  
## $ State : chr "State" "Goa" "Delhi" "Sikkim" ...  
## $ GSDP.per.capita..INR.at.Current.prices. : chr "18-19" "-" "402,172" "-" ...  
## $ GSDP.per.capita..INR.at.Current.prices..1: chr "17-18" "-" "362,790" "340,705" ...  
## $ GSDP.per.capita..INR.at.Current.prices..2: chr "16-17" "413,334" "329,836" "309,428" ...  
## $ GDP.per.capita...billion. : chr "2016" "6,045" "4,824" "4,526" ...

gsdp <- gsdp[-c(1,36),-c(2:3)]  
names(gsdp)[1] <- "state"  
names(gsdp)[2] <- "percapita16-17"  
names(gsdp)[3] <- "billion.percapita"  
gsdp$`percapita16-17` <- as.numeric(gsub("\\,","",gsdp$`percapita16-17`))  
gsdp$billion.percapita <- as.numeric(gsub("\\,","",gsdp$billion.percapita))  
  
  
#Reading State wise branchList  
branch <- import("State and District wise branch list.xlsx")  
str(branch)

## 'data.frame': 59 obs. of 5 variables:  
## $ Sl No. : num 1 2 3 4 5 6 7 8 9 10 ...  
## $ Branch Code : chr "000" "001" "002" "003" ...  
## $ Branch Name : chr "HEAD OFFICE" "DUTTAPULIA" "HANSKHALI" "MATIKUMRA" ...  
## $ District Name: chr "NORTH 24 PARGANAS" "NADIA" "NADIA" "NADIA" ...  
## $ State : chr "West Bengal" "West Bengal" "West Bengal" "West Bengal" ...

colnames(branch)=tolower(make.names(colnames(branch)))  
names(branch)[4] <- "district"  
unique(demographic$district)

## [1] "North Twenty Four Parganas" "South Twenty Four Parganas"  
## [3] "Barddhaman" "Murshidabad"   
## [5] "Paschim Medinipur" "Hugli"   
## [7] "Nadia" "Purba Medinipur"   
## [9] "Haora" "Kolkata"   
## [11] "Maldah" "Jalpaiguri"   
## [13] "Bankura" "Birbhum"   
## [15] "Uttar Dinajpur" "Puruliya"   
## [17] "Koch Bihar" "Darjiling"   
## [19] "Dakshin Dinajpur" "Ranchi"   
## [21] "Dhanbad" "Giridih"   
## [23] "Purbi Singhbhum" "Bokaro"   
## [25] "Palamu" "Hazaribagh"   
## [27] "Pashchimi Singhbhum" "Deoghar"   
## [29] "Garhwa" "Dumka"   
## [31] "Godda" "Sahibganj"   
## [33] "Saraikela Kharsawan" "Chatra"   
## [35] "Gumla" "Ramgarh"   
## [37] "Pakur" "Jamtara"   
## [39] "Latehar" "Kodarma"   
## [41] "Simdega" "Khunti"   
## [43] "Lohardaga" "Ganjam"   
## [45] "Cuttack" "Mayurbhanj"   
## [47] "Baleshwar" "Khordha"   
## [49] "Sundargarh" "Jajapur"   
## [51] "Kendujhar" "Puri"   
## [53] "Balangir" "Kalahandi"   
## [55] "Bhadrak" "Bargarh"   
## [57] "Kendrapara" "Koraput"   
## [59] "Anugul" "Nabarangapur"   
## [61] "Dhenkanal" "Jagatsinghapur"   
## [63] "Sambalpur" "Rayagada"   
## [65] "Nayagarh" "Kandhamal"   
## [67] "Malkangiri" "Nuapada"   
## [69] "Subarnapur" "Jharsuguda"   
## [71] "Gajapati" "Baudh"   
## [73] "Debagarh" "Araria"   
## [75] "Arwal" "Aurangabad"   
## [77] "Banka" "Begusarai"   
## [79] "Bhagalpur" "Bhojpur"   
## [81] "Buxar" "Darbhanga"   
## [83] "Gaya" "Gopalganj"   
## [85] "Jamui" "Jehanabad"   
## [87] "Kaimur" "Katihar"   
## [89] "Khagaria" "Kishanganj"   
## [91] "Lakhisarai" "Madhepura"   
## [93] "Madhubani" "Munger"   
## [95] "Muzaffarpur" "Nalanda"   
## [97] "Nawada" "Pashchim Champaran"   
## [99] "Patna" "Purbi Champaran"   
## [101] "Purnia" "Rohtas"   
## [103] "Saharsa" "Samastipur"   
## [105] "Saran" "Sheikhpura"   
## [107] "Sheohar" "Sitamarhi"   
## [109] "Siwan" "Supaul"   
## [111] "Vaishali" "Nagaon"   
## [113] "Dhubri" "Sonitpur"   
## [115] "Cachar" "Barpeta"   
## [117] "Kamrup" "Tinsukia"   
## [119] "Dibrugarh" "Kamrup Metropolitan"   
## [121] "Karimganj" "Sivasagar"   
## [123] "Jorhat" "Golaghat"   
## [125] "Lakhimpur" "Goalpara"   
## [127] "Morigaon" "Karbi Anglong"   
## [129] "Baksa" "Darrang"   
## [131] "Kokrajhar" "Udalguri"   
## [133] "Nalbari" "Bongaigaon"   
## [135] "Dhemaji" "Hailakandi"   
## [137] "Chirang" "Dima Hasao"

unique(branch$district)

## [1] "NORTH 24 PARGANAS" "NADIA" "NORTH DINAJPUR"   
## [4] "MURSHIDABAD" "SOUTH DINAJPUR" "KISHANGANJ"   
## [7] "PURNIA" "PURI" "ARARIA"   
## [10] "JAMTARA" "CUTTACK" "SOUTH 24 PARGANAS"  
## [13] "KATIHAR" "BHADRAK" "DHANBAD"

demographic$district <- toupper(x = demographic$district)  
demographic$district <- ifelse(demographic$district == "NORTH TWENTY FOUR PARGANAS","NORTH 24 PARGANAS",demographic$district)  
demographic$district <- ifelse(demographic$district == "SOUTH TWENTY FOUR PARGANAS","SOUTH 24 PARGANAS",demographic$district)  
  
#Merging to data frames  
d1 <- left\_join(branch,demographic, by = "district")  
  
#Imputing missing values  
  
summary(d1)

## sl.no. branch.code branch.name district   
## Min. : 1.0 Length:59 Length:59 Length:59   
## 1st Qu.:15.5 Class :character Class :character Class :character   
## Median :30.0 Mode :character Mode :character Mode :character   
## Mean :30.0   
## 3rd Qu.:44.5   
## Max. :59.0   
##   
## state.x population increase sex.ratio   
## Length:59 Min. : 791042 Min. :11.99 Min. :909.0   
## Class :character 1st Qu.: 1698730 1st Qu.:12.22 1st Qu.:947.0   
## Mode :character Median : 5167600 Median :13.05 Median :950.0   
## Mean : 4602471 Mean :18.84 Mean :947.6   
## 3rd Qu.: 7103807 3rd Qu.:28.33 3rd Qu.:956.0   
## Max. :10009781 Max. :30.40 Max. :981.0   
## NA's :6 NA's :6 NA's :6   
## literacy density state.y   
## Min. :51.08 Min. : 437 Length:59   
## 1st Qu.:55.46 1st Qu.: 897 Class :character   
## Median :74.97 Median :1316 Mode :character   
## Mean :69.71 Mean :1203   
## 3rd Qu.:77.51 3rd Qu.:1316   
## Max. :85.50 Max. :2445   
## NA's :6 NA's :6

for(i in 1:ncol(d1)){  
 if (is.numeric(d1[,i]) == TRUE){  
 d1[is.na(d1[,i]),i] <- median(d1[,i], na.rm = TRUE)  
 }  
 }  
  
datafinal <- d1[-c(1),-c(1,11)]  
str(datafinal)

## 'data.frame': 58 obs. of 9 variables:  
## $ branch.code: chr "001" "002" "003" "004" ...  
## $ branch.name: chr "DUTTAPULIA" "HANSKHALI" "MATIKUMRA" "LAUHATI" ...  
## $ district : chr "NADIA" "NADIA" "NADIA" "NORTH 24 PARGANAS" ...  
## $ state.x : chr "West Bengal" "West Bengal" "West Bengal" "West Bengal" ...  
## $ population : num 5167600 5167600 5167600 10009781 5167600 ...  
## $ increase : num 12.2 12.2 12.2 12 13.1 ...  
## $ sex.ratio : num 947 947 947 955 950 947 947 958 958 955 ...  
## $ literacy : num 75 75 75 84.1 75 ...  
## $ density : num 1316 1316 1316 2445 1316 ...

names(datafinal)[2] <- "branch"   
names(datafinal)[4] <- "state"  
head(datafinal)

## branch.code branch district state population increase  
## 2 001 DUTTAPULIA NADIA West Bengal 5167600 12.22  
## 3 002 HANSKHALI NADIA West Bengal 5167600 12.22  
## 4 003 MATIKUMRA NADIA West Bengal 5167600 12.22  
## 5 004 LAUHATI NORTH 24 PARGANAS West Bengal 10009781 12.04  
## 6 005 KARNAJORA NORTH DINAJPUR West Bengal 5167600 13.05  
## 7 006 ARANGHATA NADIA West Bengal 5167600 12.22  
## sex.ratio literacy density  
## 2 947 74.97 1316  
## 3 947 74.97 1316  
## 4 947 74.97 1316  
## 5 955 84.06 2445  
## 6 950 74.97 1316  
## 7 947 74.97 1316

#Combining demographic data and Branch level profit data  
  
finaldata <- datafinal %>%  
 left\_join(branch\_pl, by = "branch")%>%  
 left\_join(unemp,by = "state")%>%  
 left\_join(mfigrowth,by = "state")%>%  
 left\_join(gsdp,by = "state")  
  
head(finaldata)

## branch.code branch district state population increase  
## 1 001 DUTTAPULIA NADIA West Bengal 5167600 12.22  
## 2 002 HANSKHALI NADIA West Bengal 5167600 12.22  
## 3 003 MATIKUMRA NADIA West Bengal 5167600 12.22  
## 4 004 LAUHATI NORTH 24 PARGANAS West Bengal 10009781 12.04  
## 5 005 KARNAJORA NORTH DINAJPUR West Bengal 5167600 13.05  
## 6 006 ARANGHATA NADIA West Bengal 5167600 12.22  
## sex.ratio literacy density branchcode total.income expenditure  
## 1 947 74.97 1316 001 4907966 4649638  
## 2 947 74.97 1316 002 5151095 4370939  
## 3 947 74.97 1316 003 5162574 2039140  
## 4 955 84.06 2445 004 6161808 4438315  
## 5 950 74.97 1316 005 6351215 5342270  
## 6 947 74.97 1316 006 5641977 4542549  
## finalvalue result rate count.shg saving.amount.lakh.rs percapita16-17  
## 1 258328.5 Profit 6.8 760941 127348 91114  
## 2 780156.6 Profit 6.8 760941 127348 91114  
## 3 3123434.1 Profit 6.8 760941 127348 91114  
## 4 1723492.9 Profit 6.8 760941 127348 91114  
## 5 1008944.6 Profit 6.8 760941 127348 91114  
## 6 1099427.5 Profit 6.8 760941 127348 91114  
## billion.percapita  
## 1 1333  
## 2 1333  
## 3 1333  
## 4 1333  
## 5 1333  
## 6 1333

finaldata <- finaldata[,-c(1,10)]  
str(finaldata)

## 'data.frame': 58 obs. of 17 variables:  
## $ branch : chr "DUTTAPULIA" "HANSKHALI" "MATIKUMRA" "LAUHATI" ...  
## $ district : chr "NADIA" "NADIA" "NADIA" "NORTH 24 PARGANAS" ...  
## $ state : chr "West Bengal" "West Bengal" "West Bengal" "West Bengal" ...  
## $ population : num 5167600 5167600 5167600 10009781 5167600 ...  
## $ increase : num 12.2 12.2 12.2 12 13.1 ...  
## $ sex.ratio : num 947 947 947 955 950 947 947 958 958 955 ...  
## $ literacy : num 75 75 75 84.1 75 ...  
## $ density : num 1316 1316 1316 2445 1316 ...  
## $ total.income : num 4907966 5151095 5162574 6161808 6351215 ...  
## $ expenditure : num 4649638 4370939 2039140 4438315 5342270 ...  
## $ finalvalue : num 258328 780157 3123434 1723493 1008945 ...  
## $ result : chr "Profit" "Profit" "Profit" "Profit" ...  
## $ rate : num 6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.8 ...  
## $ count.shg : num 760941 760941 760941 760941 760941 ...  
## $ saving.amount.lakh.rs: num 127348 127348 127348 127348 127348 ...  
## $ percapita16-17 : num 91114 91114 91114 91114 91114 ...  
## $ billion.percapita : num 1333 1333 1333 1333 1333 ...

#Original copy of final data  
original <- finaldata  
  
#Imputing missing data for branches not mentioned in previous list  
finaldata[sapply(finaldata, is.numeric)] <- lapply(finaldata[sapply(finaldata, is.numeric)], function(x) ifelse(is.na(x), median(x, na.rm = TRUE), x))

finaldata$result <- as.factor(ifelse(finaldata$finalvalue > 0,"Profit","Loss"))

finaldata <- finaldata%>% select(-result,result) #Moving Result column to the end !  
write.csv(finaldata, file = "Finaldataforanalysis1.csv")