Guadalupe Torres[1]

03/02/2025

–> > The purpose of this document is to simulataneously analyze historical data containing one response and 20 predictor variables from credit card accounts for a hypothetical bank XYZ. In this task, I will be developing models that review credit card applications to determine which ones should be approved.

Exploratory Data Analysis

```
##   tot_balance      avg_bal_cards     credit_age    credit_age_good_account
##  Min.   :      0   Min.   :    0   Min.   :  0.0   Min.   :  0.0
##  1st Qu.: 92213   1st Qu.:10151   1st Qu.:231.0   1st Qu.:120.0
##  Median :107711   Median :12239   Median :280.0   Median :146.0
##  Mean   :107439   Mean   :12231   Mean   :280.7   Mean   :146.1
##  3rd Qu.:122751   3rd Qu.:14286   3rd Qu.:330.0   3rd Qu.:172.0
##  Max.   :200000   Max.   :25000   Max.   :560.0   Max.   :300.0
##
##  credit_card_age num_acc_30d_past_due_12_months num_acc_30d_past_due_6_months
##  Min.   :  0.0   Min.   :0.0000                 Min.   :0.0000
##  1st Qu.:242.0   1st Qu.:0.0000                 1st Qu.:0.0000
##  Median :285.0   Median :0.0000                 Median :0.0000
##  Mean   :285.1   Mean   :0.1565                 Mean   :0.0297
##  3rd Qu.:330.0   3rd Qu.:0.0000                 3rd Qu.:0.0000
##  Max.   :550.0   Max.   :5.0000                 Max.   :2.0000
##
##  num_mortgage_currently_past_due tot_amount_currently_past_due num_inq_12_month
##  Min.   :0.00                    Min.   :    0.0               Min.   : 0.000
##  1st Qu.:0.00                    1st Qu.:    0.0               1st Qu.: 0.000
##  Median :0.00                    Median :    0.0               Median : 0.000
##  Mean   :0.03                    Mean   :  352.5               Mean   : 0.616
##  3rd Qu.:0.00                    3rd Qu.:    0.0               3rd Qu.: 1.000
##  Max.   :1.00                    Max.   :35000.0               Max.   :10.000
##
##  num_card_inq_24_month num_card_12_month num_auto_ 36_month uti_open_card
##  Min.   : 0.000        Min.   :0.000     Min.   :0.0000     Min.   :0.0000
##  1st Qu.: 0.000        1st Qu.:0.000     1st Qu.:0.0000     1st Qu.:0.4039
##  Median : 0.000        Median :0.000     Median :0.0000     Median :0.4904
##  Mean   : 1.053        Mean   :0.273     Mean   :0.1641     Mean   :0.4909
##  3rd Qu.: 1.000        3rd Qu.:1.000     3rd Qu.:0.0000     3rd Qu.:0.5783
##  Max.   :18.000        Max.   :3.000     Max.   :2.0000     Max.   :1.0000
##
##  pct_over_50_uti  uti_max_credit_line pct_card_over_50_uti    ind_XYZ
##  Min.   :0.0000   Min.   :0.0000      Min.   :0.0000       Min.   :0.00
##  1st Qu.:0.4011   1st Qu.:0.3778      1st Qu.:0.4642       1st Qu.:0.00
##  Median :0.4855   Median :0.4648      Median :0.5518       Median :0.00
##  Mean   :0.4842   Mean   :0.4650      Mean   :0.5510       Mean   :0.25
##  3rd Qu.:0.5680   3rd Qu.:0.5536      3rd Qu.:0.6383       3rd Qu.:0.25
##  Max.   :1.0000   Max.   :1.0000      Max.   :1.0000       Max.   :1.00
##                                       NA's   :1958
##    rep_income     rep_education         Def_ind
##  Min.   : 20000   Length:20000       Min.   :0.0
##  1st Qu.:143504   Class :character   1st Qu.:0.0
##  Median :166463   Mode  :character   Median :0.0
##  Mean   :166374                      Mean   :0.1
##  3rd Qu.:188904                      3rd Qu.:0.0
##  Max.   :300000                      Max.   :1.0
##  NA's   :1559
```

```
view(loan_default_data_set)
print(str(loan_default_data_set))
```

```
## spc_tbl_ [20,000 × 21] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ tot_balance                   : num [1:20000] 102956 132759 124659 133969 143602 ...
##  $ avg_bal_cards                 : num [1:20000] 14819 18952 15348 14051 14859 ...
##  $ credit_age                    : num [1:20000] 238 384 277 375 374 250 249 252 263 328 ...
##  $ credit_age_good_account       : num [1:20000] 104 197 110 224 155 178 132 139 102 169 ...
##  $ credit_card_age               : num [1:20000] 264 371 288 343 278 255 251 269 269 328 ...
##  $ num_acc_30d_past_due_12_months : num [1:20000] 0 0 0 0 0 1 0 0 0 0 ...
##  $ num_acc_30d_past_due_6_months  : num [1:20000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ num_mortgage_currently_past_due: num [1:20000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ tot_amount_currently_past_due  : num [1:20000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ num_inq_12_month              : num [1:20000] 0 0 0 2 0 0 0 0 0 0 ...
##  $ num_card_inq_24_month         : num [1:20000] 0 0 0 2 0 1 0 0 0 0 ...
##  $ num_card_12_month             : num [1:20000] 1 0 0 1 0 0 0 0 0 0 ...
##  $ num_auto_ 36_month            : num [1:20000] 0 0 0 0 0 0 0 0 1 0 ...
##  $ uti_open_card                 : num [1:20000] 0.367 0.491 0.359 0.7 0.647 ...
##  $ pct_over_50_uti               : num [1:20000] 0.342 0.541 0.339 0.684 0.511 ...
##  $ uti_max_credit_line           : num [1:20000] 0.514 0.418 0.342 0.543 0.633 ...
##  $ pct_card_over_50_uti          : num [1:20000] 0.551 NA 0.451 0.608 0.574 ...
##  $ ind_XYZ                       : num [1:20000] 0 0 0 0 0 0 0 1 0 1 ...
##  $ rep_income                    : num [1:20000] 118266 89365 201365 191794 161465 ...
##  $ rep_education                 : chr [1:20000] "college" "college" "college" "college" ...
##  $ Def_ind                       : num [1:20000] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   tot_balance = col_double(),
##   ..   avg_bal_cards = col_double(),
##   ..   credit_age = col_double(),
##   ..   credit_age_good_account = col_double(),
##   ..   credit_card_age = col_double(),
##   ..   num_acc_30d_past_due_12_months = col_double(),
##   ..   num_acc_30d_past_due_6_months = col_double(),
##   ..   num_mortgage_currently_past_due = col_double(),
##   ..   tot_amount_currently_past_due = col_double(),
##   ..   num_inq_12_month = col_double(),
##   ..   num_card_inq_24_month = col_double(),
##   ..   num_card_12_month = col_double(),
##   ..   `num_auto_ 36_month` = col_double(),
##   ..   uti_open_card = col_double(),
##   ..   pct_over_50_uti = col_double(),
##   ..   uti_max_credit_line = col_double(),
##   ..   pct_card_over_50_uti = col_double(),
##   ..   ind_XYZ = col_double(),
##   ..   rep_income = col_double(),
##   ..   rep_education = col_character(),
##   ..   Def_ind = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
## NULL
```

–> The data set has 20,000 observations with 21 columns, the column categories consist of `tot_balance`, `avg_bal_cards`, `credit_age`, `credit_age_good_account` `credit_card_age`, `num_acc_30d_past_due_12_months`, `num_acc_30d_past_due_6_months`, `num_mortgage_currently_past_due` `tot_amount_currently_past_due` `num_inq_12_month` `num_card_inq_24_month`, `num_card_12_month` `num_auto_ 36_month`, `uti_open_card`, `pct_over_50_uti`, `uti_max_credit_line`, `pct_card_over_50_uti`, `ind_XYZ`, `rep_income`, `rep_education`, and `Def_ind`. We can see that almost all the variables are numeric data types, r represents these variables as col_double. There is one varaible that is a character type, r represents these variables as col_character.

Checking for Missing Values:

```
sum(is.na(loan_default_data_set))
```

```
## [1] 3518
```

–> The data set contains 3,518 missing values. We can create a function to deal with NAs for the variable we assign into the function. Then we assign multiple variables to the function at the same time. The result would clean up the data set to replace the NAs to “UNK”.

```
NAs <- function(x) (x = as.factor(ifelse(is.na(as.character(x)),'UNK',as.character(x))))
```

```
loan_default_data_set <- loan_default_data_set %>%
  mutate_at(c('tot_balance','avg_bal_cards','credit_age','credit_age_good_account', 'credit_card_age', 'num_acc_30d_past_due_12_months', 'num_acc_30d_past_due_6_months', 'num_mortgage_currently_past_due', 'tot_amount_currently_past_due', 'num_inq_12_month', 'num_card_inq_24_month', 'num_card_12_month', 'num_auto_ 36_month', 'uti_open_card', 'pct_over_50_uti', 'uti_max_credit_line', 'pct_card_over_50_uti', 'ind_XYZ', 'rep_income', 'rep_education', 'Def_ind'), NAs)
```

```
loan_default_data_set %>%
  keep(is.factor) %>%
  summary()
```

```
##       tot_balance        avg_bal_cards      credit_age
##  0           :    1   13559.79148:    2   295    :  128
##  100004.5358:    1   0           :    1   266    :  126
##  100007.2549:    1   10000.11981:    1   250    :  121
##  100008.208 :    1   10000.47475:    1   277    :  121
##  100009.809 :    1   10001.11235:    1   274    :  119
##  100015.9177:    1   10001.57879:    1   271    :  118
##  (Other)    :19994   (Other)    :19993   (Other):19267
##  credit_age_good_account credit_card_age num_acc_30d_past_due_12_months
##  143    :  223           274    :  145   0:17614
##  145    :  222           285    :  145   1: 1763
##  155    :  222           303    :  142   2:  515
##  144    :  215           270    :  141   3:   97
##  139    :  214           306    :  140   4:    8
##  154    :  213           288    :  138   5:    3
##  (Other):18691           (Other):19149
##  num_acc_30d_past_due_6_months num_mortgage_currently_past_due
##  0:19429                       0:19400
##  1:  548                       1:  600
##  2:   23
##
##
##
##
##  tot_amount_currently_past_due num_inq_12_month num_card_inq_24_month
##  0          :18599             0      :14043    0      :13193
##  0.3470184  :    1             1      : 2489    1      : 2278
##  1.201227647:    1             2      : 1742    2      : 1398
##  10.34010417:    1             3      :  991    3      :  892
##  10.88787059:    1             4      :  440    4      :  689
##  10011.81487:    1             5      :  194    5      :  486
##  (Other)    : 1396             (Other):  101    (Other): 1064
##  num_card_12_month num_auto_ 36_month     uti_open_card      pct_over_50_uti
##  0:14923           0:16766            0          :    1   0          :    1
##  1: 4701           1: 3185            0.042588038:    1   0.033488031:    1
##  2:  370           2:   49            0.049576123:    1   0.038814844:    1
##  3:    6                              0.053874452:    1   0.053363192:    1
##                                       0.055395529:    1   0.060083003:    1
##                                       0.057160698:    1   0.074805583:    1
##                                       (Other)    :19994   (Other)    :19994
##   uti_max_credit_line  pct_card_over_50_uti ind_XYZ         rep_income
##  0          :    1    UNK        : 1958     0:15000   UNK        : 1559
##  0.00015194 :    1    0          :    1     1: 5000   166074.9697:    2
##  0.005411234:    1    0.067348658:    1               100000.6275:    1
##  0.007336745:    1    0.080766515:    1               100027.3596:    1
##  0.0165712  :    1    0.101132448:    1               100038.9923:    1
##  0.016964319:    1    0.10859631 :    1               100051.8699:    1
##  (Other)    :19994    (Other)    :18037               (Other)    :18435
##      rep_education   Def_ind
##  college    :12137   0:18000
##  graduate   : 2406   1: 2000
##  high_school: 5314
##  other      :  142
##  UNK        :    1
##
##
```

```
sum(is.na(loan_default_data_set))
```

```
## [1] 0
```

–> We can see that the function successfully removed all missing values.

Checking for Duplicate Values:

```
x <- c(1, 1, 4, 5, 4, 6)
duplicated(x)
```

```
## [1] FALSE  TRUE FALSE FALSE  TRUE FALSE
```

```
x[duplicated(x)]
```

```
## [1] 1 4
```

–> To check for duplicates the code will return true if there are no duplicates and false if there is. We can see that in this case there are duplicate values in the dataset.

Plotting Variables:

loan_default_data_set %>% ggplot() + geom_histogram(mapping = aes(x = rep_education), bins = 30, fill = “yellow”, color = “black”) + labs(title = “Histogram of Education Level”, x = “Rep Education”, y = “Frequency”) + theme_minimal() + theme(text = element_text(size = 14))

loan_default_data_set %>% ggplot() + geom_bar(mapping = aes(x = rep_income, fill = Def_Ind), color = “black”) + labs(title = “Stacked Bar Chart of Rep Income by Def_Ind”, x = “Education Level”, y = “Number of defaults”) + coord_flip() + theme_minimal() + theme(text = element_text(size = 14))

–> This plot gives us insight whether income plays a part if someone will default in their loan. This will be helpful for the bank to know when someone is applying for a loan, and income will have to be a variable they will have to take into account.

–>

This graph gives us insight on the the different education levels from applicants. This is helpful to see what type of applicants are applying to loans and gives us insight whether or not there is a correlation between them. For example we might see a lot of a certain education level requesting loans.

Which Education Level is Underrespresented in the Data:

ggplot(test_data, aes(x = rep_education, y = as.numeric(as.character(rep_education)), color = as.factor(Def_Ind))) + geom_point(size = 3) + labs(title = “Education Level vs Default Status”, x = “Education Level (rep_education)”, y = “Def_Ind (0 = not defaulted, 1 = account defaulted)”) + scale_color_manual(values = c(“red”, “blue”), name = “Prediction”)

Checking for Balanced Classes in the Dataset

```
table(loan_default_data_set$Def_ind)
```

```
##
##     0     1
## 18000  2000
```

```
class_proportions <- prop.table(table(loan_default_data_set$Def_ind))
class_proportions
```

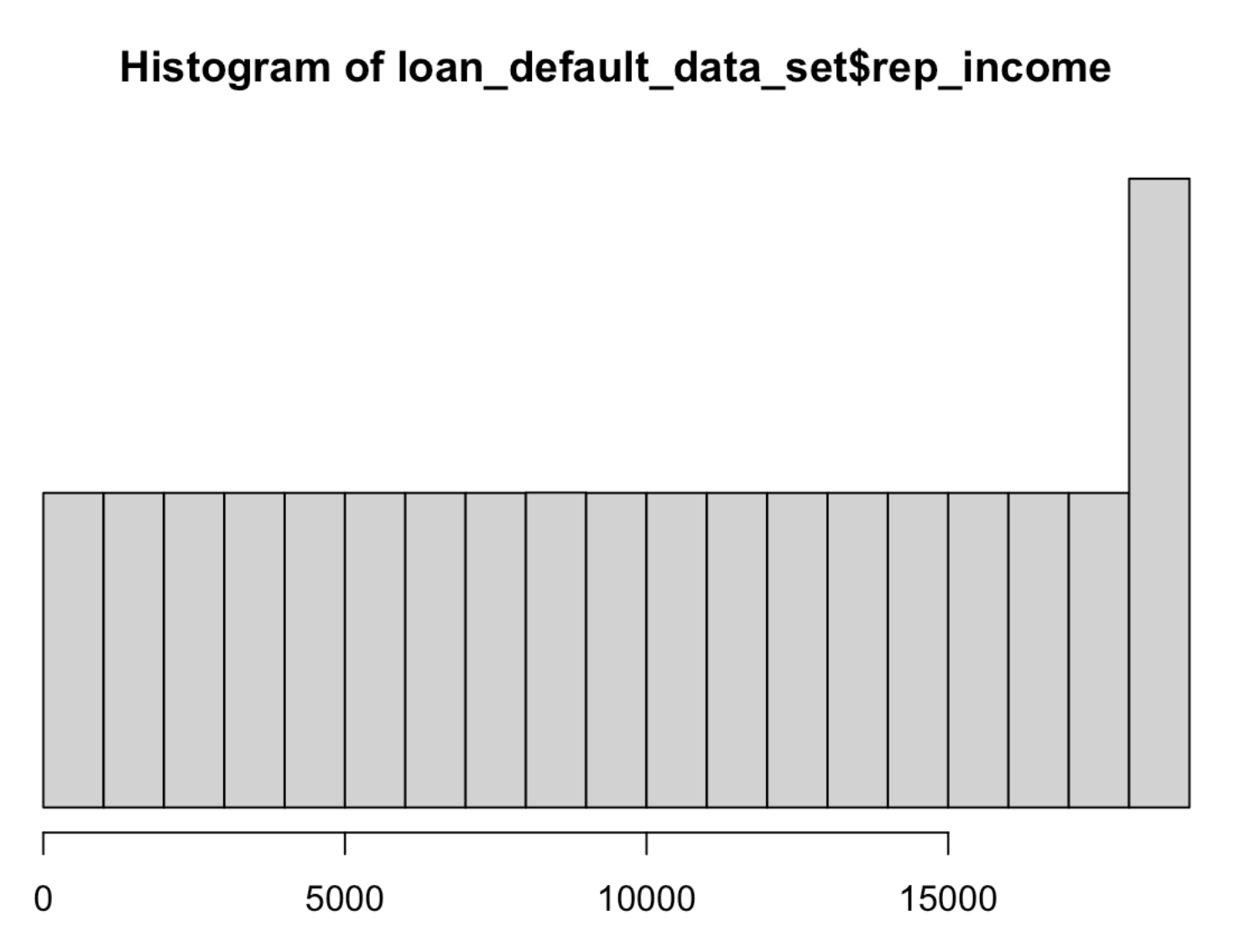
```
##
##   0   1
## 0.9 0.1
```

–> To determine if the account defaulted and not defaulted classes are balanced in the dataset, we need to check if the number of data points belonging to each class is roughly equal. In this case, one class has significalty more data points then the other making the dataset imbalanced. There are two methods you can use to handle Imbalanced Data, you can do under-sampling which is used when there is enough amount of data which will even out the dataset. The second option is over-sampling which is used when there is not enough data, it will balance out the data increasing the sample size. New rare samples are produced using techniques like bootstrapping, cross-validation or SMOTE.

Distribution of Rep_Income:

```
loan_default_data_set$rep_income <- as.numeric(loan_default_data_set$rep_income)
```

```
hist(loan_default_data_set$rep_income)
```



–> Based on the Histogram the distribution of rep income in uniformed meaning being equal spaced.

Group Default Status

library(dplyr)

df %>% group_by(Def_ind, red_education)

–> The education level more likely to default on loans is college.

Seperating Data into Training and Testing Sets:

```
set.seed(42)
```

split <- sample.split(loan_default_data_set$Def_ind, SplitRatio = 0.8)

train_data <- subset(loan_default_data_set, split == TRUE) test_data <- subset(loan_default_data_set, split == FALSE)

dim(train_data) dim(test_data)

Fitting the Model

knn_model <- train(y_variable ~ ., loan_default_data_set=train, method=‘knn’, tuneLength=5)

Confusion Matrix

pred_knn <- predict(knn_model, test) print(confusionMatrix(pred_knn, test$y_variable))

Decision Tree

dt_model <- train(y_variable ~ ., data=train, method=‘rpart’)

Confusion Matrix

pred_dt <- predict(dt_model, test) print(confusionMatrix(pred_dt, test$y_variable))

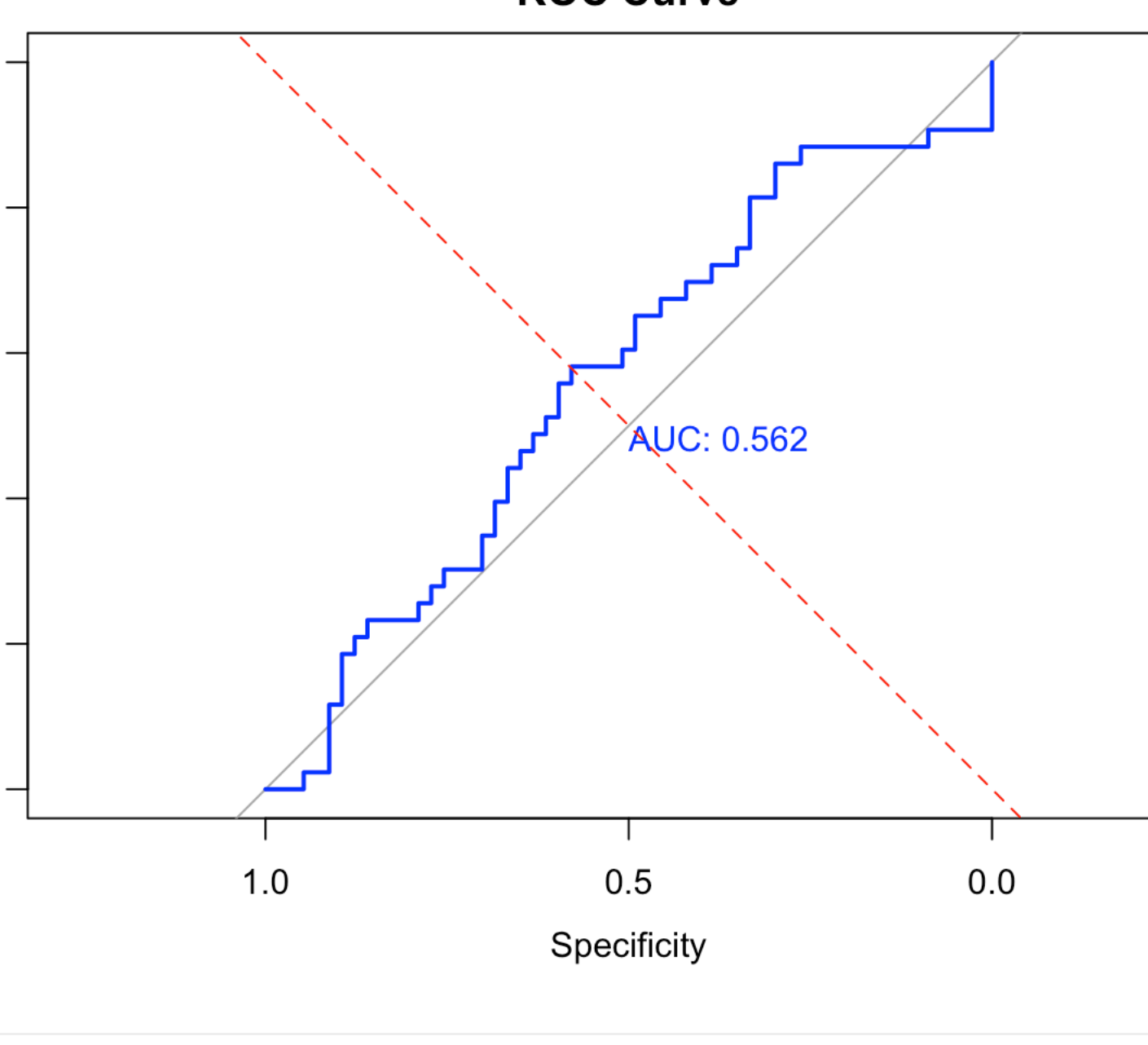
Plotting ROC Curve

```
set.seed(123)
actual <- sample(c(0, 1), 100, replace = TRUE)
```

```
predicted_probs <- runif(100)
```

```
library(pROC)
roc_curve <- roc(actual, predicted_probs)
```

```
plot(roc_curve, col = "blue", main = "ROC Curve", print.auc = TRUE)
abline(a = 0, b = 1, lty = 2, col = "red")
```



1. **Email** gtorressanchez01@hamline.edu. **Position** Student↩