# Data Mining Final Project

**Your Name**: Anasa Alamgir **Your G Number**: G01300460

## Introduction

This project aims to explore the factors that lead to loan default and use machine learning models to predict the chance of an applicant defaulting on their loan in the future. The Bank has experienced customers defaulting on their loans and therefore seeks to predict whether an applicant will default on their loan in order to protect the Bank from large financial losses.

Exploratory data analysis can help to find the relationship between whether the applicant defaults and the various factors that affect them defaulting on the loan.

## Summary of Results

The default response variable in this data frame is loan_default, which records whether an applicant will default or not. This variable has also been coded with 'Yes' and 'No' factors. Therefore using visualization techniques this report will showwhich other factors can explain why some applicants default and others do not.

Findings…

## Data Analysis [30 Points]

In this section, you must think of at least 6 relevant questions that explore the relationship between `loan_default` and the other variables in the `loan_df` data set. The goal of your analysis should be discovering which variables drive the differences between customers who do and do not default on their loans.

You must answer each question and provide supporting data summaries with either a summary data frame (using `dplyr/tidyr`) or a plot (using `ggplot`) or both.

In total, you must have a minimum of 3 plots (created with `ggplot`) and 3 summary data frames (created with `dplyr`) for the exploratory data analysis section. Among the plots you produce, you must have at least 3 different types (ex. box plot, bar chart, histogram, scatter plot, etc…)

See the example question below.

### Sample Question

#### Are there differences in loan default rates by loan purpose?

**Answer**: Yes, the data indicates that credit card and medical loans have significantly larger default rates than any other type of loan. In fact, both of these loan types have default rates at more than 50%. This is nearly two times the average default rate for all other loan types.

#### Summary Table

```r
loans_df %>%
  group_by(loan_purpose) %>%
  summarise(n_customers = n(),
            customers_default = sum(loan_default == 'yes'),
            default_percent = 100 * mean(loan_default == 'yes'))
```

```
## # A tibble: 5 x 4
##   loan_purpose       n_customers customers_default default_percent
##   <fct>                    <int>             <int>           <dbl>
## 1 debt_consolidation        1218               308            25.3
## 2 credit_card                879               470            53.5
## 3 medical                    635               384            60.5
## 4 small_business             853               221            25.9
## 5 home_improvement           525               147            28
```

### Data Visulatization

```r
default_rates <- loans_df %>%
                 group_by(loan_purpose) %>%
                 summarise(n_customers = n(),
                 customers_default = sum(loan_default == 'yes'),
                 default_percent = 100 * mean(loan_default == 'yes'))


ggplot(data = default_rates, mapping = aes(x = loan_purpose, y = default_percent)) +
    geom_bar(stat = 'identity', fill = '#006EA1', color = 'white') +
    labs(title = 'Loan Default Rate by Purpose of Loan',
         x = 'Loan Purpose',
         y = 'Default Percentage') +
    theme_light()
```

```r
head(loans_df)
```

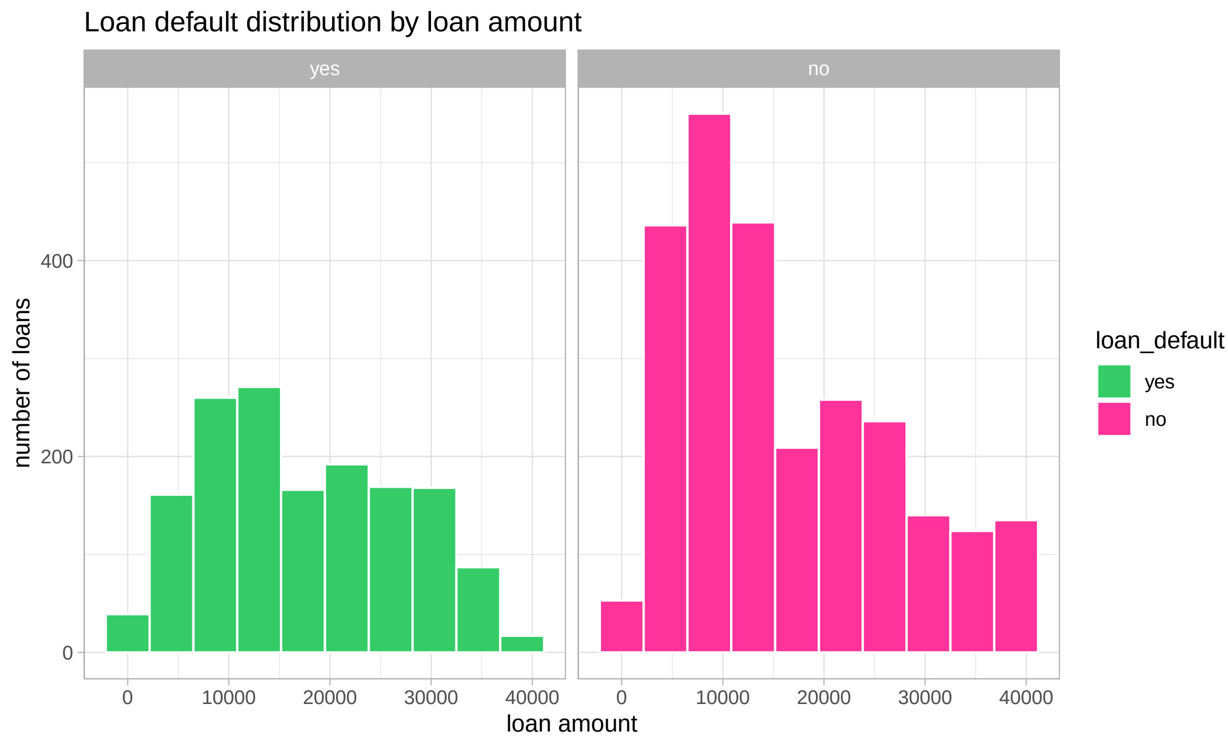
```
## # A tibble: 6 x 16
##   loan_default loan_amount installment interest_rate loan_purpose
##   <fct>              <int>       <dbl>         <dbl> <fct>
## 1 yes                35000        927.         17.2  small_business
## 2 yes                10000        260.         11.5  small_business
## 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
## # ... with 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>
```

## Question 1

**Question**: Is there a relationship between loan default and loan amount?

**Answer**:

```r
loans_df %>% ggplot(
  aes(x = loan_amount, fill = loan_default)
) +
  geom_histogram(bins=10, color="white") +
  scale_fill_manual(values = c("#33CC66","#FF3399")) +
  theme_light() +
  facet_wrap(~ loan_default)+
  labs(
    title = "Loan default distribution by loan amount",
    x = "loan amount",
    y = "number of loans"
  )
```




## Question 2

**Question**: Is there a relationship between loan default and history of missed payments in the past 2 years based on term?

**Answer**:

```
loans_df %>%
  group_by(term, loan_default) %>%
  summarise(
    num_loans = n()
  )
```

```
## # A tibble: 4 x 3
## # Groups:   term [2]
##   term       loan_default num_loans
##   <fct>      <fct>            <int>
## 1 three_year yes                693
## 2 three_year no                1895
## 3 five_year  yes                837
## 4 five_year  no                 685
```

## Question 3

**Question**: Is there a relationship between loan default and annual income?

**Answer**:

```
loans_df %>% group_by(loan_default) %>%
  summarise(
    avg_income = mean(annual_income),
    min_income = min(annual_income),
```

```
  sd_income = sd(annual_income)
)
```

```
## # A tibble: 2 x 4
##   loan_default avg_income min_income sd_income
##   <fct>             <dbl>      <dbl>     <dbl>
## 1 yes              67819.       7500    34930.
## 2 no               76096.       3000    38161.
```

## Question 4

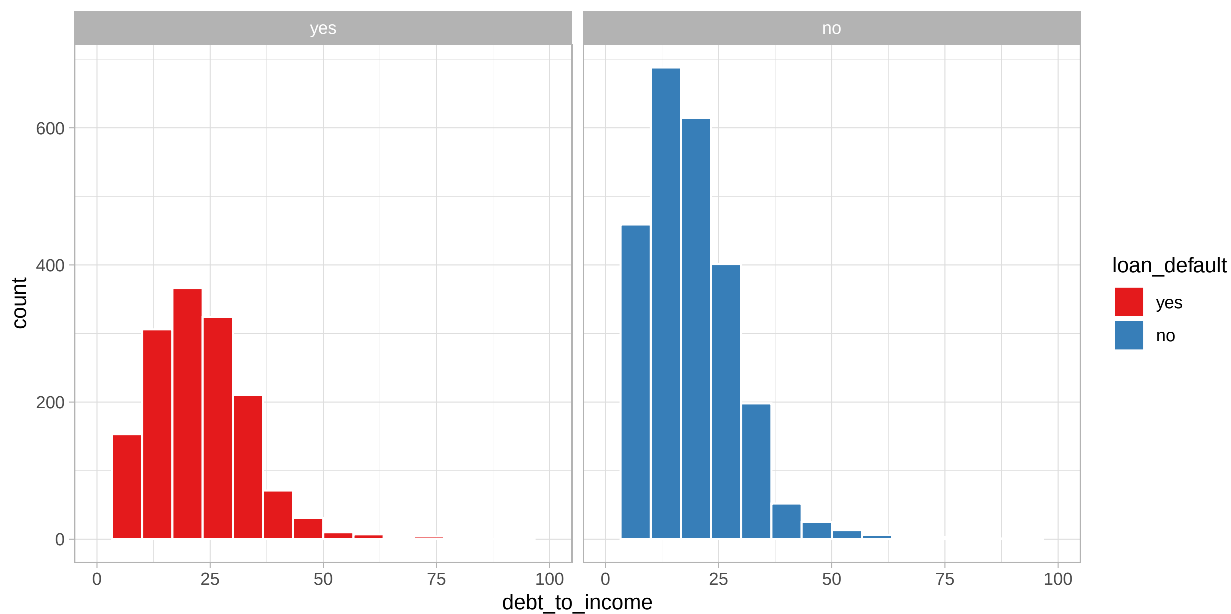
**Question**: Is there a relationship between loan default and debt-to-income ratio?

**Answer**:

```
loans_df %>% ggplot(
  aes(x = debt_to_income, fill = loan_default)
) + geom_histogram(bins=16, color = "white") + xlim(0,100)+
  theme_light() +
scale_fill_brewer(palette="Set1") +
  facet_wrap(~ loan_default)
```

```
## Warning: Removed 9 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```



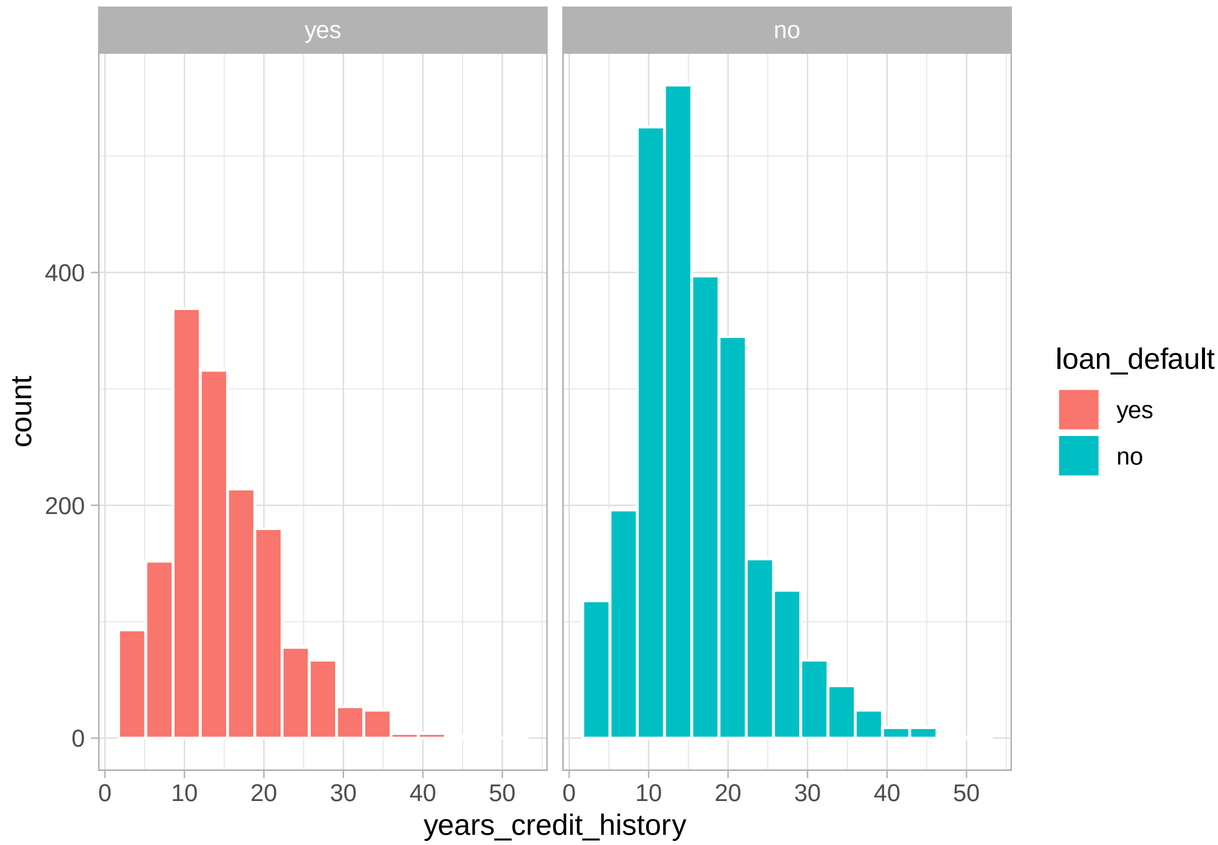

## Question 5

**Question**: Is there a relationship between loan default and years of credit history?

**Answer**:

```
loans_df %>% ggplot(
  aes(x = years_credit_history, fill = loan_default)
) + theme_light() +
```

```
  geom_histogram(bins=15, color="white") +
  facet_wrap(~loan_default)
```




### Question 6

**Question**: Is there a relationship between loan default rate and interest rates and loan purpose?

**Answer**:

```
loans_df %>% group_by(loan_default, loan_purpose) %>%
  summarise(
    avg_interestrate = mean(interest_rate),
    max_interestrate = max(interest_rate)
  )
```

```
## # A tibble: 10 x 4
## # Groups:   loan_default [2]
##    loan_default loan_purpose       avg_interestrate max_interestrate
##    <fct>        <fct>                         <dbl>            <dbl>
##  1 yes          debt_consolidation             14.8               20
##  2 yes          credit_card                    15.0               20
##  3 yes          medical                        15.0               20
##  4 yes          small_business                 14.8               20
##  5 yes          home_improvement               14.8               20
##  6 no           debt_consolidation              9.20             14.0
##  7 no           credit_card                     9.36             14.0
##  8 no           medical                         9.60             14.0
```

```
##   9 no          small_business                 9.28            14.0
##  10 no          home_improvement               9.34            14.0
```

## Predictive Modeling [70 Points]

In this section of the project, you will fit **three classification algorithms** to predict the response variable,`loan_default`. You should use all of the other variables in the `loans_df` data as predictor variables for each model.

You must follow the machine learning steps below.

The data splitting and feature engineering steps should only be done once so that your models are using the same data and feature engineering steps for training.

- Split the `loans_df` data into a training and test set (remember to set your seed)
- Specify a feature engineering pipeline with the `recipes` package
  - You can include steps such as skewness transformation, dummy variable encoding or any other steps you find appropriate
- Specify a `parsnip` model object
  - You may choose from the following classification algorithms:
    * Logistic Regression
    * LDA
    * QDA
    * KNN
    * Decision Tree
    * Random Forest
- Package your recipe and model into a workflow
- Fit your workflow to the training data
  - If your model has hyperparameters:
    * Split the training data into 5 folds for 5-fold cross validation using `vfold_cv` (remember to set your seed)
    * Perform hyperparamter tuning with a random grid search using the `grid_random()` function
    * Hyperparameter tuning can take a significant amount of computing time. Be careful not to set the `size` argument of `grid_random()` too large. I recommend `size` = 10 or smaller.
    * Select the best model with `select_best()` and finalize your workflow
- Evaluate model performance on the test set by plotting an ROC curve using `autoplot()` and calculating the area under the ROC curve on your test data

### Model 1 Logistic Regression

```r
#split loans_df into training and test sets
set.seed(172)
loan_split <- initial_split(loans_df, prop = 0.75,
                            strata = loan_default)

loan_training <- loan_split %>% training()

loan_test <- loan_split %>% testing()
```

```r
#cross validation folds for hyperparameter tuning
set.seed(172)
loan_folds <- vfold_cv(loan_training, v = 6)
```

```r
#feature engineering
loan_recipe <- recipe(loan_default ~ .,data = loan_training) %>%
```

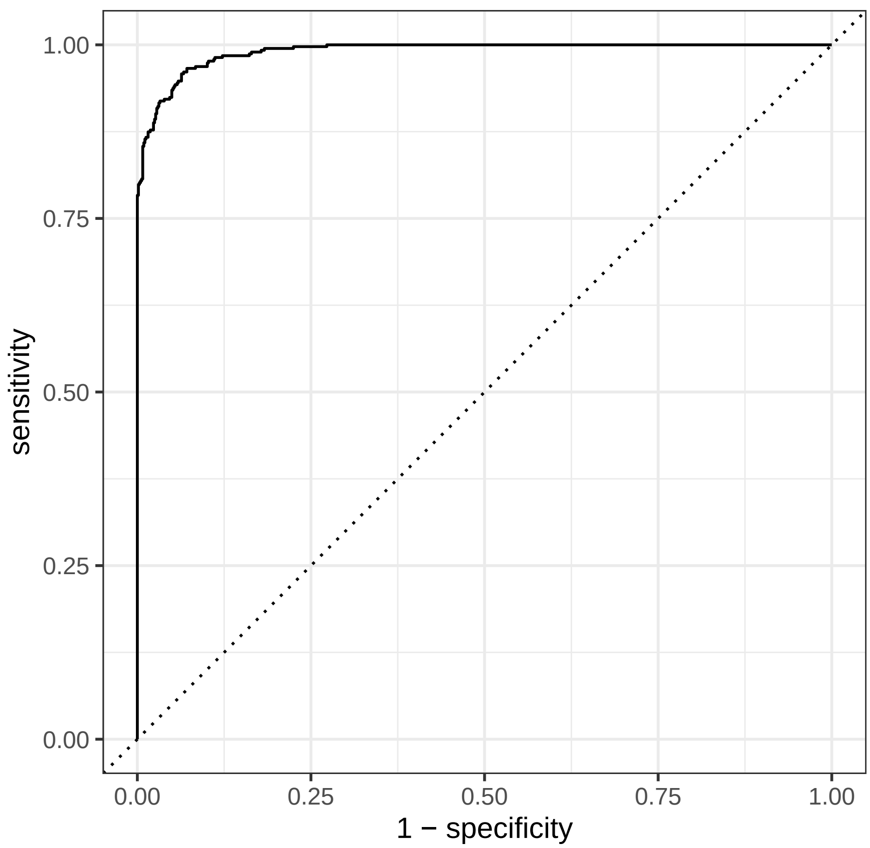
```r
  step_YeoJohnson(all_numeric(), -all_outcomes()) %>%
  step_normalize(all_numeric(), -all_outcomes()) %>%
  step_dummy(all_nominal(), -all_outcomes())

loan_recipe %>%
  prep(training = loan_training) %>%
  bake(new_data = NULL)
```

```
## # A tibble: 3,082 x 20
##    loan_amount installment interest_rate annual_income current_job_years
##          <dbl>       <dbl>         <dbl>         <dbl>             <dbl>
##  1     -1.21       -1.20         -0.849         0.117             -0.397
##  2      0.0358      0.0759        0.0344       -0.827              1.10
##  3      1.65        1.01         -1.08          2.22               1.10
##  4     -0.347      -0.821        -0.924        -1.17              -0.121
##  5     -0.531      -0.527        -1.08         -0.179              1.10
##  6     -1.04       -1.03         -0.158        -0.445             -0.691
##  7     -0.531      -0.498        -0.704         0.503              1.10
##  8     -2.20       -2.38          0.578        -0.0269            -0.691
##  9     -1.40       -1.44          0.692         0.117             -0.121
## 10      1.89        2.16          0.0971        2.16               0.873
## # ... with 3,072 more rows, and 15 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>,
## #   missed_payment_2_yr_no <dbl>, history_bankruptcy_no <dbl>, ...
```

```r
#specify model
loan_logistic <- logistic_reg() %>%
  set_engine('glm') %>%
  set_mode('classification')
```

```r
#create workflow
logistic_wf <- workflow() %>%
  add_model(loan_logistic) %>%
  add_recipe(loan_recipe)

#roc curve and auc
logistic_fit <- logistic_wf %>% last_fit(split = loan_split)

# collect predictions
logistic_predictions <- logistic_fit %>% collect_predictions()
```

```r
#roc curve and auc
roc_curve(logistic_predictions,
          truth = loan_default,
          estimate = .pred_yes) %>% autoplot()
```




```r
roc_auc(logistic_predictions, truth = loan_default, .pred_yes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.990
```

```r
#confusion matrix
conf_mat(logistic_predictions,
         truth = loan_default,
         estimate = .pred_class)
```

```
##           Truth
## Prediction yes  no
##        yes 352  23
##        no   31 622
```

```r
#model summary
log_model <- glm(loan_default ~., data = loan_training, family = binomial())
tidy(log_model)
```

```
## # A tibble: 20 x 5
##    term                          estimate  std.error statistic  p.value
##    <chr>                            <dbl>      <dbl>     <dbl>    <dbl>
##  1 (Intercept)                 10.4        0.885         11.7  1.23e-31
##  2 loan_amount                  0.00108    0.0000620     17.4  4.76e-68
##  3 installment                 -0.0342     0.00193      -17.7  1.87e-70
##  4 interest_rate               -0.698      0.0361       -19.3  2.04e-83
```

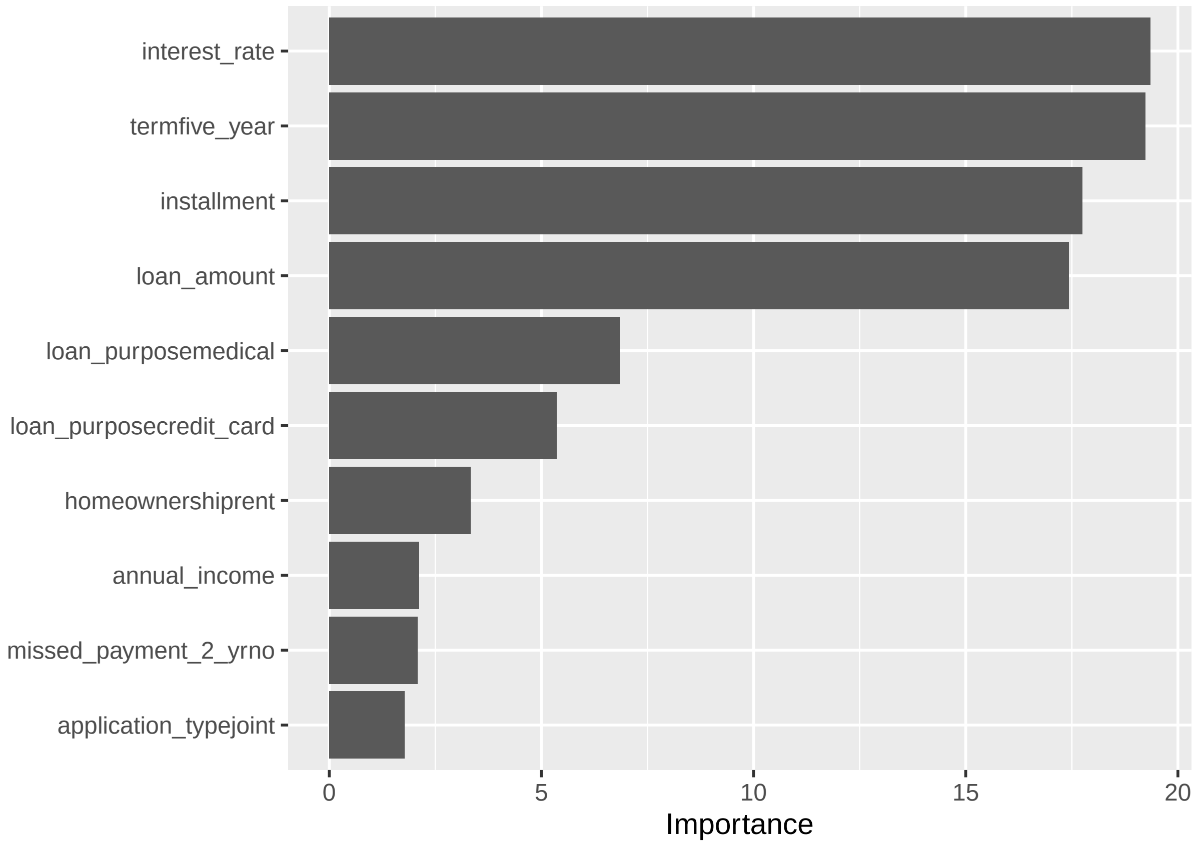
```
## 5 loan_purposecredit_card      -1.12       0.210         -5.36  8.33e- 8
## 6 loan_purposemedical          -1.58       0.231         -6.84  7.80e-12
## 7 loan_purposesmall_business   -0.0414     0.219         -0.189 8.50e- 1
## 8 loan_purposehome_improvement  0.0288     0.248          0.116 9.08e- 1
## 9 application_typejoint        -0.392      0.222         -1.77  7.67e- 2
## 10 termfive_year               -7.32       0.381        -19.2   2.00e-82
## 11 homeownershiprent           -0.560      0.168         -3.32  8.84e- 4
## 12 homeownershipown            -0.282      0.239         -1.18  2.38e- 1
## 13 annual_income                0.00000528 0.00000250     2.11  3.48e- 2
## 14 current_job_years            0.00343    0.0211         0.162 8.71e- 1
## 15 debt_to_income              -0.00727    0.00453       -1.61  1.08e- 1
## 16 total_credit_lines           0.00624    0.00689        0.906 3.65e- 1
## 17 years_credit_history         0.0178     0.0115         1.55  1.22e- 1
## 18 missed_payment_2_yrno        0.457      0.219          2.09  3.70e- 2
## 19 history_bankruptcyno        -0.0973     0.223         -0.436 6.63e- 1
## 20 history_tax_liensno          0.116      0.631          0.183 8.54e- 1
```

```
summary(log_model)
```

```
##
## Call:
## glm(formula = loan_default ~ ., family = binomial(), data = loan_training)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.08458  -0.11368   0.05583   0.27167   3.02825
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  1.035e+01  8.848e-01  11.703  < 2e-16 ***
## loan_amount                  1.081e-03  6.201e-05  17.431  < 2e-16 ***
## installment                 -3.420e-02  1.927e-03 -17.746  < 2e-16 ***
## interest_rate               -6.976e-01  3.605e-02 -19.350  < 2e-16 ***
## loan_purposecredit_card     -1.124e+00  2.097e-01  -5.360 8.33e-08 ***
## loan_purposemedical         -1.578e+00  2.306e-01  -6.842 7.80e-12 ***
## loan_purposesmall_business  -4.135e-02  2.189e-01  -0.189 0.850197
## loan_purposehome_improvement  2.883e-02  2.483e-01   0.116 0.907570
## application_typejoint       -3.924e-01  2.217e-01  -1.770 0.076731 .
## termfive_year               -7.319e+00  3.805e-01 -19.232  < 2e-16 ***
## homeownershiprent           -5.600e-01  1.684e-01  -3.325 0.000884 ***
## homeownershipown            -2.819e-01  2.390e-01  -1.179 0.238259
## annual_income                5.277e-06  2.500e-06   2.110 0.034833 *
## current_job_years            3.427e-03  2.109e-02   0.162 0.870928
## debt_to_income              -7.275e-03  4.529e-03  -1.606 0.108236
## total_credit_lines           6.242e-03  6.891e-03   0.906 0.365020
## years_credit_history         1.778e-02  1.149e-02   1.547 0.121883
## missed_payment_2_yrno        4.569e-01  2.191e-01   2.086 0.037008 *
## history_bankruptcyno        -9.730e-02  2.230e-01  -0.436 0.662595
## history_tax_liensno          1.157e-01  6.310e-01   0.183 0.854456
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4068.8  on 3081  degrees of freedom
```

```
## Residual deviance: 1255.8  on 3062  degrees of freedom
## AIC: 1295.8
##
## Number of Fisher Scoring iterations: 7
```

```r
vip(log_model)
```




### Model 2 LDA

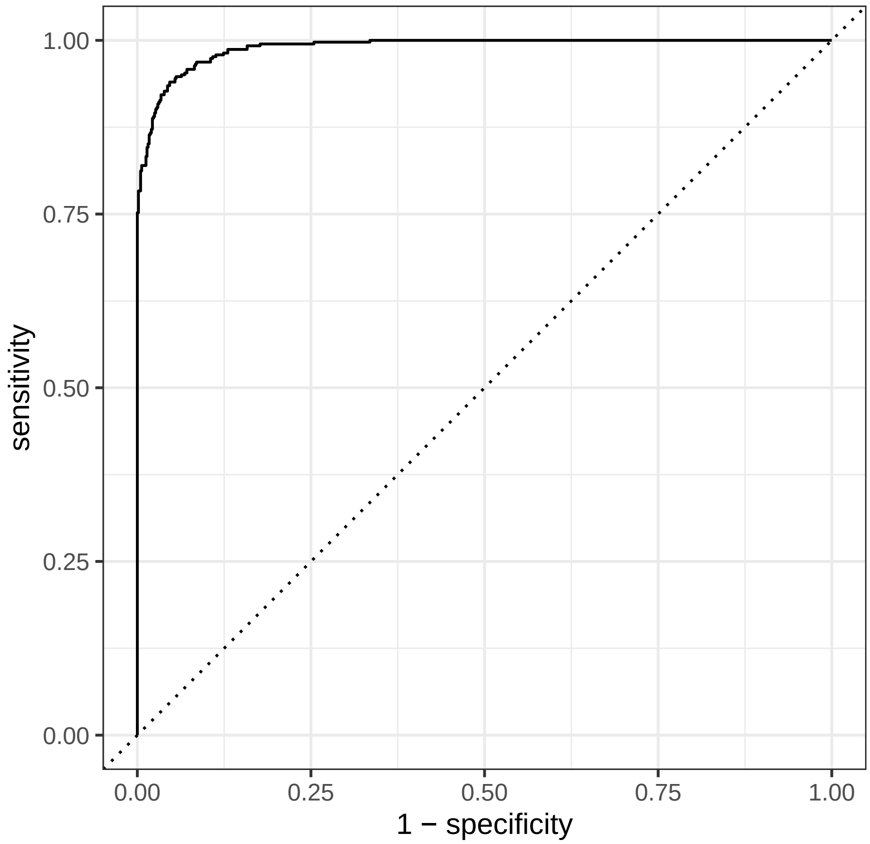
```r
#specify model
loan_lda <- discrim_regularized(frac_common_cov = 1) %>%
  set_engine("klaR") %>%
  set_mode("classification")
```

```r
#workflow
lda_wf <- workflow() %>% add_model(loan_lda) %>% add_recipe(loan_recipe)
```

```r
#fit workflow
lda_fit <- lda_wf %>% last_fit(split=loan_split)
```

```r
#collect predictions
lda_predictions <- lda_fit %>% collect_predictions()
```

```r
#roc curve
roc_curve(lda_predictions, truth=loan_default, estimate= .pred_yes) %>%
  autoplot()
```




```
#auc
roc_auc(lda_predictions, truth=loan_default, .pred_yes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.990
```

```
#confusion matrix
conf_mat(lda_predictions, truth = loan_default, estimate = .pred_class)
```

```
##           Truth
## Prediction yes  no
##        yes 348  19
##        no   35 626
```

### Model 3 K-Nearest Neighbor

```
#specify model
knn_model <- nearest_neighbor(neighbors = tune()) %>%
  set_engine("kknn") %>%
  set_mode("classification")
```

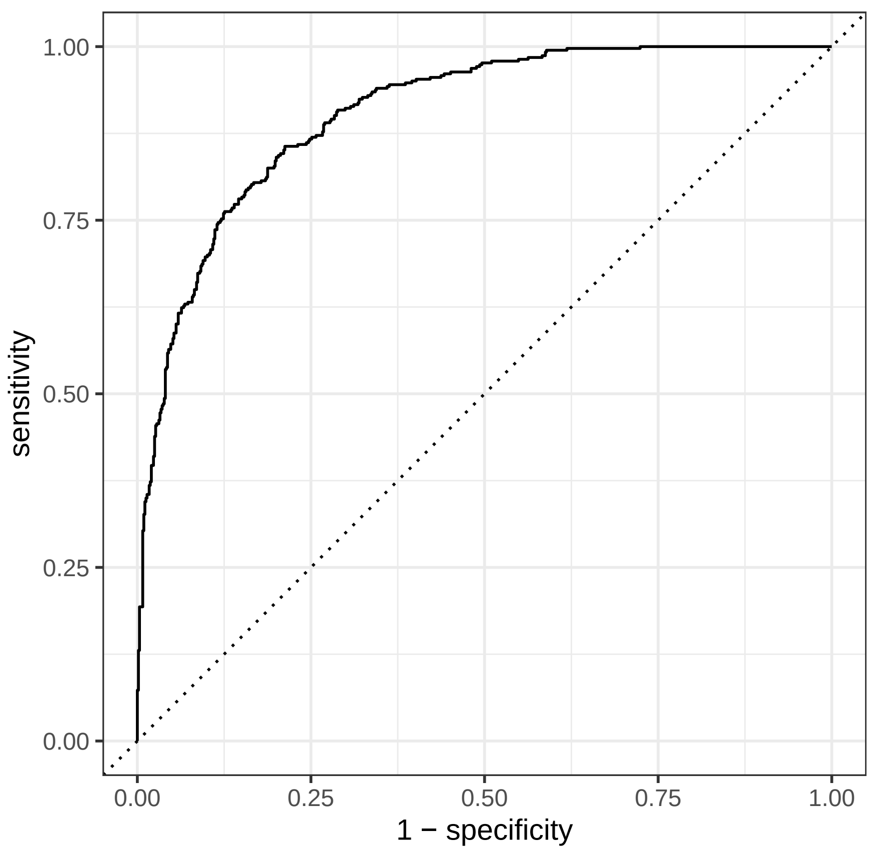
```
#workflow
knn_wf <- workflow() %>%
  add_model(knn_model) %>%
  add_recipe(loan_recipe)
```

```r
#create grid for hyperparameter testing
k_grid <- tibble(neighbors = c(10,20,30,40,50,75,100,125,150))
```

```r
#tuning wf
set.seed(271)
knn_tuning <- knn_wf %>% tune_grid(resamples=loan_folds, grid=k_grid)

#select best model from tuning result
best_k <- knn_tuning %>% select_best(metric='roc_auc')

#add optimal model to wf
final_knn_wf <- knn_wf %>% finalize_workflow(best_k)

#fit model
knn_fit <- final_knn_wf %>% last_fit(split=loan_split)
```

```r
#get df of test prediction results
knn_predictions <- knn_fit %>% collect_predictions()
```

```r
#roc curve and roc auc and confusion matrix for predictions
roc_curve(knn_predictions, truth = loan_default, estimate= .pred_yes) %>%
  autoplot()
```




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

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
```

```
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.904
```

```r
conf_mat(knn_predictions, truth= loan_default, estimate = .pred_class)
```

```
##           Truth
## Prediction yes  no
##        yes 213  28
##        no  170 617
```

— End of the Project —