GBus738\_FinalProject

Ivan Francis

2023-10-29

library("tidyverse")

## Warning: package 'tidyverse' was built under R version 4.2.3

## Warning: package 'ggplot2' was built under R version 4.2.3

## Warning: package 'tibble' was built under R version 4.2.3

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'purrr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'stringr' was built under R version 4.3.0

## Warning: package 'forcats' was built under R version 4.3.0

## Warning: package 'lubridate' was built under R version 4.2.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library("ggplot2")  
library("dplyr")  
library("summarytools")

## Warning: package 'summarytools' was built under R version 4.2.3

##   
## Attaching package: 'summarytools'  
##   
## The following object is masked from 'package:tibble':  
##   
## view

library("skimr")

## Warning: package 'skimr' was built under R version 4.2.3

library("rsample")

## Warning: package 'rsample' was built under R version 4.2.3

library("recipes")

## Warning: package 'recipes' was built under R version 4.2.3

##   
## Attaching package: 'recipes'  
##   
## The following object is masked from 'package:stringr':  
##   
## fixed  
##   
## The following object is masked from 'package:stats':  
##   
## step

library("parsnip")

## Warning: package 'parsnip' was built under R version 4.2.3

library("tidymodels")

## Warning: package 'tidymodels' was built under R version 4.2.3

## ── Attaching packages ────────────────────────────────────── tidymodels 1.1.1 ──  
## ✔ broom 1.0.5 ✔ tune 1.1.2  
## ✔ dials 1.2.0 ✔ workflows 1.1.3  
## ✔ infer 1.0.5 ✔ workflowsets 1.0.1  
## ✔ modeldata 1.2.0 ✔ yardstick 1.2.0

## Warning: package 'broom' was built under R version 4.2.3

## Warning: package 'dials' was built under R version 4.2.3

## Warning: package 'scales' was built under R version 4.3.0

## Warning: package 'infer' was built under R version 4.2.3

## Warning: package 'modeldata' was built under R version 4.2.3

## Warning: package 'tune' was built under R version 4.2.3

## Warning: package 'workflows' was built under R version 4.2.3

## Warning: package 'workflowsets' was built under R version 4.2.3

## Warning: package 'yardstick' was built under R version 4.2.3

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## ✖ scales::discard() masks purrr::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ recipes::fixed() masks stringr::fixed()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ yardstick::spec() masks readr::spec()  
## ✖ recipes::step() masks stats::step()  
## ✖ summarytools::view() masks tibble::view()  
## • Search for functions across packages at https://www.tidymodels.org/find/

library("yardstick")  
library("pROC")

## Warning: package 'pROC' was built under R version 4.2.3

## Type 'citation("pROC")' for a citation.  
##   
## Attaching package: 'pROC'  
##   
## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library("ranger")

## Warning: package 'ranger' was built under R version 4.2.3

library("randomForest")

## Warning: package 'randomForest' was built under R version 4.2.3

## randomForest 4.7-1.1  
## Type rfNews() to see new features/changes/bug fixes.  
##   
## Attaching package: 'randomForest'  
##   
## The following object is masked from 'package:ranger':  
##   
## importance  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine  
##   
## The following object is masked from 'package:ggplot2':  
##   
## margin

library("caret")

## Warning: package 'caret' was built under R version 4.2.3

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following objects are masked from 'package:yardstick':  
##   
## precision, recall, sensitivity, specificity  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library("kknn")

## Warning: package 'kknn' was built under R version 4.2.3

##   
## Attaching package: 'kknn'  
##   
## The following object is masked from 'package:caret':  
##   
## contr.dummy

loan\_data <- readRDS("loan\_data.rds")

#Exploring the dataset  
dim(loan\_data)

## [1] 4110 16

str(loan\_data)

## tibble [4,110 × 16] (S3: tbl\_df/tbl/data.frame)  
## $ loan\_default : Factor w/ 2 levels "yes","no": 1 1 2 1 2 1 1 2 2 2 ...  
## $ loan\_amount : int [1:4110] 35000 10000 28800 4475 3600 12800 35000 26000 5500 40000 ...  
## $ installment : num [1:4110] 927 260 942 165 111 ...  
## $ interest\_rate : num [1:4110] 17.25 11.5 8.97 10 9.72 ...  
## $ loan\_purpose : Factor w/ 5 levels "debt\_consolidation",..: 4 4 1 3 3 3 1 1 1 5 ...  
## $ application\_type : Factor w/ 2 levels "individual","joint": 1 1 1 1 1 1 1 1 1 1 ...  
## $ term : Factor w/ 2 levels "three\_year","five\_year": 2 2 1 1 1 2 2 2 1 2 ...  
## $ homeownership : Factor w/ 3 levels "mortgage","rent",..: 2 1 2 2 1 2 1 1 2 1 ...  
## $ annual\_income : num [1:4110] 104660 57000 160000 37000 72000 ...  
## $ current\_job\_years : num [1:4110] 2 10 10 1 4 10 0 5 4 3 ...  
## $ debt\_to\_income : num [1:4110] 29.41 23.79 5.96 13.82 22.68 ...  
## $ total\_credit\_lines : int [1:4110] 27 14 35 7 35 57 34 24 12 12 ...  
## $ years\_credit\_history: num [1:4110] 15 4 17 5 11 14 22 16 9 12 ...  
## $ missed\_payment\_2\_yr : Factor w/ 2 levels "yes","no": 2 2 2 2 2 2 2 2 2 2 ...  
## $ history\_bankruptcy : Factor w/ 2 levels "yes","no": 2 2 1 2 2 2 2 2 2 2 ...  
## $ history\_tax\_liens : Factor w/ 2 levels "yes","no": 2 2 2 2 2 2 2 2 2 2 ...

head(loan\_data)

## # A tibble: 6 × 16  
## loan\_default loan\_amount installment interest\_rate loan\_purpose   
## <fct> <int> <dbl> <dbl> <fct>   
## 1 yes 35000 927. 17.2 small\_business   
## 2 yes 10000 260. 11.5 small\_business   
## 3 no 28800 942. 8.97 debt\_consolidation  
## 4 yes 4475 165. 10 medical   
## 5 no 3600 111. 9.72 medical   
## 6 yes 12800 389. 20 medical   
## # ℹ 11 more variables: application\_type <fct>, term <fct>, homeownership <fct>,  
## # annual\_income <dbl>, current\_job\_years <dbl>, debt\_to\_income <dbl>,  
## # total\_credit\_lines <int>, years\_credit\_history <dbl>,  
## # missed\_payment\_2\_yr <fct>, history\_bankruptcy <fct>,  
## # history\_tax\_liens <fct>

glimpse(loan\_data)

## Rows: 4,110  
## Columns: 16  
## $ loan\_default <fct> yes, yes, no, yes, no, yes, yes, no, no, no, no, …  
## $ loan\_amount <int> 35000, 10000, 28800, 4475, 3600, 12800, 35000, 26…  
## $ installment <dbl> 927.29, 259.58, 941.65, 164.99, 110.70, 389.10, 9…  
## $ interest\_rate <dbl> 17.25, 11.50, 8.97, 10.00, 9.72, 20.00, 18.25, 11…  
## $ loan\_purpose <fct> small\_business, small\_business, debt\_consolidatio…  
## $ application\_type <fct> individual, individual, individual, individual, i…  
## $ term <fct> five\_year, five\_year, three\_year, three\_year, thr…  
## $ homeownership <fct> rent, mortgage, rent, rent, mortgage, rent, mortg…  
## $ annual\_income <dbl> 104660, 57000, 160000, 37000, 72000, 73000, 16700…  
## $ current\_job\_years <dbl> 2, 10, 10, 1, 4, 10, 0, 5, 4, 3, 10, 10, 5, 10, 1…  
## $ debt\_to\_income <dbl> 29.41, 23.79, 5.96, 13.82, 22.68, 30.94, 25.91, 7…  
## $ total\_credit\_lines <int> 27, 14, 35, 7, 35, 57, 34, 24, 12, 12, 16, 9, 17,…  
## $ years\_credit\_history <dbl> 15, 4, 17, 5, 11, 14, 22, 16, 9, 12, 22, 9, 8, 17…  
## $ missed\_payment\_2\_yr <fct> no, no, no, no, no, no, no, no, no, no, no, no, n…  
## $ history\_bankruptcy <fct> no, no, yes, no, no, no, no, no, no, no, no, no, …  
## $ history\_tax\_liens <fct> no, no, no, no, no, no, no, no, no, no, no, no, n…

summary(loan\_data)

## loan\_default loan\_amount installment interest\_rate   
## yes:1530 Min. : 1000 Min. : 31.04 Min. : 4.72   
## no :2580 1st Qu.: 9600 1st Qu.: 274.82 1st Qu.: 8.22   
## Median :15000 Median : 421.97 Median :11.25   
## Mean :16693 Mean : 489.42 Mean :11.38   
## 3rd Qu.:24000 3rd Qu.: 663.99 3rd Qu.:13.75   
## Max. :40000 Max. :1566.59 Max. :20.00   
## loan\_purpose application\_type term homeownership   
## debt\_consolidation:1218 individual:3494 three\_year:2588 mortgage:1937   
## credit\_card : 879 joint : 616 five\_year :1522 rent :1666   
## medical : 635 own : 507   
## small\_business : 853   
## home\_improvement : 525   
##   
## annual\_income current\_job\_years debt\_to\_income total\_credit\_lines  
## Min. : 3000 Min. : 0.000 Min. : 0.00 Min. : 2.00   
## 1st Qu.: 45000 1st Qu.: 2.000 1st Qu.: 11.85 1st Qu.:14.00   
## Median : 65000 Median : 5.000 Median : 18.59 Median :20.00   
## Mean : 73015 Mean : 5.802 Mean : 20.04 Mean :22.47   
## 3rd Qu.: 92000 3rd Qu.:10.000 3rd Qu.: 26.13 3rd Qu.:29.00   
## Max. :200000 Max. :10.000 Max. :437.61 Max. :87.00   
## years\_credit\_history missed\_payment\_2\_yr history\_bankruptcy history\_tax\_liens  
## Min. : 3.00 yes: 470 yes: 486 yes: 60   
## 1st Qu.:11.00 no :3640 no :3624 no :4050   
## Median :14.00   
## Mean :15.76   
## 3rd Qu.:19.00   
## Max. :51.00

skim(loan\_data)

Data summary

|  |  |
| --- | --- |
| Name | loan\_data |
| Number of rows | 4110 |
| Number of columns | 16 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| factor | 8 |
| numeric | 8 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

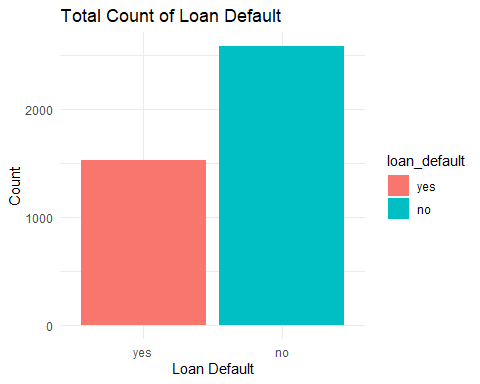
**Variable type: factor**

| skim\_variable | n\_missing | complete\_rate | ordered | n\_unique | top\_counts |
| --- | --- | --- | --- | --- | --- |
| loan\_default | 0 | 1 | FALSE | 2 | no: 2580, yes: 1530 |
| loan\_purpose | 0 | 1 | FALSE | 5 | deb: 1218, cre: 879, sma: 853, med: 635 |
| application\_type | 0 | 1 | FALSE | 2 | ind: 3494, joi: 616 |
| term | 0 | 1 | FALSE | 2 | thr: 2588, fiv: 1522 |
| homeownership | 0 | 1 | FALSE | 3 | mor: 1937, ren: 1666, own: 507 |
| missed\_payment\_2\_yr | 0 | 1 | FALSE | 2 | no: 3640, yes: 470 |
| history\_bankruptcy | 0 | 1 | FALSE | 2 | no: 3624, yes: 486 |
| history\_tax\_liens | 0 | 1 | FALSE | 2 | no: 4050, yes: 60 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| loan\_amount | 0 | 1 | 16692.79 | 10038.89 | 1000.00 | 9600.00 | 15000.00 | 24000.00 | 40000.00 | ▆▇▅▃▂ |
| installment | 0 | 1 | 489.42 | 289.50 | 31.04 | 274.82 | 421.97 | 663.98 | 1566.59 | ▇▇▅▂▁ |
| interest\_rate | 0 | 1 | 11.38 | 3.92 | 4.72 | 8.22 | 11.25 | 13.75 | 20.00 | ▆▆▇▃▃ |
| annual\_income | 0 | 1 | 73015.01 | 37203.11 | 3000.00 | 45000.00 | 65000.00 | 92000.00 | 200000.00 | ▃▇▃▁▁ |
| current\_job\_years | 0 | 1 | 5.80 | 3.69 | 0.00 | 2.00 | 5.00 | 10.00 | 10.00 | ▆▃▂▂▇ |
| debt\_to\_income | 0 | 1 | 20.04 | 14.23 | 0.00 | 11.85 | 18.59 | 26.13 | 437.61 | ▇▁▁▁▁ |
| total\_credit\_lines | 0 | 1 | 22.47 | 12.03 | 2.00 | 14.00 | 20.00 | 29.00 | 87.00 | ▇▇▂▁▁ |
| years\_credit\_history | 0 | 1 | 15.76 | 7.22 | 3.00 | 11.00 | 14.00 | 19.00 | 51.00 | ▆▇▂▁▁ |

#The dataset is already clean and does not contain any missing values so we can being Exploring the dataset now.  
  
df\_summary <- loan\_data %>%  
 group\_by(loan\_default) %>%  
 summarise(count = n())  
  
ggplot(df\_summary, aes(x = loan\_default, y = count, fill = loan\_default)) +  
 geom\_bar(stat = "identity") +  
 labs(title = "Total Count of Loan Default",  
 y = "Count",  
 x = "Loan Default") +  
 theme\_minimal()



# Calculating the percentage  
df\_summary$percentage <- (df\_summary$count / sum(df\_summary$count)) \* 100  
print(df\_summary)

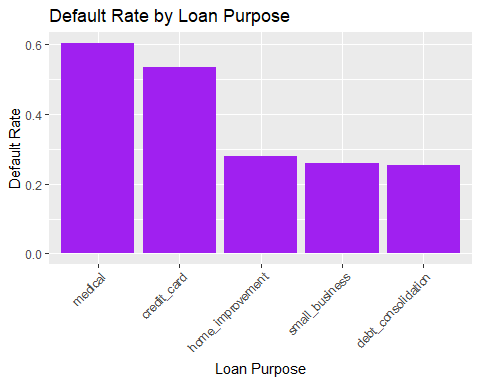
## # A tibble: 2 × 3  
## loan\_default count percentage  
## <fct> <int> <dbl>  
## 1 yes 1530 37.2  
## 2 no 2580 62.8

#Question 1 : What purpose of the loan has the highest rate of defaulting ?  
loan\_data %>%   
 group\_by(loan\_default,loan\_purpose)%>%  
 summarize(no\_of\_defaults=n())

## `summarise()` has grouped output by 'loan\_default'. You can override using the  
## `.groups` argument.

## # A tibble: 10 × 3  
## # Groups: loan\_default [2]  
## loan\_default loan\_purpose no\_of\_defaults  
## <fct> <fct> <int>  
## 1 yes debt\_consolidation 308  
## 2 yes credit\_card 470  
## 3 yes medical 384  
## 4 yes small\_business 221  
## 5 yes home\_improvement 147  
## 6 no debt\_consolidation 910  
## 7 no credit\_card 409  
## 8 no medical 251  
## 9 no small\_business 632  
## 10 no home\_improvement 378

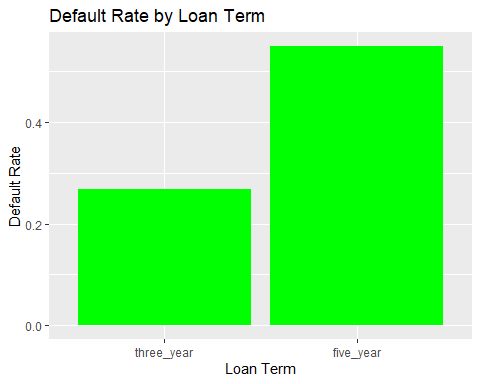
loan\_data %>%  
 group\_by(loan\_purpose) %>%  
 summarize(default\_rate = mean(loan\_default == "yes")) %>%  
 ggplot(aes(x = reorder(loan\_purpose, -default\_rate), y = default\_rate)) +  
 geom\_bar(stat = "identity", fill = "purple") +  
 labs(title = "Default Rate by Loan Purpose", x = "Loan Purpose", y = "Default Rate") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



#Question 2 : Is there a difference in default rate between loans of different terms?  
  
#Calculating the default rate for each loan term  
default\_loanterm <- loan\_data %>%  
 group\_by(term) %>%  
 summarize(default\_rate = mean(loan\_default == "yes"))  
  
default\_loanterm

## # A tibble: 2 × 2  
## term default\_rate  
## <fct> <dbl>  
## 1 three\_year 0.268  
## 2 five\_year 0.550

default\_loanterm %>%  
 ggplot(aes(x = term, y = default\_rate)) +  
 geom\_bar(stat = "identity", fill = "green") +  
 labs(title = "Default Rate by Loan Term", x = "Loan Term", y = "Default Rate") +  
 theme(axis.text.x = element\_text(angle = 0, hjust = 0.5))

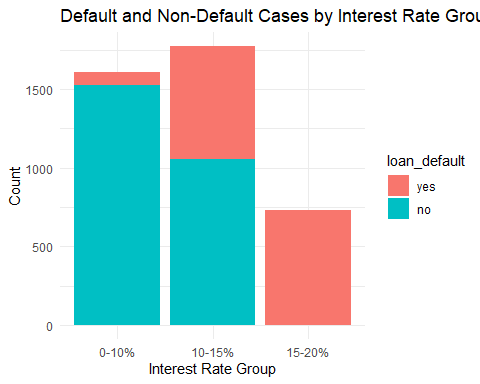


#Question 3: Did the applicants with a higher interest rate have a higher rate of default?  
  
loan\_data$Interest\_Rate\_Group <- cut(loan\_data$interest\_rate,   
 breaks = c(0, 10, 15, 20),   
 labels = c("0-10%", "10-15%", "15-20%"))  
  
table\_default\_interest <- table(loan\_data$Interest\_Rate\_Group, loan\_data$loan\_default)

default\_rates <- prop.table(table\_default\_interest, margin = 1) \* 100 # Convert to percentages  
default\_rates

##   
## yes no  
## 0-10% 4.797508 95.202492  
## 10-15% 40.665539 59.334461  
## 15-20% 100.000000 0.000000

ggplot(data = loan\_data, aes(x = Interest\_Rate\_Group, fill = loan\_default)) +  
 geom\_bar(position = "stack") +  
 labs(title ="Default and Non-Default Cases by Interest Rate Group",  
 x = "Interest Rate Group",  
 y = "Count") +  
 theme\_minimal()



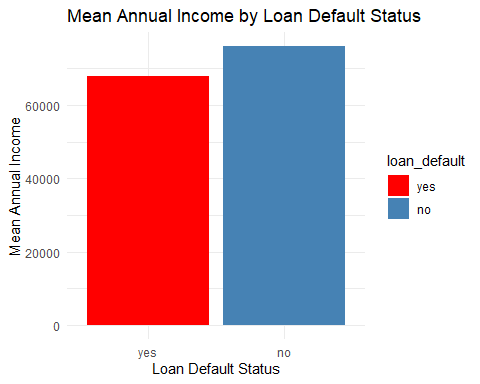
loan\_data%>%  
 group\_by(loan\_default)%>%  
 summarize(defaults = n(),avg\_interest\_rate=mean(interest\_rate))

## # A tibble: 2 × 3  
## loan\_default defaults avg\_interest\_rate  
## <fct> <int> <dbl>  
## 1 yes 1530 14.9   
## 2 no 2580 9.30

#Question 4: Did applicants with a lower average annual income have a higher rate of default?  
income\_summary <- loan\_data%>%  
 group\_by(loan\_default)%>%  
 summarise(mean\_annual\_income = mean(annual\_income),Count=n())  
  
print(income\_summary)

## # A tibble: 2 × 3  
## loan\_default mean\_annual\_income Count  
## <fct> <dbl> <int>  
## 1 yes 67819. 1530  
## 2 no 76096. 2580

income\_plot <- ggplot(income\_summary, aes(x = loan\_default, y = mean\_annual\_income, fill = loan\_default)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(title = "Mean Annual Income by Loan Default Status",  
 x = "Loan Default Status",  
 y = "Mean Annual Income") +  
 scale\_fill\_manual(values = c("yes" = "red", "no" = "steelblue")) + # Customize colors  
 theme\_minimal()  
  
income\_plot



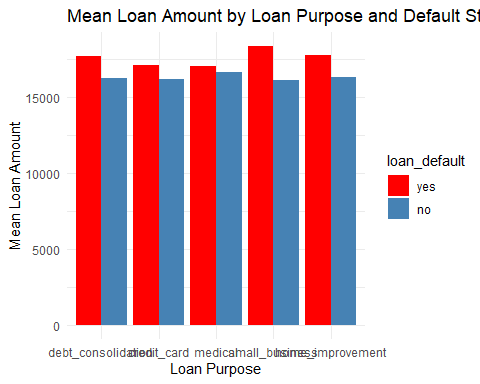
#Question 5: Is there a relationship between defaulting on the loan, loan purpose and loan amount?  
loan\_sum\_5 <- loan\_data %>%  
 group\_by(loan\_default, loan\_purpose) %>%  
 summarise(mean\_loan\_amount = mean(loan\_amount), Count = n())

## `summarise()` has grouped output by 'loan\_default'. You can override using the  
## `.groups` argument.

loan\_sum\_5

## # A tibble: 10 × 4  
## # Groups: loan\_default [2]  
## loan\_default loan\_purpose mean\_loan\_amount Count  
## <fct> <fct> <dbl> <int>  
## 1 yes debt\_consolidation 17704. 308  
## 2 yes credit\_card 17076. 470  
## 3 yes medical 17058. 384  
## 4 yes small\_business 18351. 221  
## 5 yes home\_improvement 17755. 147  
## 6 no debt\_consolidation 16224. 910  
## 7 no credit\_card 16173. 409  
## 8 no medical 16635. 251  
## 9 no small\_business 16116. 632  
## 10 no home\_improvement 16330. 378

loan\_plot <- ggplot(loan\_sum\_5, aes(x = loan\_purpose, y = mean\_loan\_amount, fill = loan\_default)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(title = "Mean Loan Amount by Loan Purpose and Default Status",  
 x = "Loan Purpose",  
 y = "Mean Loan Amount") +  
 scale\_fill\_manual(values = c("yes" = "red", "no" = "steelblue")) + # Customize colors  
 theme\_minimal()  
  
loan\_plot



#Initializing split and feature selection   
  
set.seed(123)  
loans\_split <- initial\_split(loan\_data , prop = 0.80, strata = loan\_default )  
loan\_train <- loans\_split %>% training()  
loan\_test <- loans\_split %>% testing()  
set.seed(123)

#Cross validation fold of 10 is selected  
loans\_folds <- vfold\_cv(loan\_train, v = 10)  
  
#Defining a recipe for preprocessing the data  
loans\_rec <- recipe(loan\_default ~., data = loan\_train) %>%  
 step\_YeoJohnson(all\_numeric(), -all\_outcomes()) %>%  
 step\_normalize(all\_numeric(), -all\_outcomes()) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
#Training data is preprocessed  
loans\_rec%>%  
 prep() %>%  
 bake(new\_data = loan\_train)

## # A tibble: 3,288 × 22  
## loan\_amount installment interest\_rate annual\_income current\_job\_years  
## <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1.16 1.42 -0.566 1.90 1.10   
## 2 -1.61 -1.69 -0.357 0.177 -0.392  
## 3 -1.23 -1.21 -0.857 0.121 -0.392  
## 4 1.88 1.44 -0.0243 0.121 -0.683  
## 5 0.0108 0.0603 0.0400 -0.834 1.10   
## 6 1.64 0.991 -1.09 2.25 1.10   
## 7 -1.14 -1.10 0.0400 -1.35 1.10   
## 8 -0.557 -0.920 0.531 -1.70 -1.34   
## 9 -0.372 -0.837 -0.933 -1.18 -0.117  
## 10 -0.557 -0.543 -1.09 -0.178 1.10   
## # ℹ 3,278 more rows  
## # ℹ 17 more variables: debt\_to\_income <dbl>, total\_credit\_lines <dbl>,  
## # years\_credit\_history <dbl>, loan\_default <fct>,  
## # loan\_purpose\_credit\_card <dbl>, loan\_purpose\_medical <dbl>,  
## # loan\_purpose\_small\_business <dbl>, loan\_purpose\_home\_improvement <dbl>,  
## # application\_type\_joint <dbl>, term\_five\_year <dbl>,  
## # homeownership\_rent <dbl>, homeownership\_own <dbl>, …

#First model is a logistic regression model:  
loan\_log\_model <- logistic\_reg() %>%  
 set\_engine('glm') %>%  
 set\_mode('classification')  
  
loan\_log\_model

## Logistic Regression Model Specification (classification)  
##   
## Computational engine: glm

#workflow and model fit  
log\_pip <- workflow()%>%  
 add\_model(loan\_log\_model)%>%  
 add\_recipe(loans\_rec)  
  
model\_fit <- log\_pip%>%  
 last\_fit(split=loans\_split)

## → A | warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## There were issues with some computations A: x1There were issues with some computations A: x1

#model prediction stored in model 1  
model1 <- model\_fit %>%  
 collect\_predictions()  
  
str(model1)

## tibble [822 × 7] (S3: tbl\_df/tbl/data.frame)  
## $ id : chr [1:822] "train/test split" "train/test split" "train/test split" "train/test split" ...  
## $ .pred\_yes : num [1:822] 0.0148 0.2286 0.9125 0.7472 0.8139 ...  
## $ .pred\_no : num [1:822] 0.9852 0.7714 0.0875 0.2528 0.1861 ...  
## $ .row : int [1:822] 8 12 15 21 25 28 32 39 48 52 ...  
## $ .pred\_class : Factor w/ 2 levels "yes","no": 2 2 1 1 1 2 1 2 1 2 ...  
## $ loan\_default: Factor w/ 2 levels "yes","no": 2 1 1 1 1 2 1 2 1 2 ...  
## $ .config : chr [1:822] "Preprocessor1\_Model1" "Preprocessor1\_Model1" "Preprocessor1\_Model1" "Preprocessor1\_Model1" ...

#ROC AUC   
roc\_auc(model1, truth = loan\_default, .pred\_yes)

## # A tibble: 1 × 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 roc\_auc binary 0.988

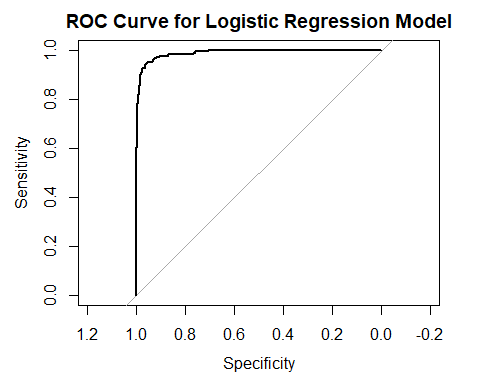
conf\_mat(model1, truth = loan\_default, estimate = .pred\_class)

## Truth  
## Prediction yes no  
## yes 281 12  
## no 25 504

roc\_obj <- roc(response = model1$loan\_default,  
 predictor = model1$.pred\_yes,  
 levels = c("no", "yes"))

## Setting direction: controls < cases

# Plot the ROC-AUC curve  
plot(roc\_obj, main = "ROC Curve for Logistic Regression Model")



#Applying KNN method as the second model for classification:  
#Creating the knn model while hypertuning  
knn\_model <- nearest\_neighbor(neighbors = tune()) %>%  
 set\_engine('kknn') %>%  
 set\_mode('classification')  
knn\_model

## K-Nearest Neighbor Model Specification (classification)  
##   
## Main Arguments:  
## neighbors = tune()  
##   
## Computational engine: kknn

#Making a workflow with the recipe defined above  
knn\_pip <- workflow() %>%  
 add\_model(knn\_model) %>%  
 add\_recipe(loans\_rec)  
k\_grid <- tibble(neighbors = c(10, 15, 25, 35, 50, 70, 100))  
#grid parameter is defined which is number of 'k' to tune over  
k\_grid

## # A tibble: 7 × 1  
## neighbors  
## <dbl>  
## 1 10  
## 2 15  
## 3 25  
## 4 35  
## 5 50  
## 6 70  
## 7 100

#Applying cross validation with k=10   
set.seed(231)  
knn\_tuning <- knn\_pip %>%  
 tune\_grid(resamples = loans\_folds, grid = k\_grid)  
best\_k <- knn\_tuning %>%  
 select\_best(metric = 'roc\_auc')

#Model fit  
final\_knn <- knn\_pip %>%  
 finalize\_workflow(best\_k)  
  
knn\_model <- final\_knn %>%  
 last\_fit(split = loans\_split)  
  
#model prediction stored in 'model2'  
model2 <- knn\_model %>%  
 collect\_predictions()  
  
model2

## # A tibble: 822 × 7  
## id .pred\_yes .pred\_no .row .pred\_class loan\_default .config   
## <chr> <dbl> <dbl> <int> <fct> <fct> <chr>   
## 1 train/test split 0.213 0.787 8 no no Preproces…  
## 2 train/test split 0.625 0.375 12 yes yes Preproces…  
## 3 train/test split 0.239 0.761 15 no yes Preproces…  
## 4 train/test split 0.538 0.462 21 yes yes Preproces…  
## 5 train/test split 0.175 0.825 25 no yes Preproces…  
## 6 train/test split 0.475 0.525 28 no no Preproces…  
## 7 train/test split 0.926 0.0737 32 yes yes Preproces…  
## 8 train/test split 0.242 0.758 39 no no Preproces…  
## 9 train/test split 0.508 0.492 48 yes yes Preproces…  
## 10 train/test split 0.253 0.747 52 no no Preproces…  
## # ℹ 812 more rows

#ROC AUC   
roc\_auc(model2, truth = loan\_default, .pred\_yes)

## # A tibble: 1 × 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 roc\_auc binary 0.938

conf\_mat(model2, truth = loan\_default, estimate = .pred\_class)

## Truth  
## Prediction yes no  
## yes 204 14  
## no 102 502

roc\_obj <- roc(response = model2$loan\_default,  
 predictor = model2$.pred\_yes,  
 levels = c("no", "yes"))

## Setting direction: controls < cases

# Plot the ROC-AUC curve  
plot(roc\_obj, main = "ROC Curve for KNN Classification Model")

