

**DR. D. Y. PATIL VIDYAPEETH**

**PIMPRI, PUNE – 411 018**

**DR. D. Y. PATIL SCHOOL OF SCIENCE & TECHNOLOGY**

**TATHAWADE, PUNEs**

**A practical Report on**

**………………BANKING………………**

**Submitted By**

**Name: Sachin Vijay Rathod**

**Roll No: BTAI - 2145**

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**ACADEMIC YEAR 2022-2023**

**INTRODUCTION TO R PROGRAMMING**

R is an open-source programming language that is widely used as a statistical software and data analysis tool. R generally comes with the Command-line interface. R is available across widely used platforms like Windows, Linux, and macOS. Also, the R programming language is the latest cutting-edge tool.

It was designed by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team. R programming language is an implementation of the S programming language. It also combines with lexical scoping semantics inspired by Scheme. Moreover, the project conceives in 1992, with an initial version released in 1995 and a stable beta version in 2000.

**FEATURES USED IN THE PROJECT**

Each row represents characteristic of a single customer . Many categorical data has

been coded to mask the data, you don’t need to worry about their exact meaning

1 - age (numeric)

2 - job: type of job (categorical: “admin.”,“unknown”,“unemployed”,“management”,“housemaid”,“entrepreneur”,“student”, “blue-collar”, “self-employed”,“retired”,“technician”, “services”)

3 - marital: marital status (categorical: “married”,“divorced”,“single”; note: “divorced” means divorced or widowed)

4 - education (categorical: “unknown”,“secondary”,“primary”,“tertiary”)

5 - default: has credit in default? (binary: “yes”,“no”)

6 - balance: average yearly balance, in euros (numeric)

7 - housing: has housing loan? (binary: “yes”,“no”)

8 - loan: has personal loan? (binary: “yes”,“no”)

Related with the last contact of the current campaign:

9 - contact: contact communication type (categorical: “unknown”,“telephone”,“cellular”)

10 - day: last contact day of the month (numeric))

Direct Marketing Campaign: Details and Phase I Tasks

11 - month: last contact month of year (categorical: “jan”, “feb”, “mar”, . . . , “nov”, “dec”)

12 - duration: last contact duration, in seconds (numeric)

other attributes: 13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)

15 - previous: number of contacts performed before this campaign and for this client (numeric)

16 - poutcome: outcome of the previous marketing campaign (categorical: “unknown”,“other”,“failure”,“success”)

Output variable (desired target):

17 - y - has the client subscribed a term deposit? (binary: “yes”,“no”)

**PROBLEM STATEMENT**

A Portuguese bank is rolling out term deposit for its customers. They have in the past connected to their customer base through phone calls. Results for these previous campaigns were recorded and have been provided to the current campaign manager to use the same in making this campaign more effective.

 Challenges that the manager faces are following:

 Customers have recently started to complain that bank’s marketing staff bothers them

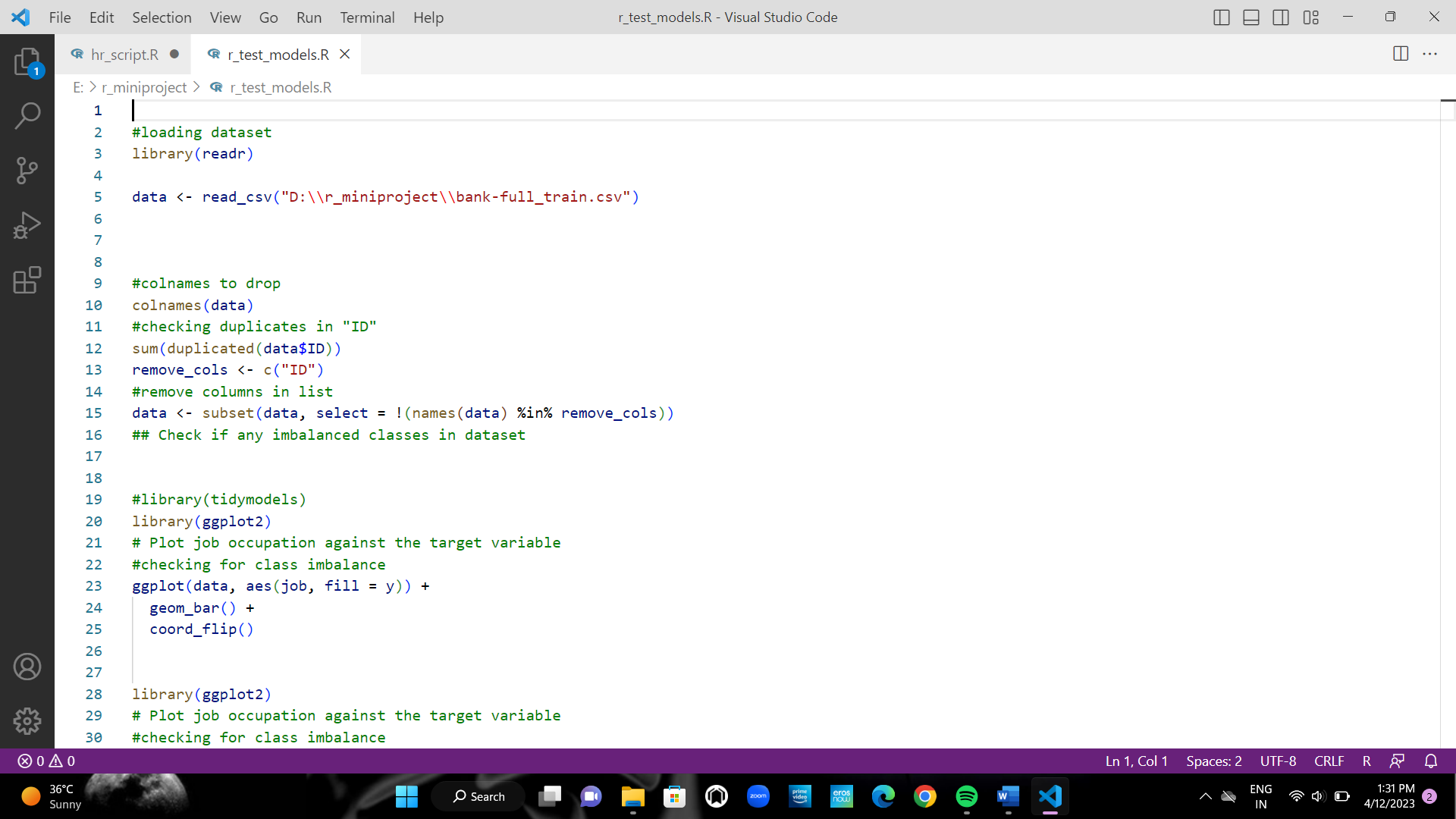
with irrelevant product calls and this should immediately stop

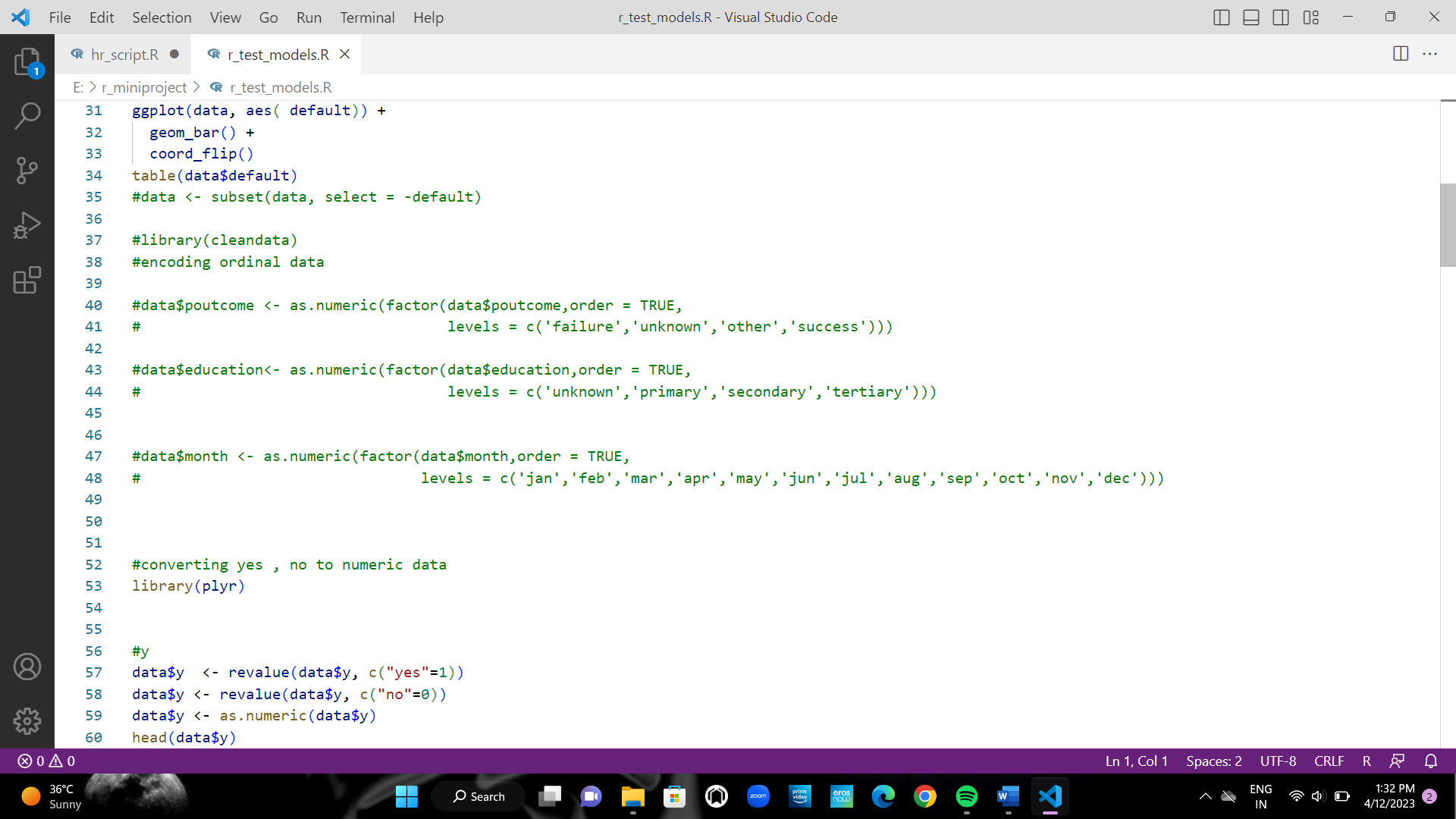
There is no prior framework for her decide and choose which customer to call and which one to leave alone. She has decided to use past data to automate this decision, instead of manually

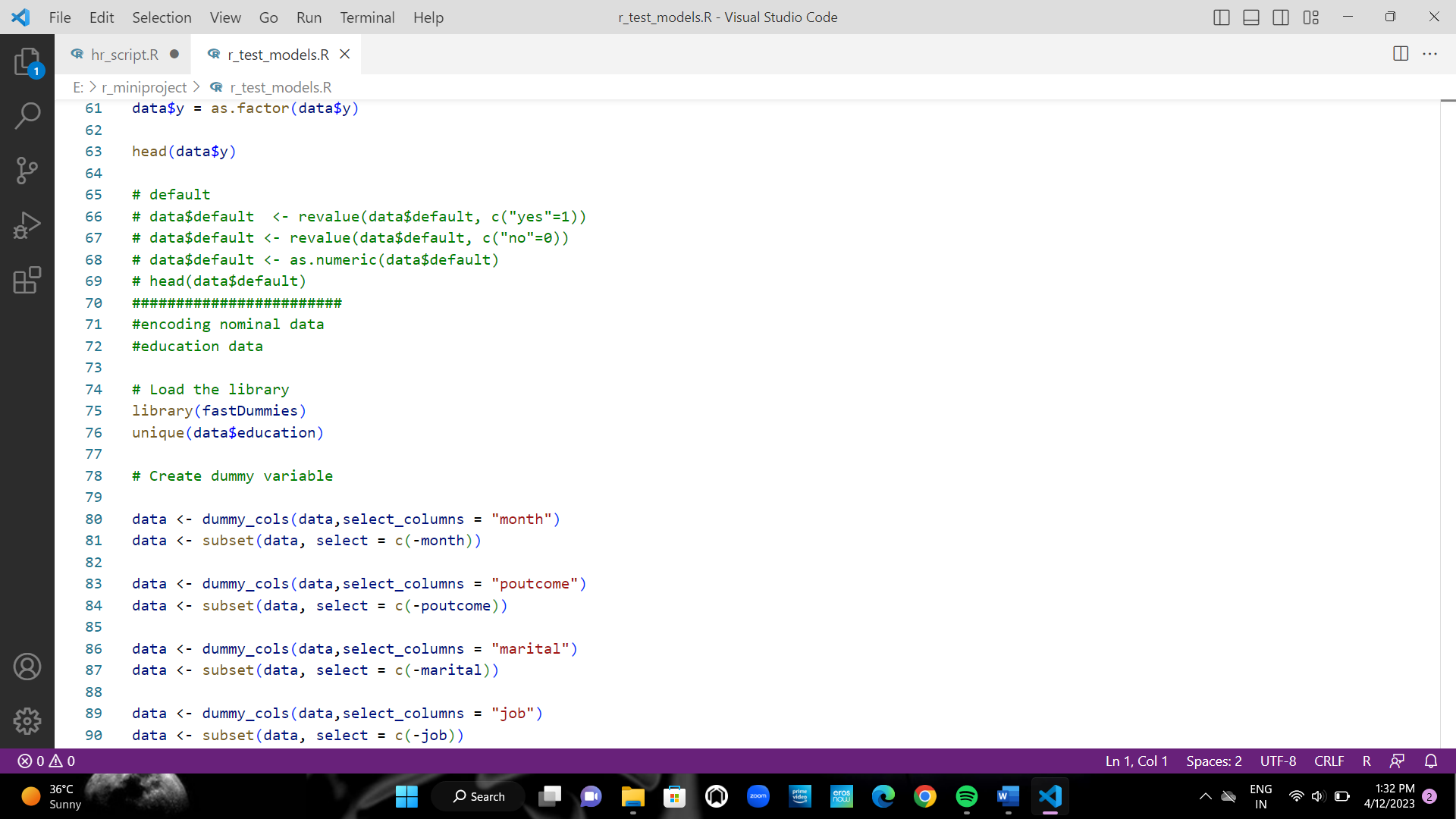
choosing through each and every customer. Previous campaign data which has been made

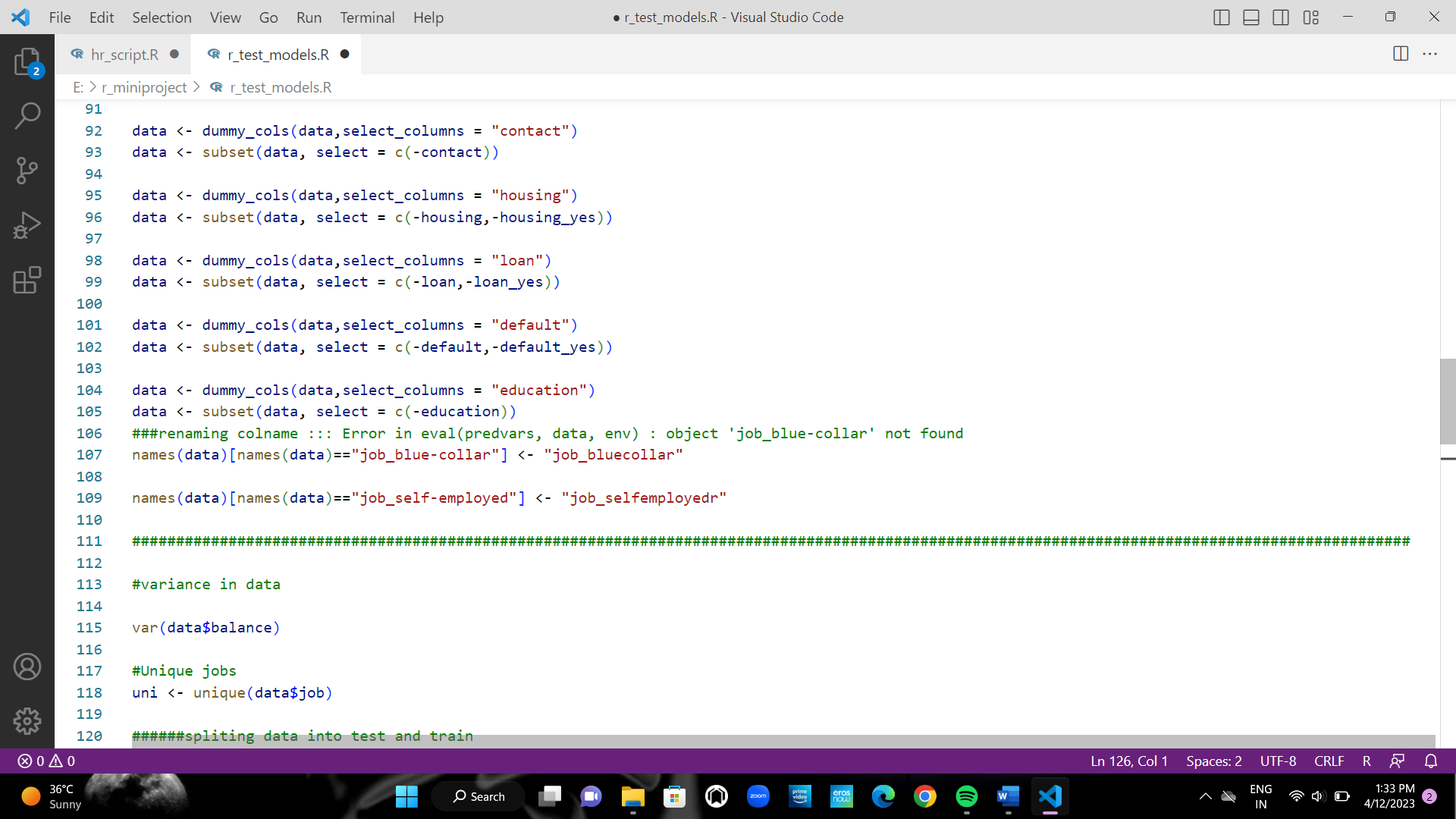
available to her; contains customer characteristics, campaign characteristics, previous campaign information as well as whether customer ended up subscribing to the product as a result of that campaign or not. Using this she plans to develop a statistical model which given this information predicts whether customer in question will subscribe to the product or not. A successful model which is able to do this, will make her campaign efficiently targeted and less bothering to uninterested customers. Use two datasets, bank-full\_train.csv and bank-full\_test.csv. You need to use data bank-full\_train to build predictive model for response variable “y”. bank- full\_test data contains all other factors except “y”, you need to predict that using the model that you developed and submit your predicted values in a csv files.

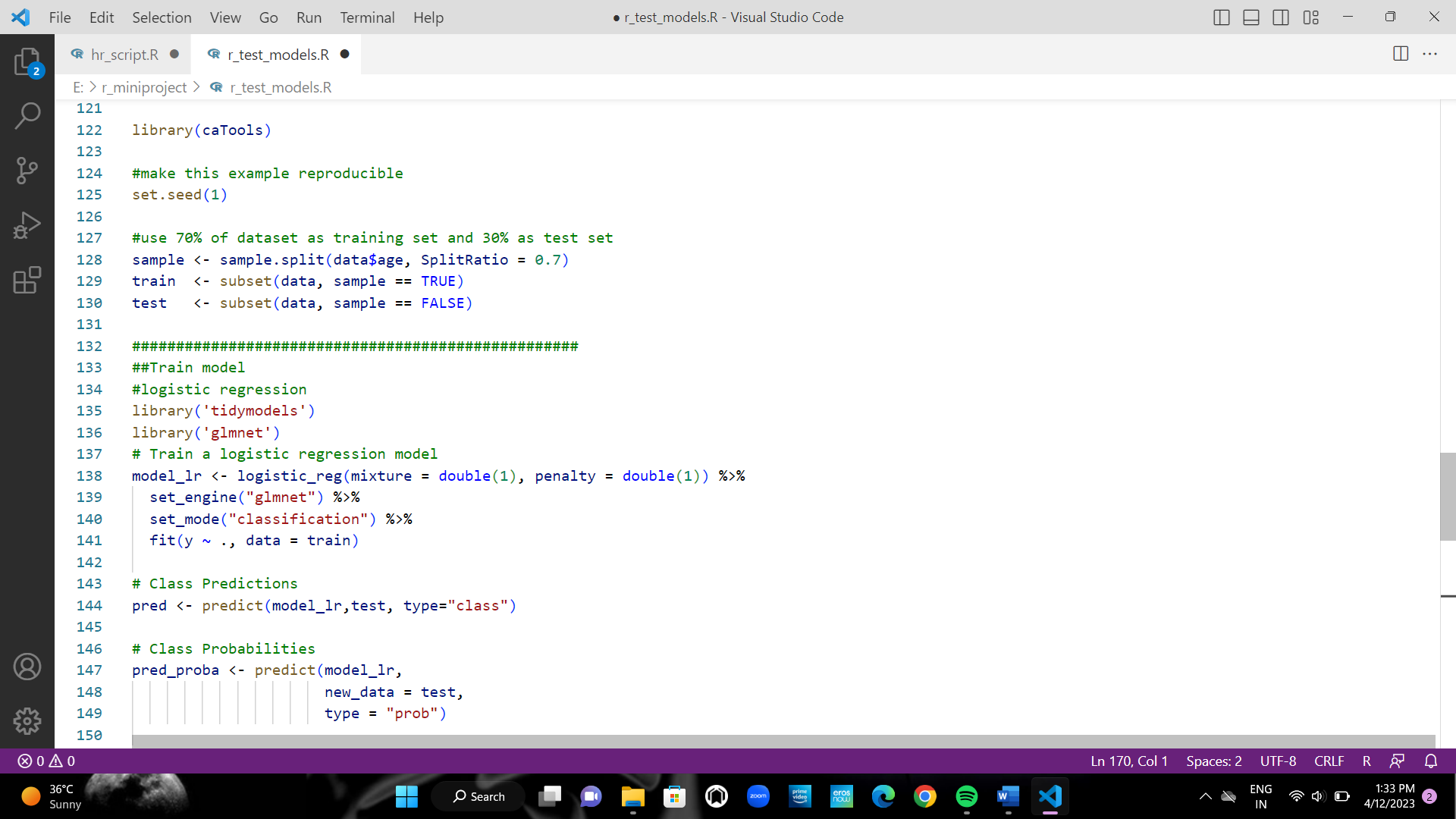
**SOURCE CODE**

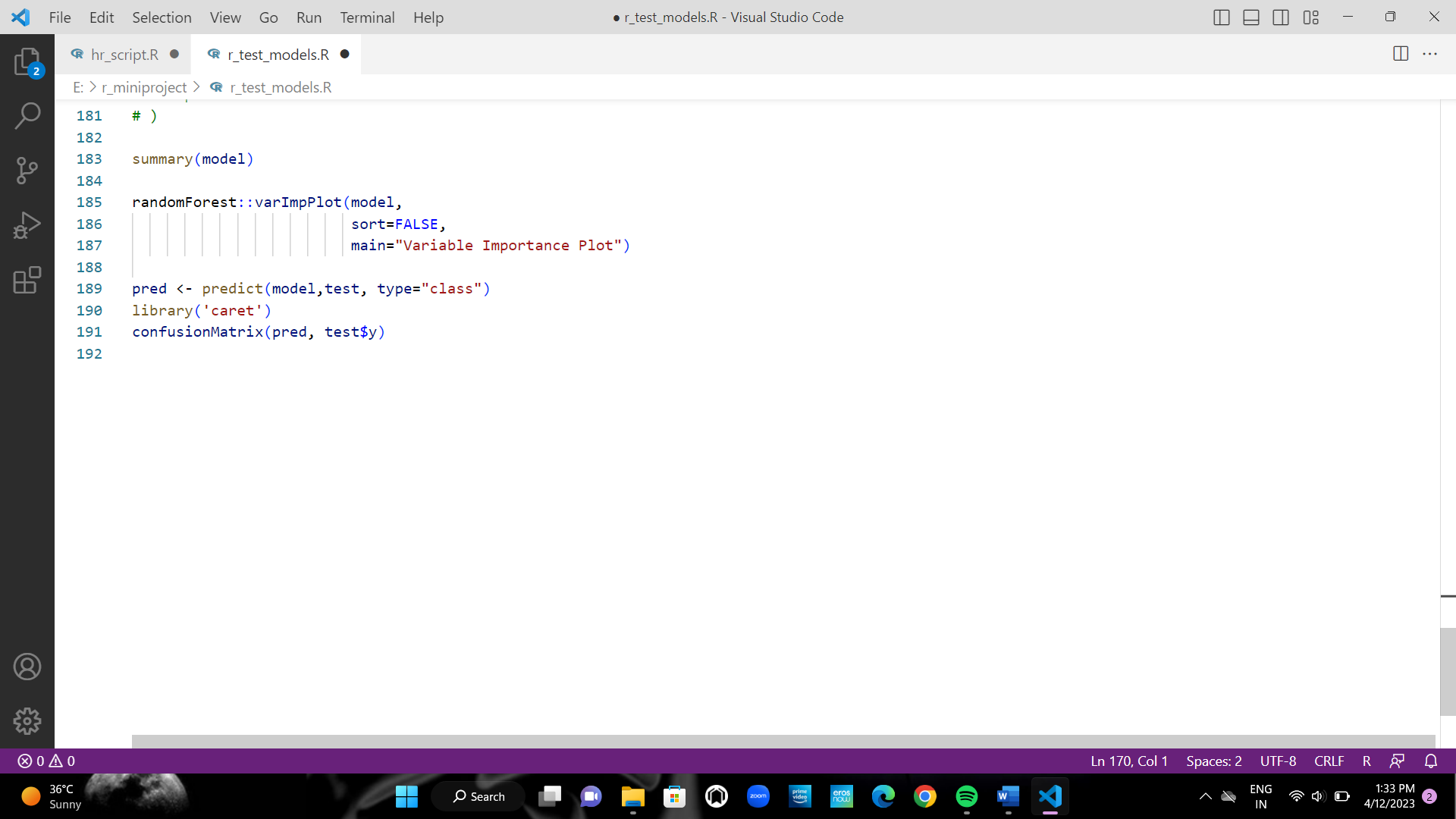












**OUTPUT**

#loading dataset

> library(readr)

> data <- read\_csv("E:\\r\_miniproject\\bank-full\_train.csv")

**Rows:** 31647 **Columns:** 18

**Delimiter:** ","

chr (10): job, marital, education, default, housing, loan, contact, month, poutcome, y

dbl (8): age, balance, day, duration, campaign, pdays, previous, ID

> #colnames to drop

> colnames(data)

[1] "age" "job" "marital" "education" "default" "balance" "housing"

[8] "loan" "contact" "day" "month" "duration" "campaign" "pdays"

[15] "previous" "poutcome" "ID" "y"

> #checking duplicates in "ID"

> sum(duplicated(data$ID))

[1] 0

> remove\_cols <- c("ID")

> #remove columns in list

> data <- subset(data, select = !(names(data) %in% remove\_cols))

> ## Check if any imbalanced classes in dataset

>

>

> #library(tidymodels)

> library(ggplot2)

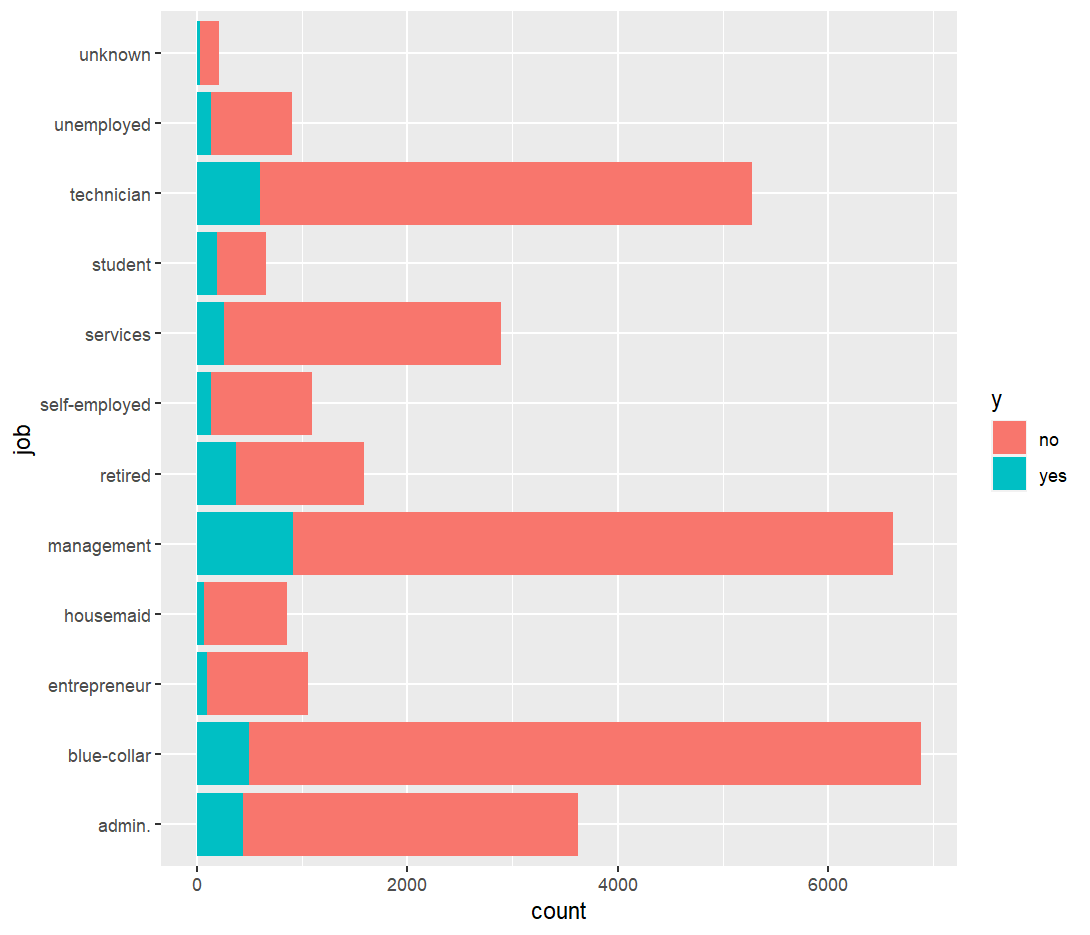
> # Plot job occupation against the target variable

> #checking for class imbalance

> ggplot(data, aes(job, fill = y)) +

+ geom\_bar() +

+ coord\_fl .... [TRUNCATED]



> library(ggplot2)

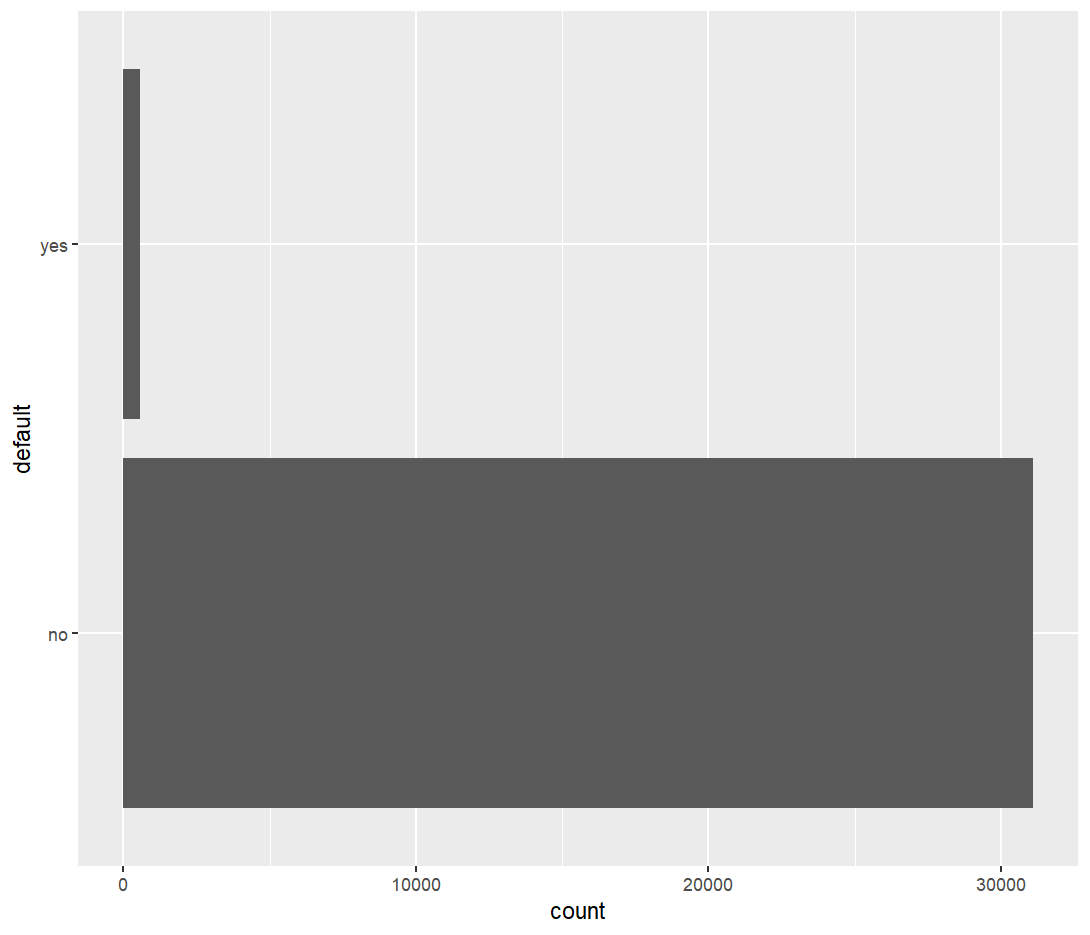
> # Plot job occupation against the target variable

> #checking for class imbalance

> ggplot(data, aes( default)) +

+ geom\_bar() +

+ coord\_flip()

****

> table(data$default)

no yes

31090 557

> #data <- subset(data, select = -default)

>

> #library(cleandata)

> #encoding ordinal data

>

> #data$poutcome <- as.numeric(factor(data$poutcome,or .... [TRUNCATED]

> #y

> data$y <- revalue(data$y, c("yes"=1))

> data$y <- revalue(data$y, c("no"=0))

> data$y <- as.numeric(data$y)

> head(data$y)

[1] 0 0 1 1 0 0

> data$y = as.factor(data$y)

> head(data$y)

[1] 0 0 1 1 0 0

Levels: 0 1

> # default

> # data$default <- revalue(data$default, c("yes"=1))

> # data$default <- revalue(data$default, c("no"=0))

> # data$default <- as.numeric .... [TRUNCATED]

> unique(data$education)

[1] "secondary" "tertiary" "primary" "unknown"

> # Create dummy variable

>

> data <- dummy\_cols(data,select\_columns = "month")

> data <- subset(data, select = c(-month))

> data <- dummy\_cols(data,select\_columns = "poutcome")

> data <- subset(data, select = c(-poutcome))

> data <- dummy\_cols(data,select\_columns = "marital")

> data <- subset(data, select = c(-marital))

> data <- dummy\_cols(data,select\_columns = "job")

> data <- subset(data, select = c(-job))

> data <- dummy\_cols(data,select\_columns = "contact")

> data <- subset(data, select = c(-contact))

> data <- dummy\_cols(data,select\_columns = "housing")

> data <- subset(data, select = c(-housing,-housing\_yes))

> data <- dummy\_cols(data,select\_columns = "loan")

> data <- subset(data, select = c(-loan,-loan\_yes))

> data <- dummy\_cols(data,select\_columns = "default")

> data <- subset(data, select = c(-default,-default\_yes))

> data <- dummy\_cols(data,select\_columns = "education")

> data <- subset(data, select = c(-education))

> ###renaming colname ::: Error in eval(predvars, data, env) : object 'job\_blue-collar' not found

> names(data)[names(data)=="job\_blue-collar"] <- "jo ..." ... [TRUNCATED]

> names(data)[names(data)=="job\_self-employed"] <- "job\_selfemployedr"

> ##################################################################################################################################################

> .... [TRUNCATED]

[1] 9273256

> #Unique jobs

> uni <- unique(data$job)

> ######spliting data into test and train

>

> library(caTools)

> #make this example reproducible

> set.seed(1)

> #use 70% of dataset as training set and 30% as test set

> sample <- sample.split(data$age, SplitRatio = 0.7)

> train <- subset(data, sample == TRUE)

> test <- subset(data, sample == FALSE)

> ###################################################

> ##Train model

> #logistic regression

> library('tidymodels')

> library('glmnet')

> # Train a logistic regression model

> model\_lr <- logistic\_reg(mixture = double(1), penalty = double(1)) %>%

+ set\_engine("glmnet") %>%

+ set\_mo .... [TRUNCATED]

> # Class Predictions

> pred <- predict(model\_lr,test, type="class")

> # Class Probabilities

> pred\_proba <- predict(model\_lr,

+ new\_data = test,

+ type = "prob")

> ## Evaluate the model performance on the testing set

> results <- test %>%

+ select(y) %>%

+ bind\_cols(pred, pred\_proba)

> accuracy(results, truth = y, estimate = .pred\_class)

# A tibble: 1 × 3

.metric .estimator .estimate

*<chr>* *<chr>* *<dbl>*

1 accuracy binary 0.896

> #######################################################################

>

>

>

> # Create confusion matrix

> conf\_mat(results, truth = y,

+ .... [TRUNCATED]

Truth

Prediction 0 1

0 8162 793

1 190 345

> # variables impacting the subscription buying decision.

> coeff <- tidy(model\_lr) %>%

+ arrange(desc(abs(estimate))) %>%

+ filter(abs(estimate) .... [TRUNCATED]

> #####################################################################

> #random forest

> library(randomForest)

> #fit the random forest model

> model <- randomForest(

+ formula = y ~ .,

+ data = train

+ )

> # model <- randomForest(

> # formula = y ~ .,

> # data = train,tree=100, keep.forest=FALSE,

> # importance=TRUE

> summary(model)

Length Class Mode

call 3 -none- call

type 1 -none- character

predicted 22157 factor numeric

err.rate 1500 -none- numeric

confusion 6 -none- numeric

votes 44314 matrix numeric

oob.times 22157 -none- numeric

classes 2 -none- character

importance 48 -none- numeric

importanceSD 0 -none- NULL

localImportance 0 -none- NULL

proximity 0 -none- NULL

ntree 1 -none- numeric

mtry 1 -none- numeric

forest 14 -none- list

y 22157 factor numeric

test 0 -none- NULL

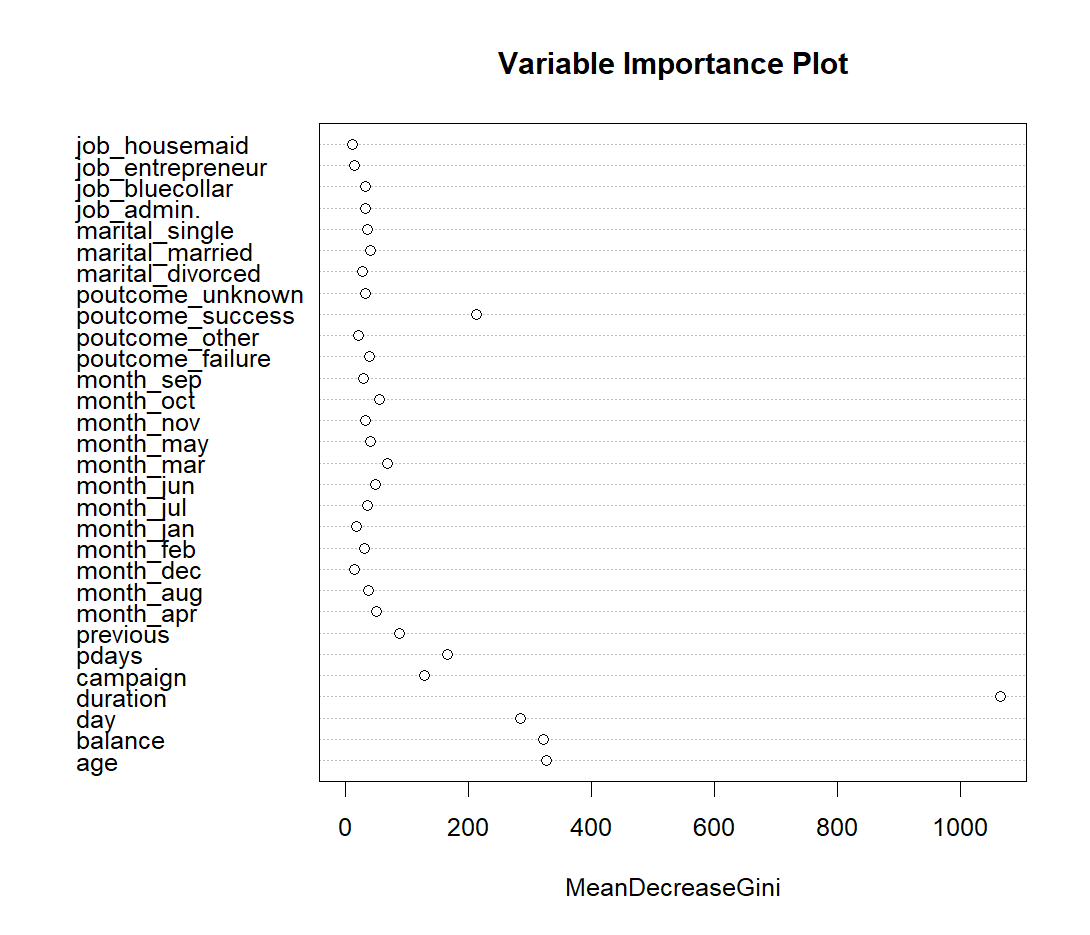
inbag 0 -none- NULL

terms 3 terms call

> randomForest::varImpPlot(model,

+ sort=FALSE,

+ main="Variable Importance Plot")



> pred <- predict(model,test, type="class")

> library('caret')

> confusionMatrix(pred, test$y)

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 8151 732

1 201 406

Accuracy : 0.9017

95% CI : (0.8955, 0.9076)

No Information Rate : 0.8801

P-Value [Acc > NIR] : 1.573e-11

Kappa : 0.4167

Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9759

Specificity : 0.3568

Pos Pred Value : 0.9176

Neg Pred Value : 0.6689

Prevalence : 0.8801

Detection Rate : 0.8589

Detection Prevalence : 0.9360

Balanced Accuracy : 0.6664

'Positive' Class : 0

will\_subscribe <- sum(test$predicted\_y == 1)

> will\_not\_subscribe <-sum(test$predicted\_y == 0)

>

> x <- c(will\_subscribe,will\_not\_subscribe)

> labels <- c("will subscribe ", "will not subscribe")

>

>

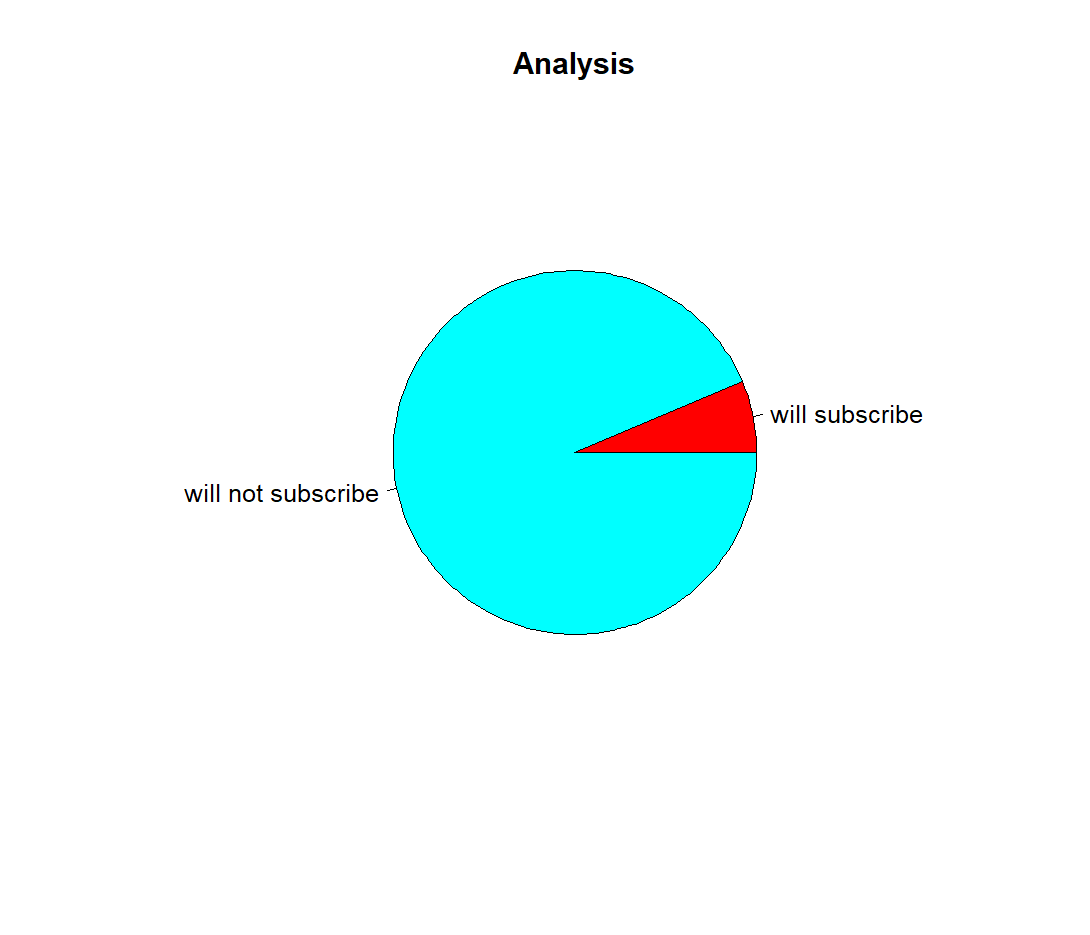
>

> # Plot the chart with title and rainbow

> # color pallet.

> pie(x, labels, main = "Analysis",

+ col = rainbow(length(x)))

****

**CONCLUSION**

We successfully built a model to churn customers for the bank based on all the dependent values. All the necessary process such as handling data, data preprocessing and plotting the model has been done under R using various useful libraries such as dplyr, caret and ggplot. The project is done using logistic regression and random forest . Here we compared these models using R function and algorithms.This project made me learn more about R programming.