Final Project

Name: Gathu Macharia

```
#install.packages("vip")
```

```
# Suppress dplyr summarise grouping warning messages
options(dplyr.summarise.inform = FALSE)

## Add R libraries here
#library(tidyverse)
library(tidymodels)
library(klaR)
library(kknn)
library(discrim)
library(vip)
library(rpart.plot)
library(ranger)
library(readr)



# Load data
loans_df <- read_rds("/home/gathu/Documents/CLASSES/data science/datascience with r/logistic  bank defa
```

Data Analysis [30 Points]

The Data Analysis section will cover 6 questions to explore the relationship between "loan_default" and the other variables in the "loan_df" data set. It will include 3 tibbles and 3 plots to exemplify and answer each related question.

Question 1

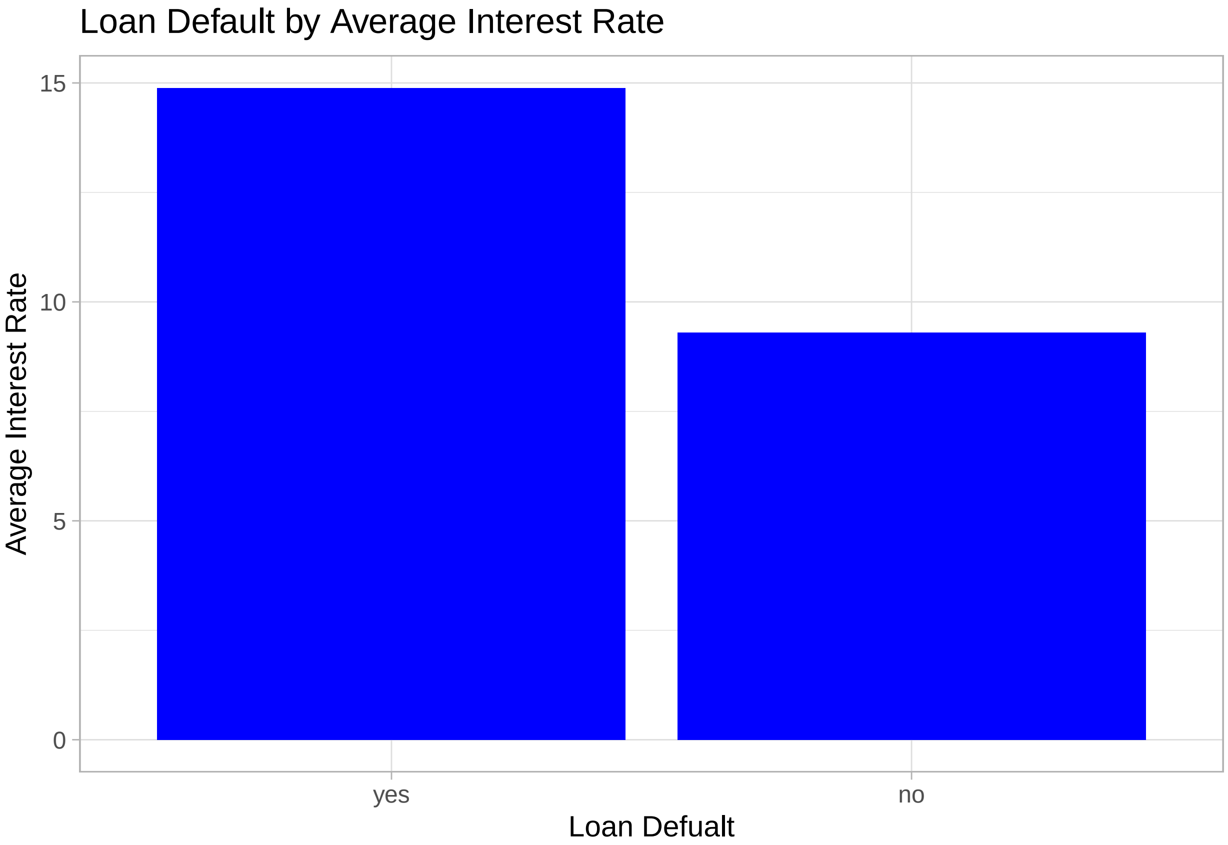
Question: Are there differences in loan default rates by interest rates?

Answer: As the plot shows, there is a difference between the average interest rates for those that do default their loan compared to those who do not. The average interest rate for a defaulted loan is just under 15%, compared to just over 9% for non defaulted loans. With almost a 6% difference between the two, it can be said that the interest rate does play a part in determining whether or not someone defaults their loans.

```
interest_rates <- loans_df %>%
  group_by(loan_default) %>%
  summarise(n_customers = n(),
            avg_int_rate = round(mean(interest_rate),2))
interest_rates
```

```
## # A tibble: 2 x 3
##   loan_default n_customers avg_int_rate
##   <fct>              <int>        <dbl>
## 1 yes                 1530         14.9
## 2 no                  2580          9.3
```

```
ggplot(data = interest_rates, mapping = aes(x = loan_default, y =avg_int_rate)) +
  geom_bar(stat = 'identity', fill = "blue") +
  labs(title = "Loan Default by Average Interest Rate",
       x = "Loan Defualt",
       y = "Average Interest Rate") +
  theme_light()
```



Question 2

Question: Does the loan purpose have an impact on the interest rate?

Answer: Based on the tibble below, it can be seen that credit cards and medical purposes have the highest average interest rates, with 12.36% and 12.85% respectively. All other loan purposes have an average interest rate below 11%. The loan with the highest default rate were the medical loans, followed by credit card loans.

```
loans_df %>% group_by(loan_purpose) %>%
  summarise(avg_interest = round(mean(interest_rate), 2),
            avg_loan_amount = round(mean(loan_amount), 2),
            default_percent = 100 * round(mean(loan_default == "yes"),4))
```

```
## # A tibble: 5 x 4
##   loan_purpose       avg_interest avg_loan_amount default_percent
##   <fct>                     <dbl>           <dbl>           <dbl>
## 1 debt_consolidation         10.6          16599.            25.3
## 2 credit_card                12.4          16656.            53.5
## 3 medical                    12.8          16891.            60.5
## 4 small_business             10.7          16695.            25.9
## 5 home_improvement           10.9          16729.            28
```

Question 3

Question: Does the type of home ownership and average income have a relation to the average debt to income?

Answer: It can be seen that there is a relationship between home ownership, average annual income, and the average debt to income. People with mortgages have the highest annual income and the highest average debt to income, owning has the second highest average annual income and average debt to income, and renting has the lowest average annual income and average debt to income.

```
loans_df %>% group_by(homeownership) %>%
  summarise(avg_ann_income = round(mean(annual_income),2),
            avg_debt_to_income = round(mean(debt_to_income),2),
            default_percent = 100 * mean(loan_default == "yes")) %>%
  arrange(desc(avg_debt_to_income))
```

```
## # A tibble: 3 x 4
##   homeownership avg_ann_income avg_debt_to_income default_percent
##   <fct>                  <dbl>              <dbl>           <dbl>
## 1 mortgage              81239.               21.3            32.4
## 2 own                   68759.               19.3            37.3
## 3 rent                  64748.               18.8            42.8
```

Question 4

Question: Does the term impact the loan default rate?

Answer: Yes, the term does impact the default rate. Five year terms have more people defaulting, and have an average default rate of 54.99%, compared to just 26.78% defaulted by three year terms.

```
loans_df %>% group_by(term) %>%
  summarise(num_default = sum(loan_default == "yes"),
            default_percent = round(100* mean(loan_default == "yes"),2))
```

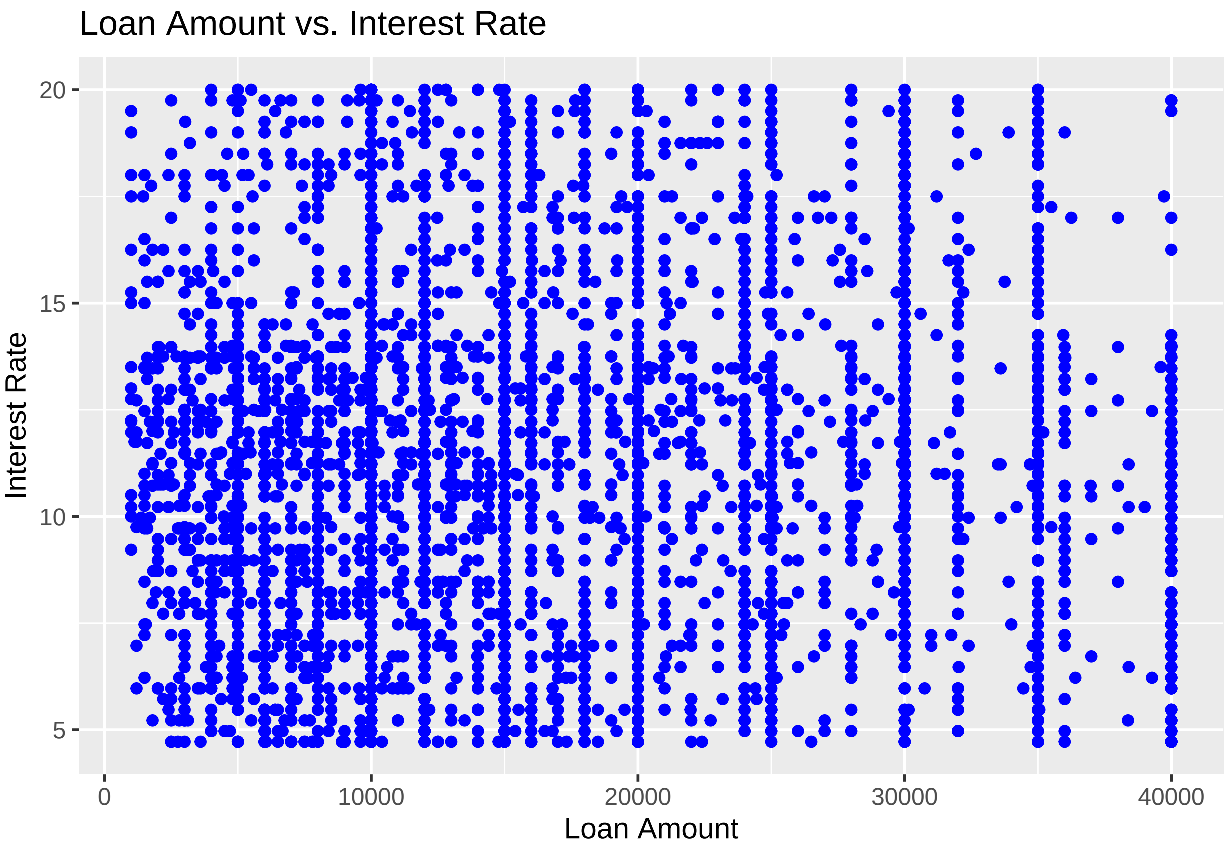
```
## # A tibble: 2 x 3
##   term       num_default default_percent
##   <fct>            <int>           <dbl>
## 1 three_year         693            26.8
## 2 five_year          837            55.0
```

Question 5

Question: Is interest rate dependent on loan amount?

Answer: When plotting the loan amounts vs. interest rates, it can be seen that no matter the amount of the loan, the interest rate can range anywhere from the lowest amount to the highest amount possible. Based on the scatter plot, it can be said that there is no relationship between the interest rate and the amount of the loan. However, there is a higher concentration of 5%-15% interest rates between $0-$15,000 loans.

```
ggplot(data = loans_df, mapping = aes(x = loan_amount, y = interest_rate)) +
  geom_point(color = "blue") +
  labs(title = "Loan Amount vs. Interest Rate",
       x = "Loan Amount",
       y = "Interest Rate")
```



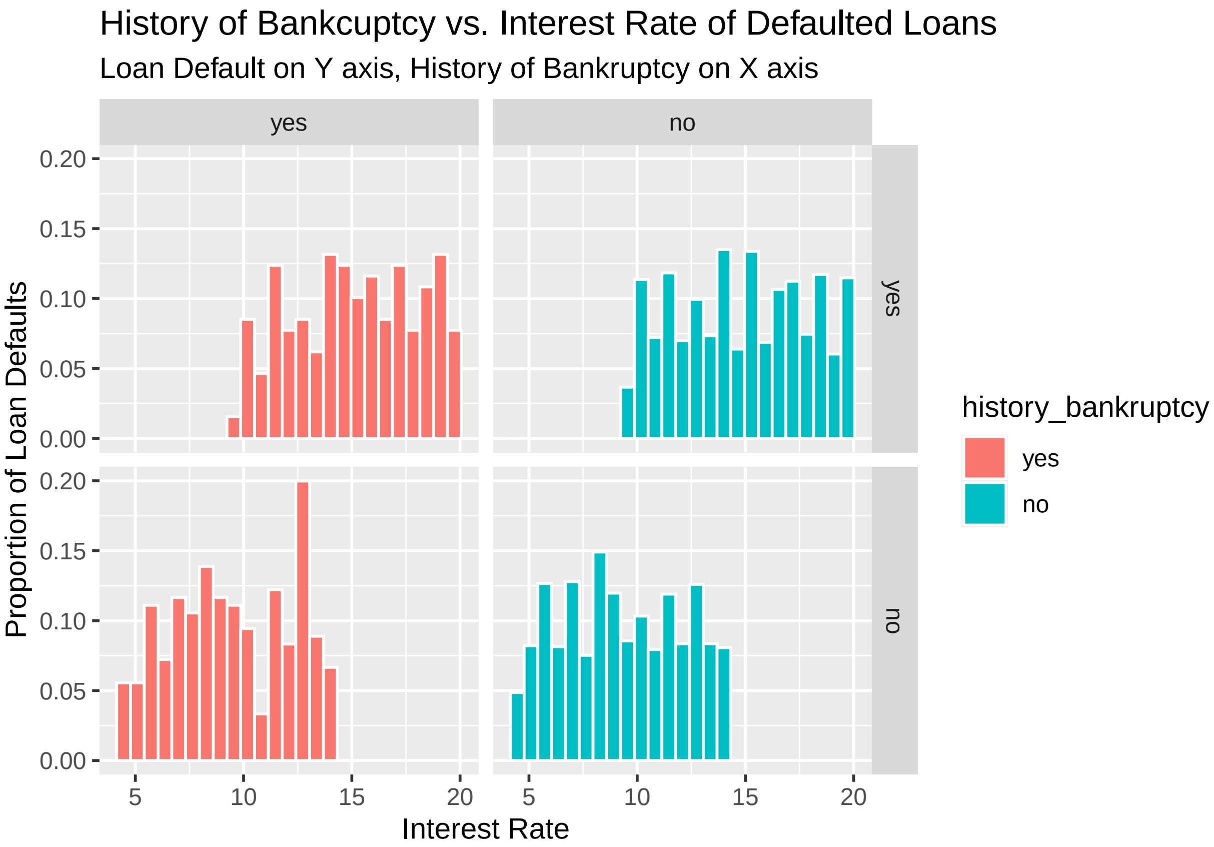
Question 6

Question: Does having a history of bankruptcy have an impact on the interest rate between those who default their loans or not?

Answer: From the histogram below, it can be concluded that whether or not someone was bankrupt does not impact their interest rate. However, whether they defaulted their loan also plays an important factor. While almost every interest rate is over 10% for a bankrupt person that defaulted their loan, for a bankrupt person who did not default their interest rate ranges between just under 5% to just under 15%; almost a 10% range. An almost identical histogram can be seen for those who were not bankrupt in regards to their interest rates. The main factor that can be seen in these histograms is that those who have defaulted their loans have a much higher interest rate.

```
ggplot(loans_df, aes(x = interest_rate, y = ..density.., color = history_bankruptcy, fill = history_ban
  geom_histogram(color = "white", bins = 25) +
  facet_grid(loan_default~history_bankruptcy) +
  labs(title = "History of Bankcuptcy vs. Interest Rate of Defaulted Loans",
       subtitle = "Loan Default on Y axis, History of Bankruptcy on X axis",
       x = "Interest Rate",
       y = "Proportion of Loan Defaults")
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

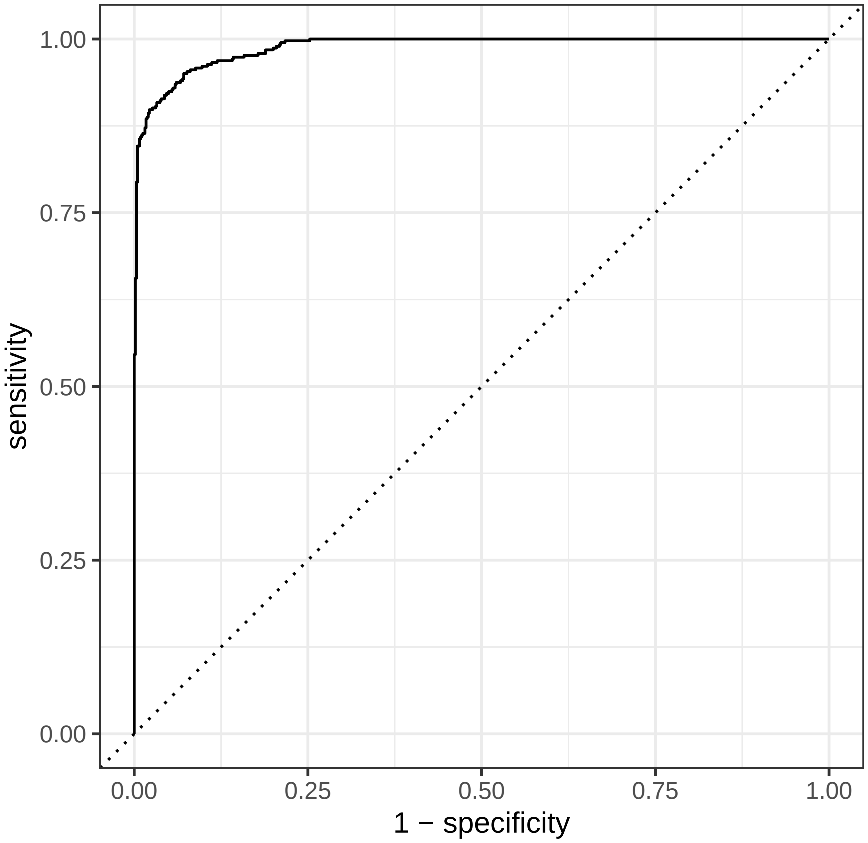


Predictive Modeling [70 Points]

Model 1: Logistic Regression

```
#Create Split, training, and test
set.seed(150)

loans_split <- initial_split(loans_df, prop = 0.75, strata = loan_default)

loans_training <- loans_split %>% training()

loans_test <- loans_split %>% testing()

#Cross validation folds for hyperparameter tuning
set.seed(75)
loans_folds <- vfold_cv(loans_training, v = 5)

#Feature Engineering
loans_recipe <- recipe(loan_default ~ ., data = loans_training) %>%
  step_YeoJohnson(all_numeric(), -all_outcomes()) %>%
  step_normalize(all_numeric(), -all_outcomes()) %>%
  step_dummy(all_nominal(), -all_outcomes())

#Logistic Regression Model Specification
logistic_model <- logistic_reg() %>%
  set_engine('glm') %>%
```

```r
  set_mode('classification')

#Create Workflow
logistic_wf <- workflow() %>%
  add_model(logistic_model) %>%
  add_recipe(loans_recipe)

#Fit Model
logistic_fit <- logistic_wf %>%
  last_fit(split = loans_split)

#Collect Predictions
logistic_results <- logistic_fit %>%
  collect_predictions()


# Assuming logistic_results is your data frame
roc_curve_data <- roc_curve(logistic_results,
                             truth = loan_default,
                             .pred_yes)

# Plotting the ROC curve
autoplot(roc_curve_data)
```



```r
#ROC Area Under Curve
roc_auc(logistic_results, truth = loan_default, .pred_yes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.988
```

```
#Confusion Matrix
conf_mat(logistic_results, truth = loan_default, .pred_class)
```

```
##           Truth
## Prediction yes  no
##        yes 352  29
##        no   31 616
```

Model 2: KNN

```
#KNN Model Specifcation
knn_model <- nearest_neighbor(neighbors = tune()) %>%
  set_engine('kknn') %>%
  set_mode('classification')

#Create Workflow
knn_wf <- workflow() %>%
  add_model(knn_model) %>%
  add_recipe(loans_recipe)

#Create grid of values to test
k_grid <- tibble(neighbors = c(10, 20, 30, 50 , 75, 100, 125, 150))

#Tune workflow
set.seed(250)

knn_tuning <- knn_wf %>%
  tune_grid(resamples = loans_folds, grid = k_grid)

#Show the top 5 best models for ROC AUC
# Show the top 5 best models for ROC AUC
knn_tuning %>% show_best(metric = "roc_auc", n = 5)
```

```
## # A tibble: 5 x 7
##   neighbors .metric .estimator  mean     n std_err .config
##       <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1       150 roc_auc binary     0.903     5 0.00501 Preprocessor1_Model8
## 2       125 roc_auc binary     0.902     5 0.00510 Preprocessor1_Model7
## 3       100 roc_auc binary     0.901     5 0.00510 Preprocessor1_Model6
## 4        75 roc_auc binary     0.899     5 0.00529 Preprocessor1_Model5
## 5        50 roc_auc binary     0.895     5 0.00529 Preprocessor1_Model4
```

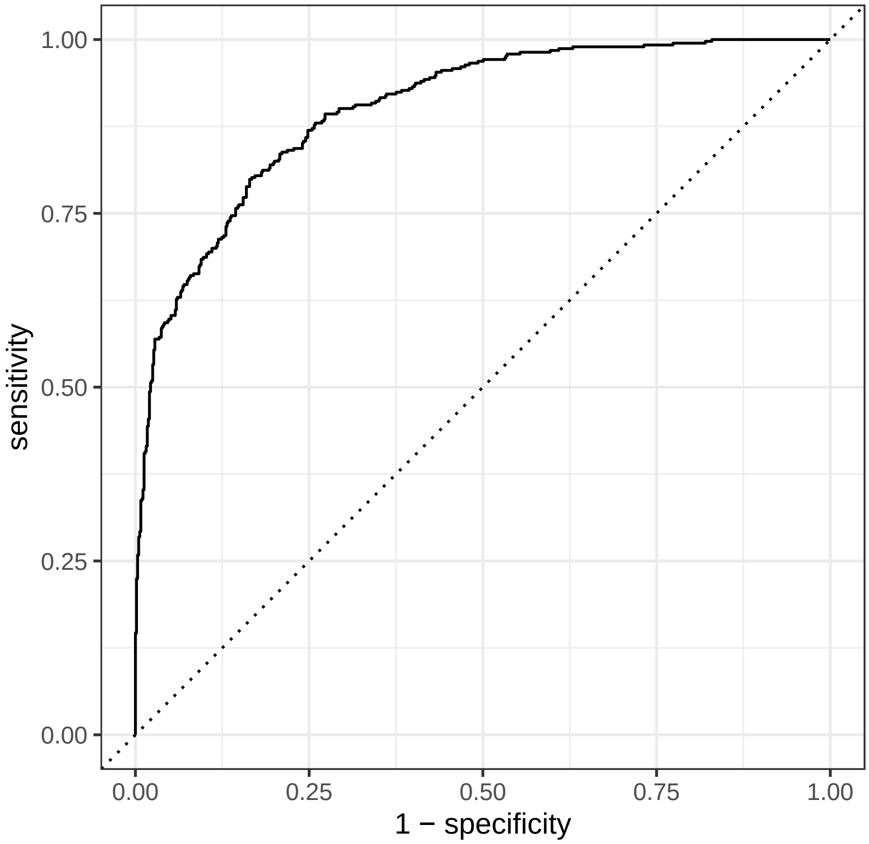
```
#Select and view the best model
best_k <- knn_tuning %>% select_best(metric = 'roc_auc')
best_k
```

```
## # A tibble: 1 x 2
##   neighbors .config
##       <dbl> <chr>
## 1       150 Preprocessor1_Model8
```

```r
#Finalize the knn workflow by adding the best model
final_knn_wf <- knn_wf %>%
  finalize_workflow(best_k)

#Train and Evaluate with last_fit()
last_fit_knn <- final_knn_wf %>%
  last_fit(split = loans_split)

#ROC Curve
knn_predictions <- last_fit_knn %>%
  collect_predictions()
knn_predictions
```

```
## # A tibble: 1,028 x 7
##    .pred_class .pred_yes .pred_no id                .row loan_default .config
##    <fct>           <dbl>    <dbl> <chr>            <int> <fct>        <chr>
##  1 yes            0.569     0.431 train/test split     1 yes          Preproces~
##  2 no             0.209     0.791 train/test split     8 no           Preproces~
##  3 no             0.0842    0.916 train/test split     9 no           Preproces~
##  4 yes            0.715     0.285 train/test split    12 yes          Preproces~
##  5 no             0.0877    0.912 train/test split    14 no           Preproces~
##  6 no             0.263     0.737 train/test split    15 yes          Preproces~
##  7 no             0.197     0.803 train/test split    20 no           Preproces~
##  8 no             0.344     0.656 train/test split    39 no           Preproces~
##  9 no             0.315     0.685 train/test split    50 no           Preproces~
## 10 no             0.438     0.562 train/test split    53 yes          Preproces~
## # i 1,018 more rows
```

```r
roc_curve_data <- roc_curve(knn_predictions,
                             truth = loan_default,
                             .pred_yes)

# Plot the ROC curve
autoplot(roc_curve_data)
```



```
roc_auc(knn_predictions, truth = loan_default, .pred_yes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.901
```

```
#Confusion Matrix
conf_mat(knn_predictions, truth = loan_default, estimate = .pred_class)
```

```
##           Truth
## Prediction yes  no
##        yes 219  22
##        no  164 623
```

Model 3: Decision Tree

```
#Tree Model Specification
tree_model <- decision_tree(cost_complexity = tune(),
                            tree_depth = tune(),
                            min_n = tune()) %>%
  set_engine('rpart') %>%
  set_mode('classification')

#Workflow
tree_workflow <- workflow() %>%
  add_model(tree_model) %>%
  add_recipe(loans_recipe)
```

```r
#Hyperparameter Tuning -  grid to test
tree_grid <- grid_regular(cost_complexity(),
                          tree_depth(),
                          min_n(),
                          levels = 2)

#Tune decision tree workflow
set.seed(300)

tree_tuning <- tree_workflow %>%
  tune_grid(resamples = loans_folds,
            grid = tree_grid)

#Show top 5 best tree based on ROC AUC
# Show the top models for ROC AUC
best_models <- tree_tuning %>% show_best(metric = "roc_auc", n = 5)

#Select the best model and show it
best_tree <- tree_tuning %>%
  select_best(metric = 'roc_auc')

best_tree
```

```
## # A tibble: 1 x 4
##   cost_complexity tree_depth min_n .config
##             <dbl>      <int> <int> <chr>
## 1    0.0000000001         15    40 Preprocessor1_Model7
```

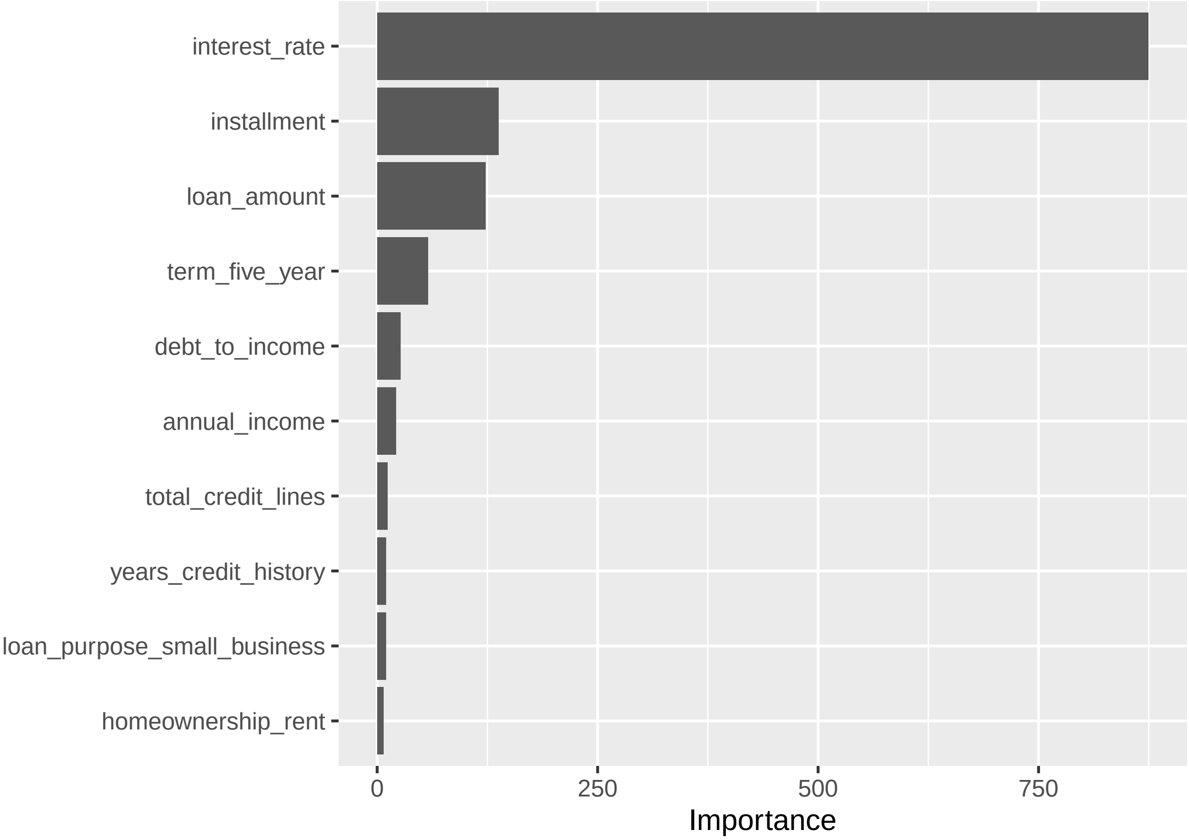
```r
#Finalize workflow with best model
final_tree_wf <- tree_workflow %>% finalize_workflow(best_tree)

final_tree_wf
```

```
## == Workflow ====================================================================
## Preprocessor: Recipe
## Model: decision_tree()
##
## -- Preprocessor ----------------------------------------------------------------
## 3 Recipe Steps
##
## * step_YeoJohnson()
## * step_normalize()
## * step_dummy()
##
## -- Model -----------------------------------------------------------------------
## Decision Tree Model Specification (classification)
##
## Main Arguments:
##   cost_complexity = 1e-10
##   tree_depth = 15
##   min_n = 40
##
## Computational engine: rpart
```

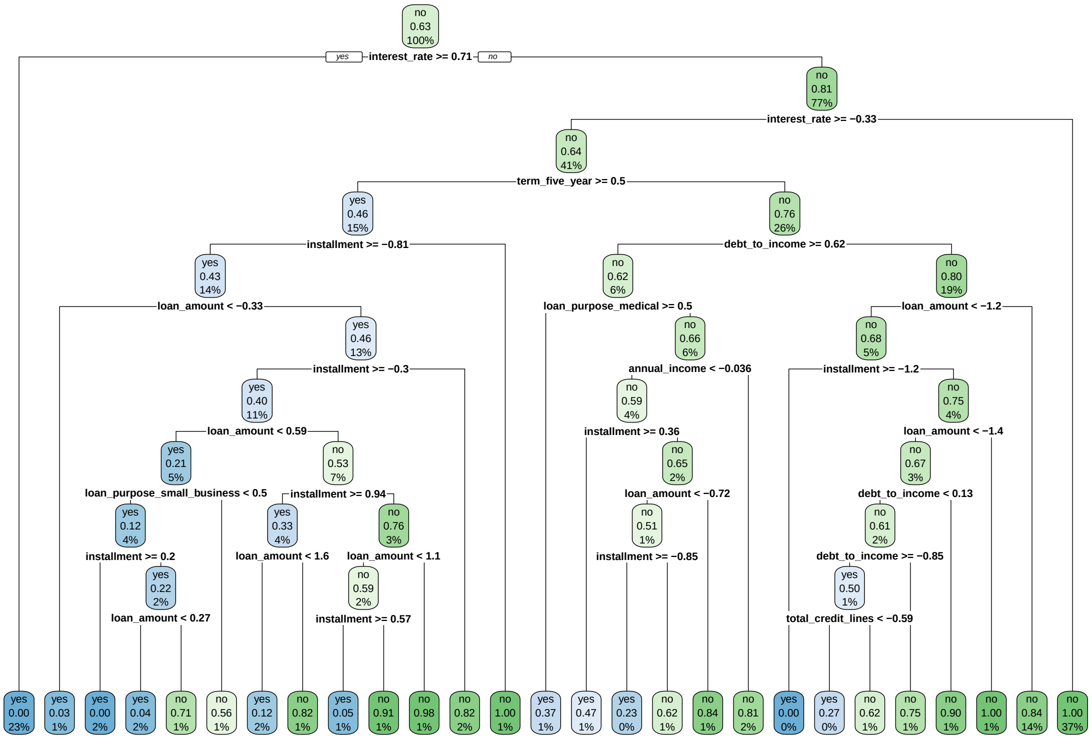
```
#Fit model
tree_wf_fit <- final_tree_wf %>%
  fit(data = loans_training)

#Trained model
tree_fit <- tree_wf_fit %>%
  pull_workflow_fit()
```

```
## Warning: `pull_workflow_fit()` was deprecated in workflows 0.2.3.
## i Please use `extract_fit_parsnip()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
#Variable Importance
vip(tree_fit)
```



```
#Decision Tree Plot
rpart.plot(tree_fit$fit, roundint = FALSE)
```



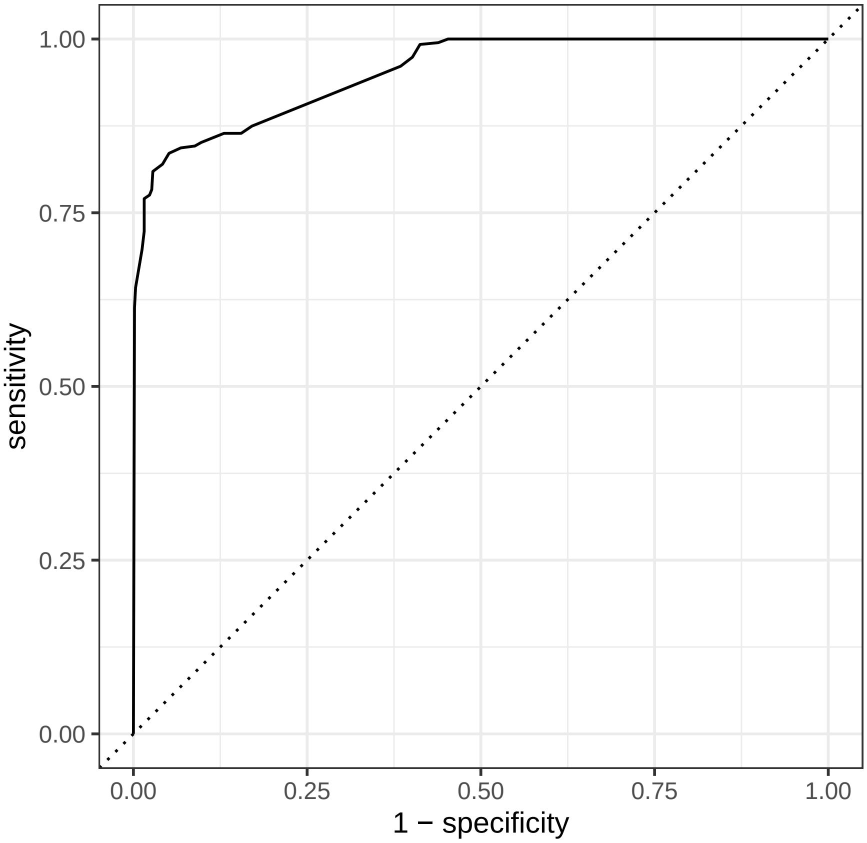
```
#Train and Evaluate with last_fit()
tree_last_fit <- final_tree_wf %>%
  last_fit(loans_split)

#Accuracy and area under the ROC curve
tree_last_fit %>% collect_metrics()
```

```
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary        0.907  Preprocessor1_Model1
## 2 roc_auc     binary        0.952  Preprocessor1_Model1
## 3 brier_class binary        0.0729 Preprocessor1_Model1
```

```
#Estimated Probabilities
tree_predictions <- tree_last_fit %>%
  collect_predictions()

tree_predictions
```

```
## # A tibble: 1,028 x 7
##    .pred_class .pred_yes .pred_no id                .row loan_default .config
##    <fct>           <dbl>    <dbl> <chr>            <int> <fct>        <chr>
##  1 yes             1       0      train/test split     1 yes          Preproces~
##  2 yes             0.952   0.0476 train/test split     8 no           Preproces~
##  3 no              0       1      train/test split     9 no           Preproces~
##  4 yes             1       0      train/test split    12 yes          Preproces~
##  5 no              0       1      train/test split    14 no           Preproces~
##  6 no              0.158   0.842  train/test split    15 yes          Preproces~
##  7 no              0.158   0.842  train/test split    20 no           Preproces~
##  8 no              0.158   0.842  train/test split    39 no           Preproces~
##  9 no               0.158   0.842  train/test split    50 no            Preproces~
## 10 yes              1       0      train/test split    53 yes           Preproces~
## # i 1,018 more rows
```

```
#ROC Curve
# ROC Curve
tree_predictions %>%
  roc_curve(truth = loan_default, .pred_yes) %>%
  autoplot()
```



```
roc_auc(tree_predictions, truth = loan_default, .pred_yes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.952
```

```
#Confusion Matrix
conf_mat(tree_predictions, truth = loan_default, estimate = .pred_class)
```

```
##           Truth
## Prediction yes  no
##        yes 314  27
##        no   69 618
```

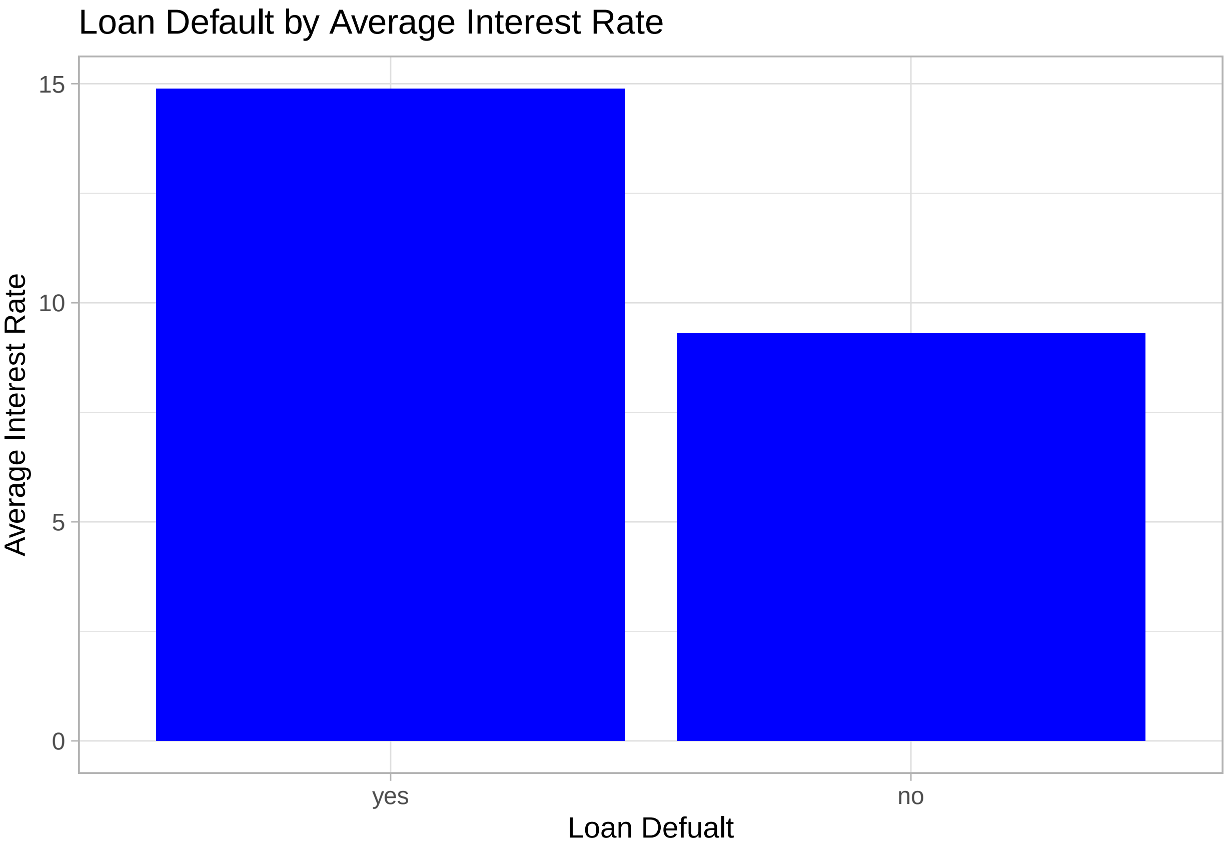
Summary of Results [50 points]

#1 The most important issue here is determining the factors that influence whether or not a person has defaulted their loan or not. Defaulting a loan means that a person has failed to meet the legal obligations of that loan, such as failing to pay it off. The goal of this analysis was to determine the factors that influenced defaulting a loan and identifying possible ways to encourage someone to not default. Some questions looked at are ones such as “Are there differences in loan default rates by interest rates?”, “Does the term impact the loan default rate?”, and “Does having a history of bankruptcy have an impact on the interest rate between those who default their loans or not?”.

#2 Some interesting findings included the average interest rate of defaulted loans, which loans had the highest average interest rate and highest rate of defaulting, and importance of each factor. These findings give a better insight into which factors could lead to more defaulted loans in the future and to plan ahead in order to try and avoid someone defaulting.

The average interest rate for defaulted loans can be seen in Question 1 of the Data Analysis section. Defaulted loans had an average interest rate of 14.89% where non-defaulted had an average rate of just 9.3%. The difference between the interest rates is 5.59%. The importance of the interest rates here exemplifies that with higher rates, a person is more likely to default their loan because as they miss payments, the higher interest proves to be too much to handle.

```r
ggplot(data = interest_rates, mapping = aes(x = loan_default, y =avg_int_rate)) +
  geom_bar(stat = 'identity', fill = "blue") +
  labs(title = "Loan Default by Average Interest Rate",
       x = "Loan Defualt",
       y = "Average Interest Rate") +
  theme_light()
```



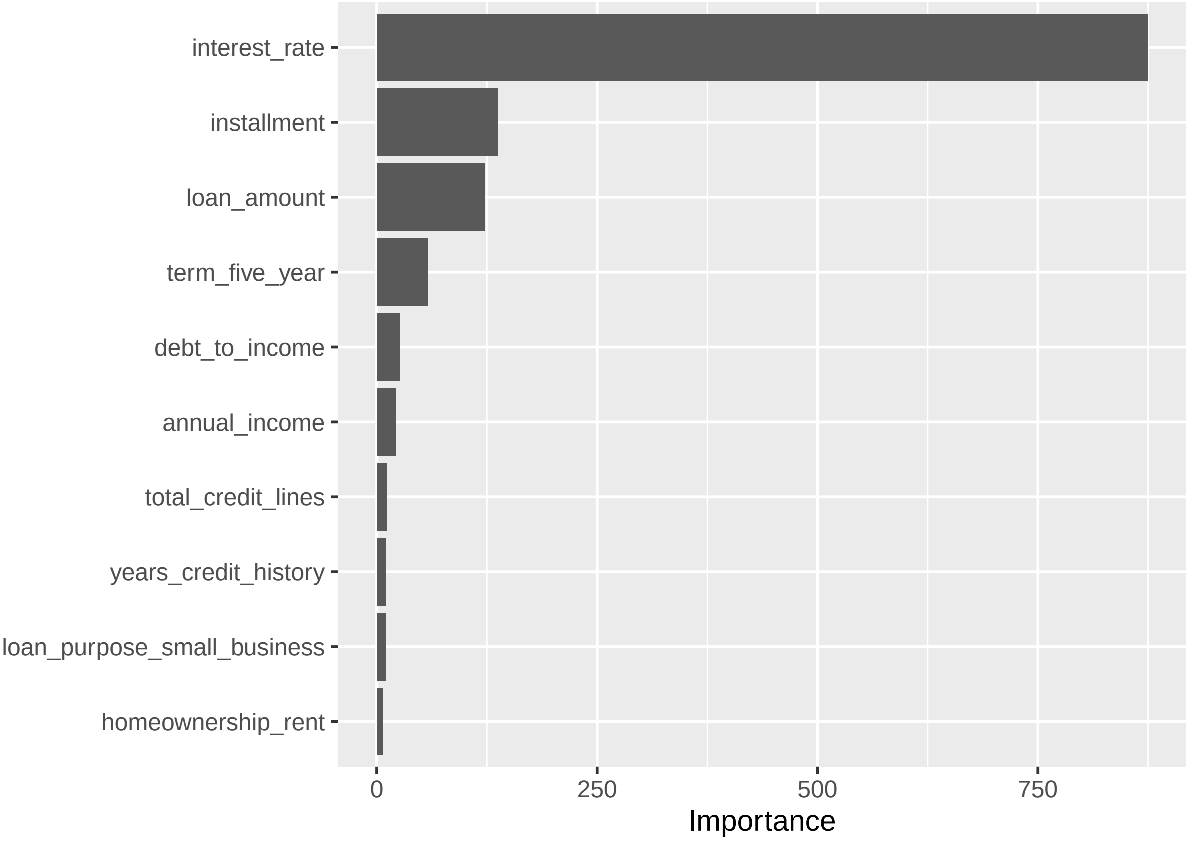
The next finding were the loans with the highest interest and default rates. These findings can be found in Question 2 of the Data Analysis. What stood out was that the two loans with the highest interest also had the highest rate of defaulted loans, despite the average loan amount for all 5 loans being within $300 of each other. Medical and credit card loans had the highest interest rate and the highest default rates. Both interest rates were over 12% and default rates were 60.47% and 53.47% respectively. Both default rates were at least 25% more than the other loans.

```
loans_df %>% group_by(loan_purpose) %>%
  summarise(avg_interest = round(mean(interest_rate), 2),
            avg_loan_amount = round(mean(loan_amount), 2),
            default_percent = 100 * round(mean(loan_default == "yes"),4))
```

```
## # A tibble: 5 x 4
##   loan_purpose       avg_interest avg_loan_amount default_percent
##   <fct>                     <dbl>           <dbl>           <dbl>
## 1 debt_consolidation         10.6          16599.            25.3
## 2 credit_card                12.4          16656.            53.5
## 3 medical                    12.8          16891.            60.5
## 4 small_business             10.7          16695.            25.9
## 5 home_improvement           10.9          16729.            28
```

The final finding to go over is the importance of each factor from the Decision Tree analysis. Here we can see that the interest rate was clearly the most important factor when compared to the others.

```
vip(tree_fit)
```



#3 The model with the best analysis was the Linear Regression model. Despite the KNN and Decision Tree offering some new and additional insights, the Linear Regression model has an ROC-AUC of 0.9879. This is the highest AUC scoring of the three models, and means that it is the most accurate. Having an AUC score as close to 1 means that it is the most accurate and reliable, with scores closer to 0 being less accurate and reliable. The Linear Regression AUC score of 0.9879 is essentially a 98.79% accurate representation and predictability when using this model. Just because the Linear Regression yielded the highest score does not mean that the other two were not accurate. The KNN had an AUC score of 0.9 and the Decision Tree yielded 0.9516.

#4 My recommendation to reduce the amount of defaulted loans is to level out the interest rates to make them more even across the board. It was seen in Question 2 that the average loan amount was not an issue as each amount was within a range of $300 from lowest to highest average price. The distinguishing factor there was the average interest rate. Despite the average interest rates for the two most defaulted loans were only 2% higher than the other, the variable importance chart from the Decision Tree analysis backs this statement up. It would appear that the higher the interest rate is, the bigger the chance of someone defaulting that loan. Leveling out the interest rates will retain those who are taking out loans while still making profit off of the interest. Even though this would mean less interest profit, it would be worth it compared to losing a whole loan from being defaulted.

#5 The overall findings of this data analysis and report support the decision that interest rates are the leading factor in a loan being defaulted or not. Between the tibble and the variable importance findings, it was clear to see why that decision was reached based on how much of an influence it had on loans compared to other reasons such as the loan amount and installments. Despite the other factors having some disparities as well, none came close to the level of influence the interest rate had on defaulted loans. All three Predictive Models returned an AUC greater than or equal to 0.9, which shows that the findings were accurate and reliable. Reducing interest rates appears to be the best way to reduce the amount of loans being defaulted based on its level of importance and influence.

— End of the Project —