Classification_Assessment

Yubo Xu

2024-03-13

```
if (!requireNamespace("caret", quietly = TRUE)) install.packages("caret")
if (!requireNamespace("e1071", quietly = TRUE)) install.packages("e1071")
if (!requireNamespace("randomForest", quietly = TRUE)) install.packages("randomForest")
if (!requireNamespace("pROC", quietly = TRUE)) install.packages("pROC")
if (!requireNamespace("skimr", quietly = TRUE)) install.packages("skimr")
if (!requireNamespace("corrplot", quietly = TRUE)) install.packages("corrplot")
if (!requireNamespace("dplyr", quietly = TRUE)) install.packages("dplyr")
if (!requireNamespace("GGally", quietly = TRUE)) install.packages("GGally")
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library("caret")
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library("e1071")
library("randomForest")
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library("pROC")
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library("corrplot")
```

```
## corrplot 0.92 loaded
```

```
library("skimr")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:randomForest':
##
##     combine
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("GGally")
```

```
# Load raw data from URL
data <- readr::read_csv("https://www.louisaslett.com/Courses/MISCADA/bank_personal_loan.csv")
```

```
## Rows: 5000 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (13): Age, Experience, Income, ZIP.Code, Family, CCAvg, Education, Mortg...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str(data)
```

```
## spc_tbl_ [5,000 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Age               : num [1:5000] 25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : num [1:5000] 1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : num [1:5000] 49 34 11 100 45 29 72 22 81 180 ...
##  $ ZIP.Code          : num [1:5000] 91107 90089 94720 94112 91330 ...
##  $ Family            : num [1:5000] 4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num [1:5000] 1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Education         : num [1:5000] 1 1 1 2 2 2 2 3 2 3 ...
##  $ Mortgage          : num [1:5000] 0 0 0 0 0 155 0 0 104 0 ...
##  $ Personal.Loan     : num [1:5000] 0 0 0 0 0 0 0 0 0 1 ...
##  $ Securities.Account: num [1:5000] 1 1 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : num [1:5000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : num [1:5000] 0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : num [1:5000] 0 0 0 0 1 0 0 1 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Age = col_double(),
##   ..   Experience = col_double(),
##   ..   Income = col_double(),
##   ..   ZIP.Code = col_double(),
##   ..   Family = col_double(),
##   ..   CCAvg = col_double(),
##   ..   Education = col_double(),
##   ..   Mortgage = col_double(),
##   ..   Personal.Loan = col_double(),
##   ..   Securities.Account = col_double(),
##   ..   CD.Account = col_double(),
##   ..   Online = col_double(),
##   ..   CreditCard = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
skim(data)
```

Data summary

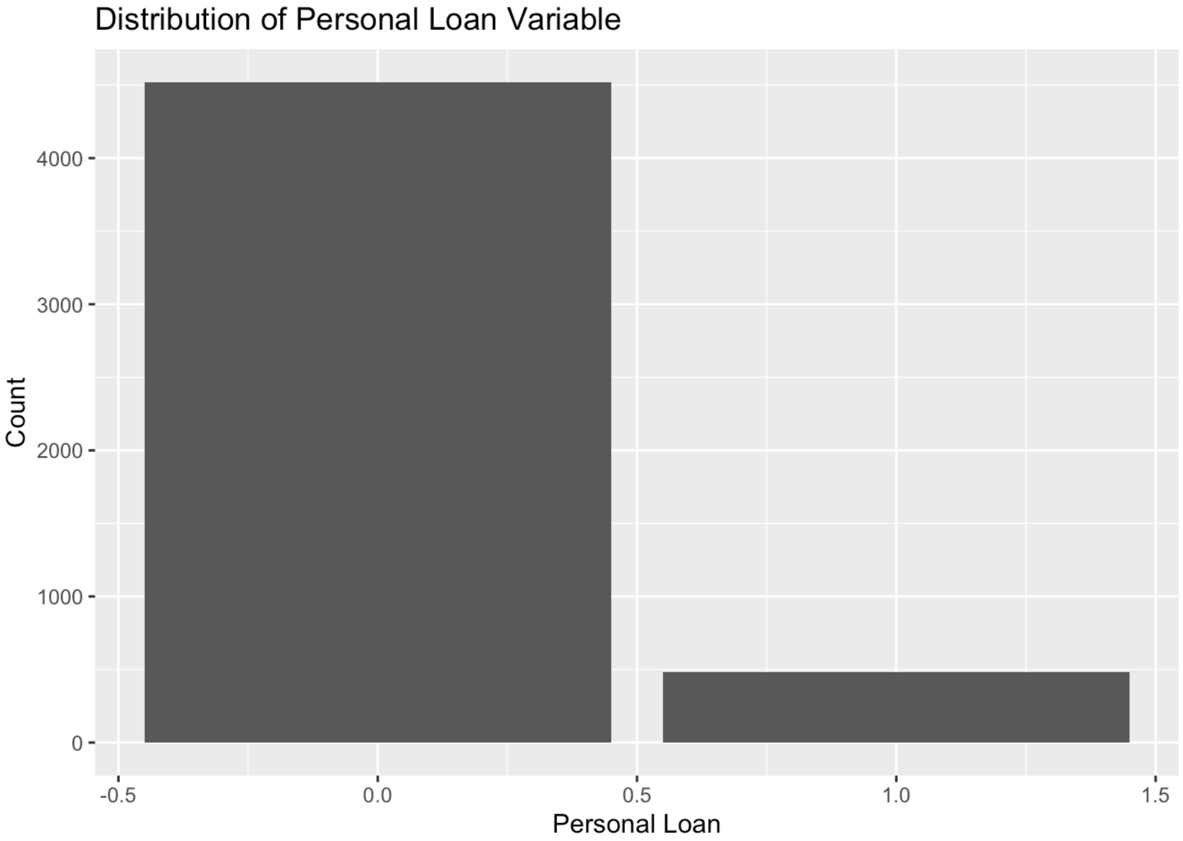
Name	data
Number of rows	5000
Number of columns	13
_______________________	
Column type frequency:	
numeric	13
________________________	
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Age	0	1	45.34	11.46	23	35.0	45.0	55.0	67	▅▇▇▇▆
Experience	0	1	20.10	11.47	-3	10.0	20.0	30.0	43	▅▇▇▇▅
Income	0	1	73.77	46.03	8	39.0	64.0	98.0	224	▇▇▃▂▁
ZIP.Code	0	1	93152.50	2121.85	9307	91911.0	93437.0	94608.0	96651	▁▁▁▁▇

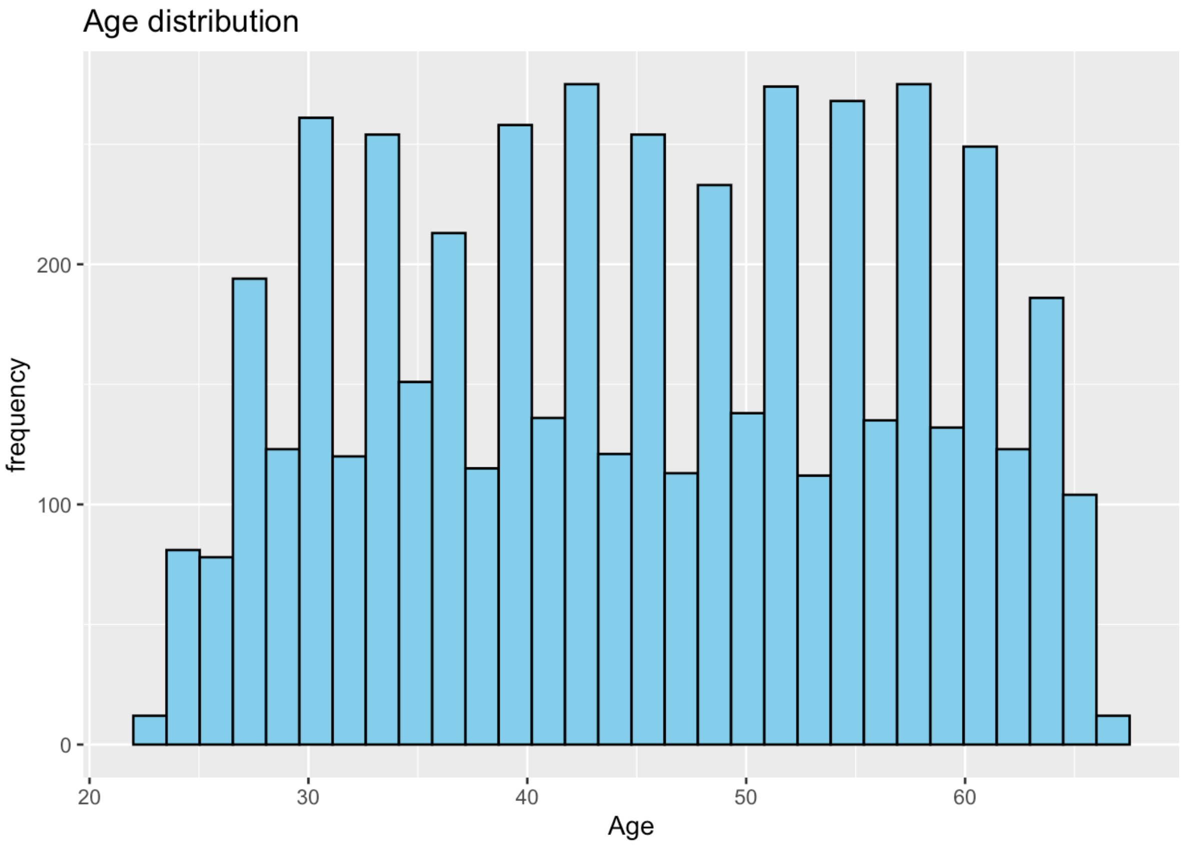
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Family	0	1	2.40	1.15	1	1.0	2.0	3.0	4	▇▇▁▆▆
CCAvg	0	1	1.94	1.75	0	0.7	1.5	2.5	10	▇▂▁▁▁
Education	0	1	1.88	0.84	1	1.0	2.0	3.0	3	▇▁▆▁▆
Mortgage	0	1	56.50	101.71	0	0.0	0.0	101.0	635	▇▂▁▁▁
Personal.Loan	0	1	0.10	0.29	0	0.0	0.0	0.0	1	▇▁▁▁▁
Securities.Account	0	1	0.10	0.31	0	0.0	0.0	0.0	1	▇▁▁▁▁
CD.Account	0	1	0.06	0.24	0	0.0	0.0	0.0	1	▇▁▁▁▁
Online	0	1	0.60	0.49	0	0.0	1.0	1.0	1	▆▁▁▁▇
CreditCard	0	1	0.29	0.46	0	0.0	0.0	1.0	1	▇▁▁▁▃

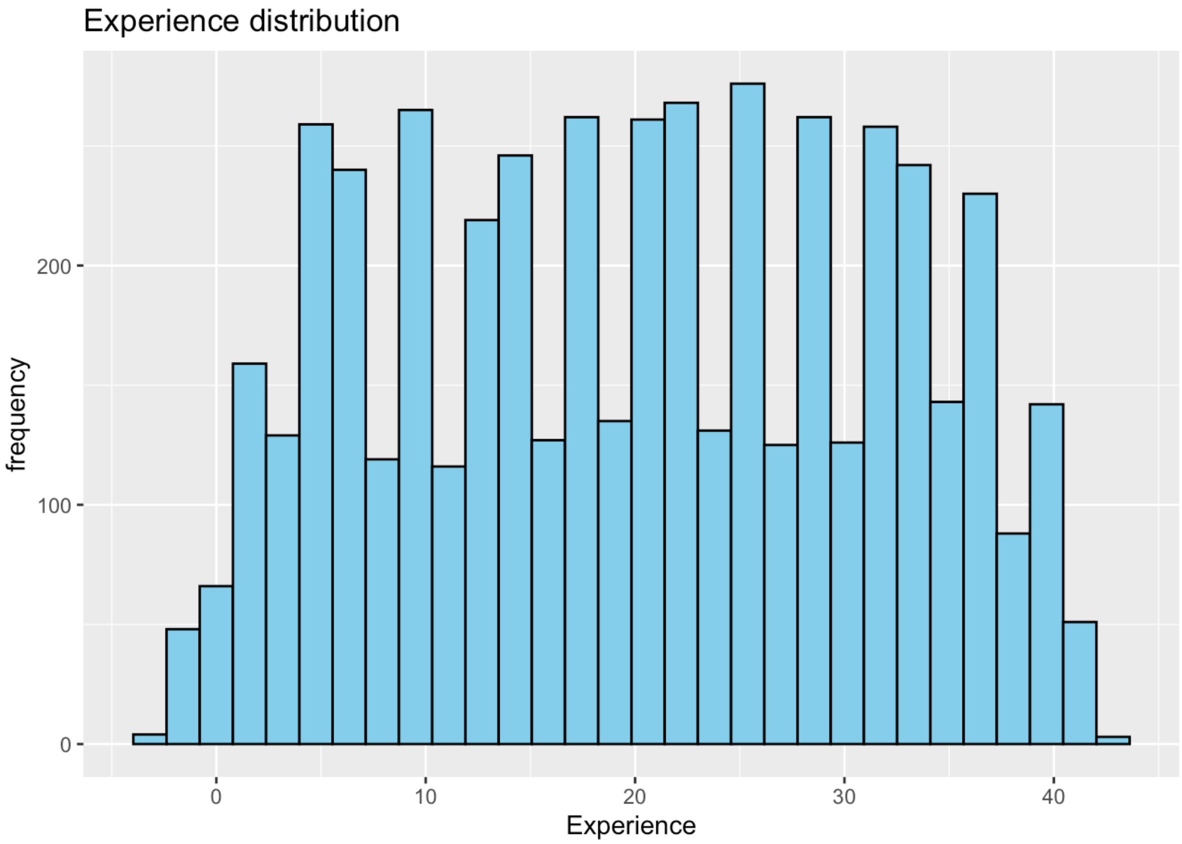
```
ggplot(data, aes(x = Personal.Loan)) +
  geom_bar() +
  labs(title = "Distribution of Personal Loan Variable", x = "Personal Loan", y = "Count")
```

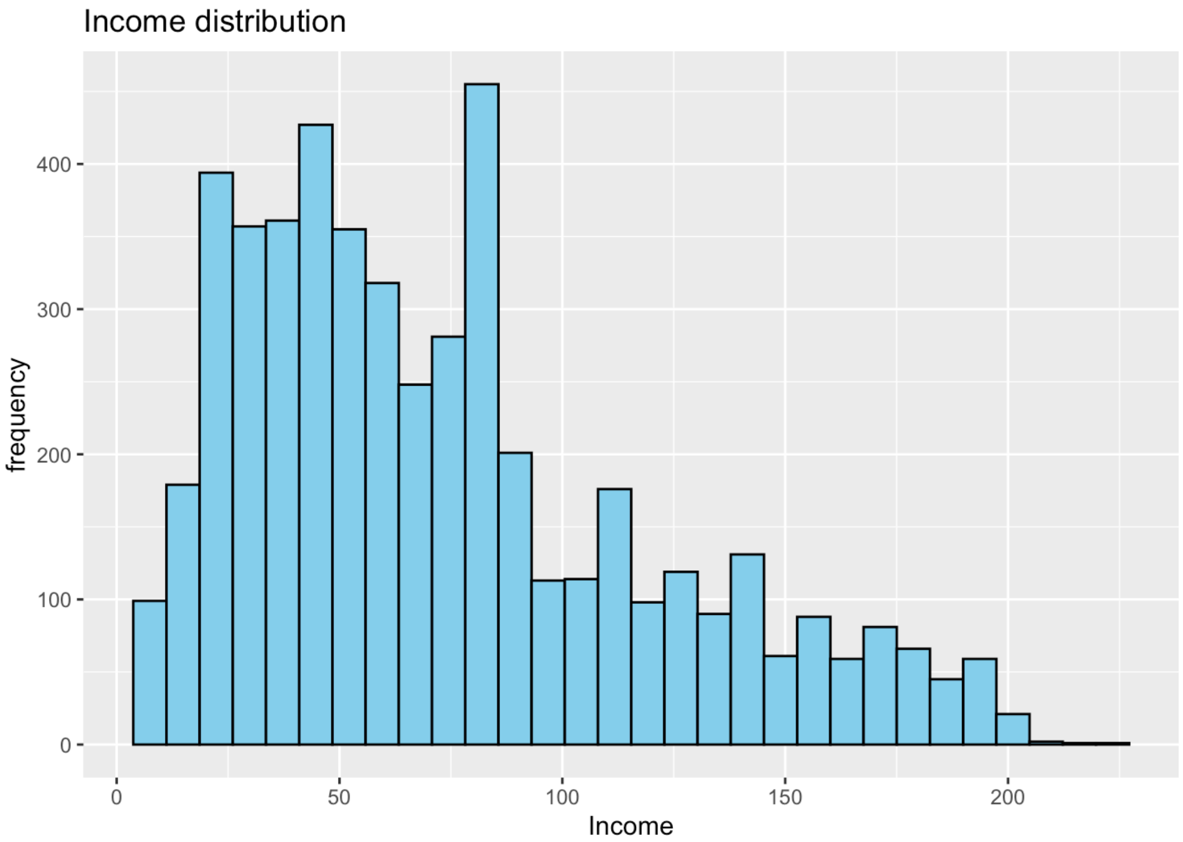


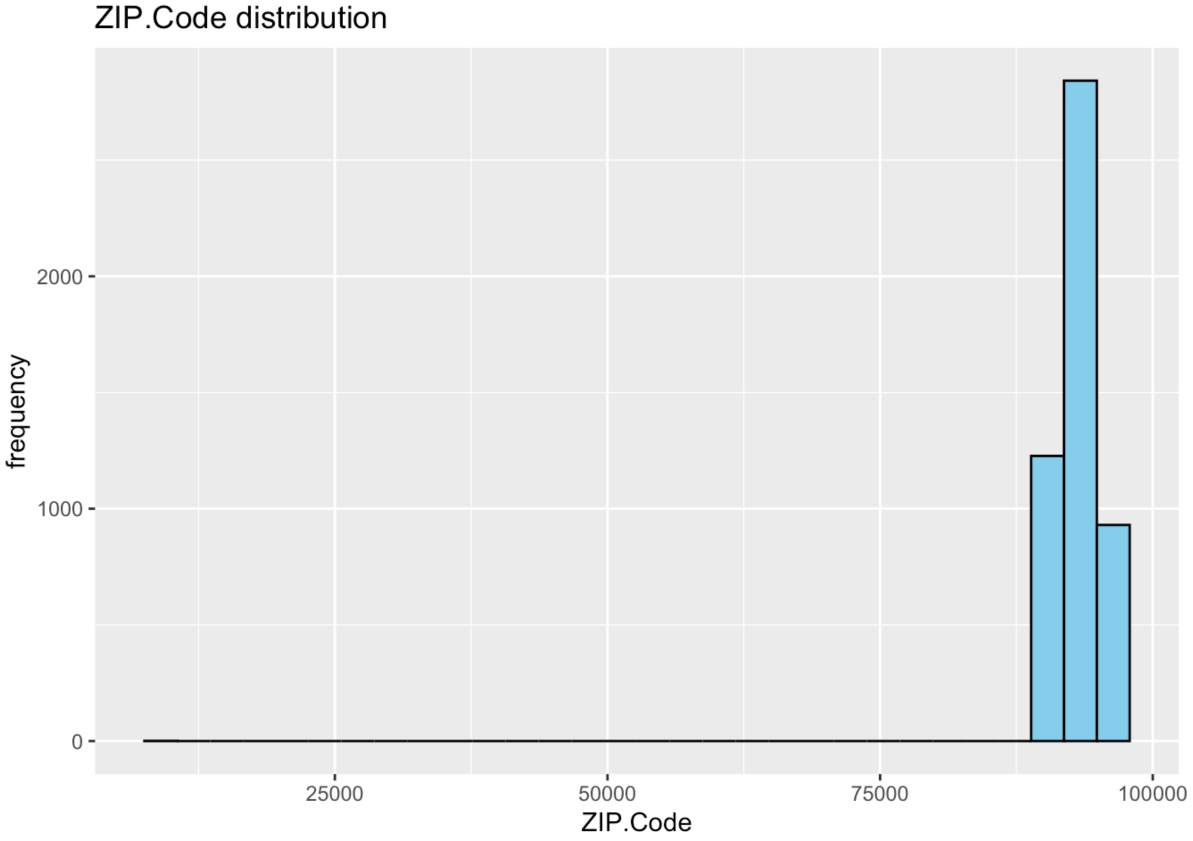
```
numeric_features <- setdiff(names(data)[sapply(data, is.numeric)], "Personal.Loan")

for (feature in numeric_features) {
  # Check if the feature is continuous
  if (length(unique(data[[feature]])) > 10) { # Arbitrary threshold, adjust based on your data
    p <- ggplot(data, aes_string(x = feature)) +
      geom_histogram(bins = 30, fill = "skyblue", color = "black") +
      labs(title = paste(feature, "distribution"), x = feature, y = "frequency")
  } else {
    # For discrete data, use a bar plot
    p <- ggplot(data, aes_string(x = feature)) +
      geom_bar(fill = "orange", color = "black") +
      labs(title = paste(feature, "distribution"), x = feature, y = "count")
  }
  print(p)
}
```

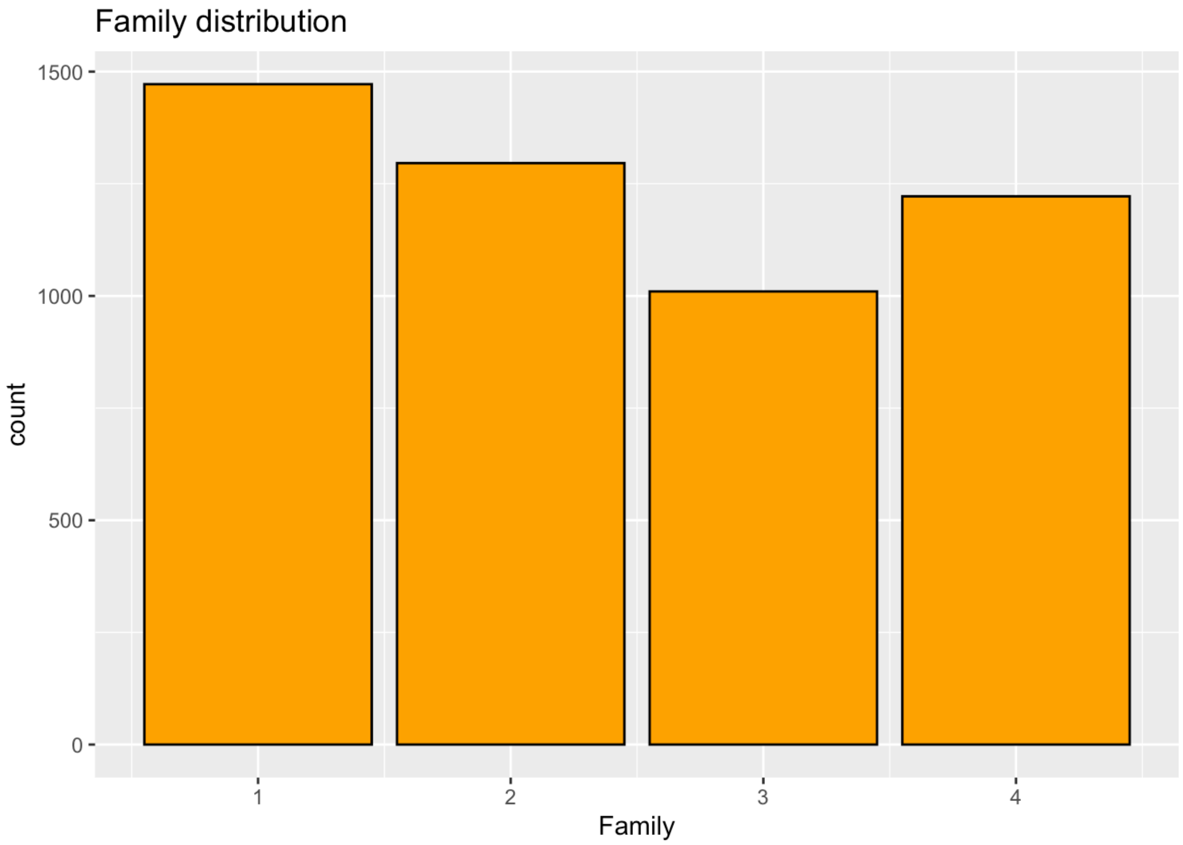
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

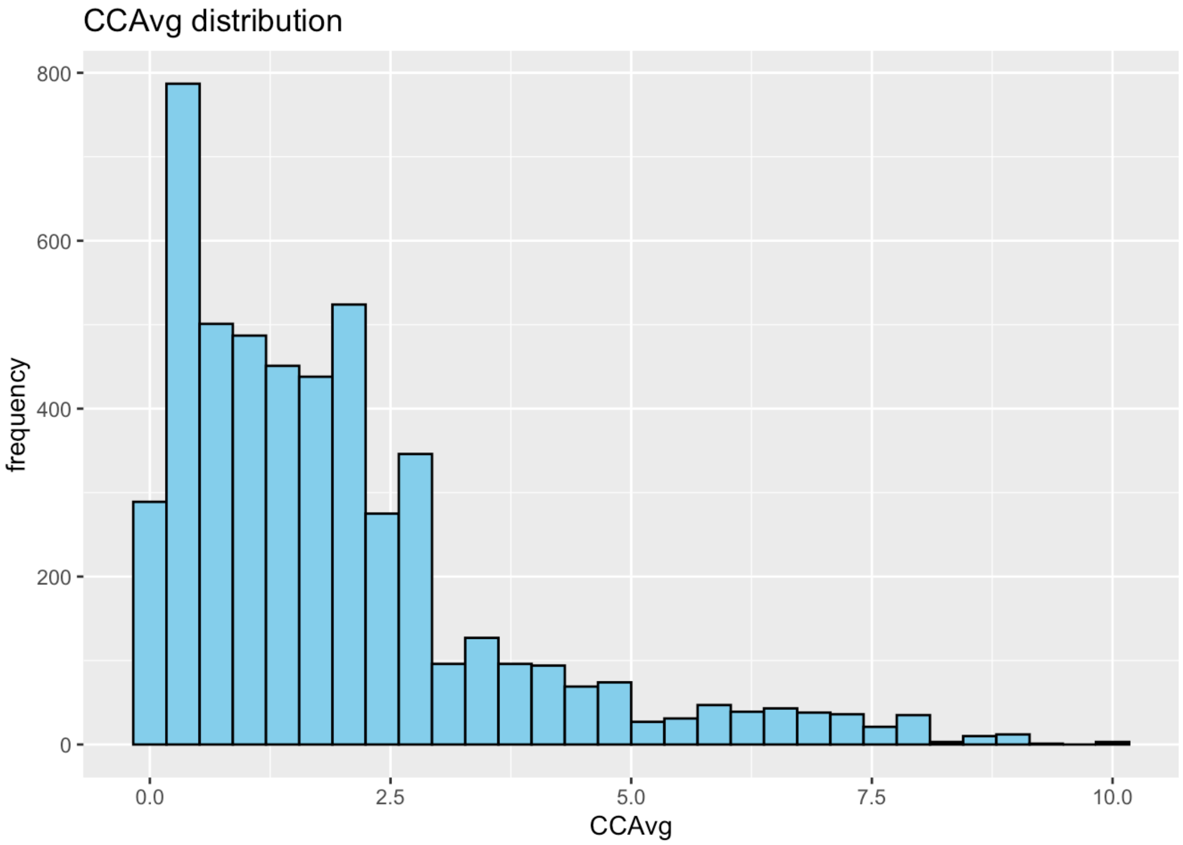

Age distribution
frequency
0
100
200
20
30
40
50
60
Age

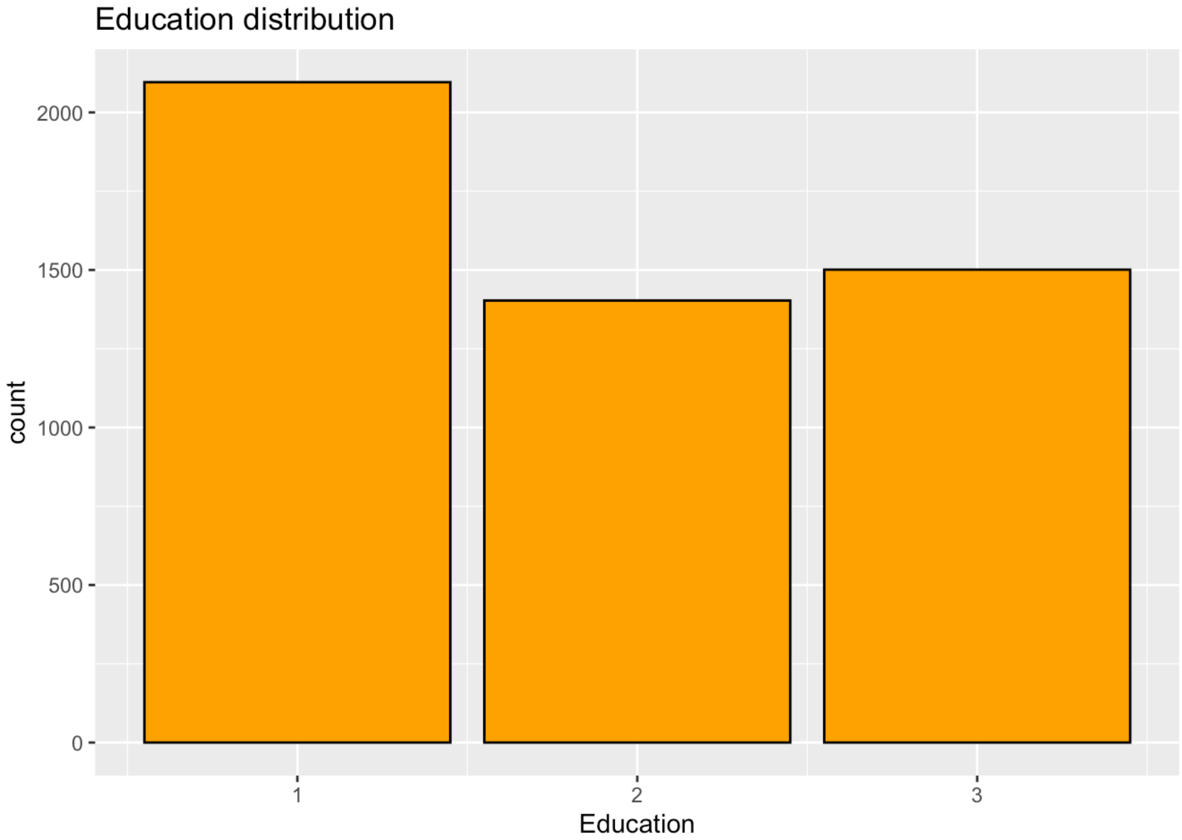

Experience distribution
frequency
0
100
200
0
10
20
30
40
Experience

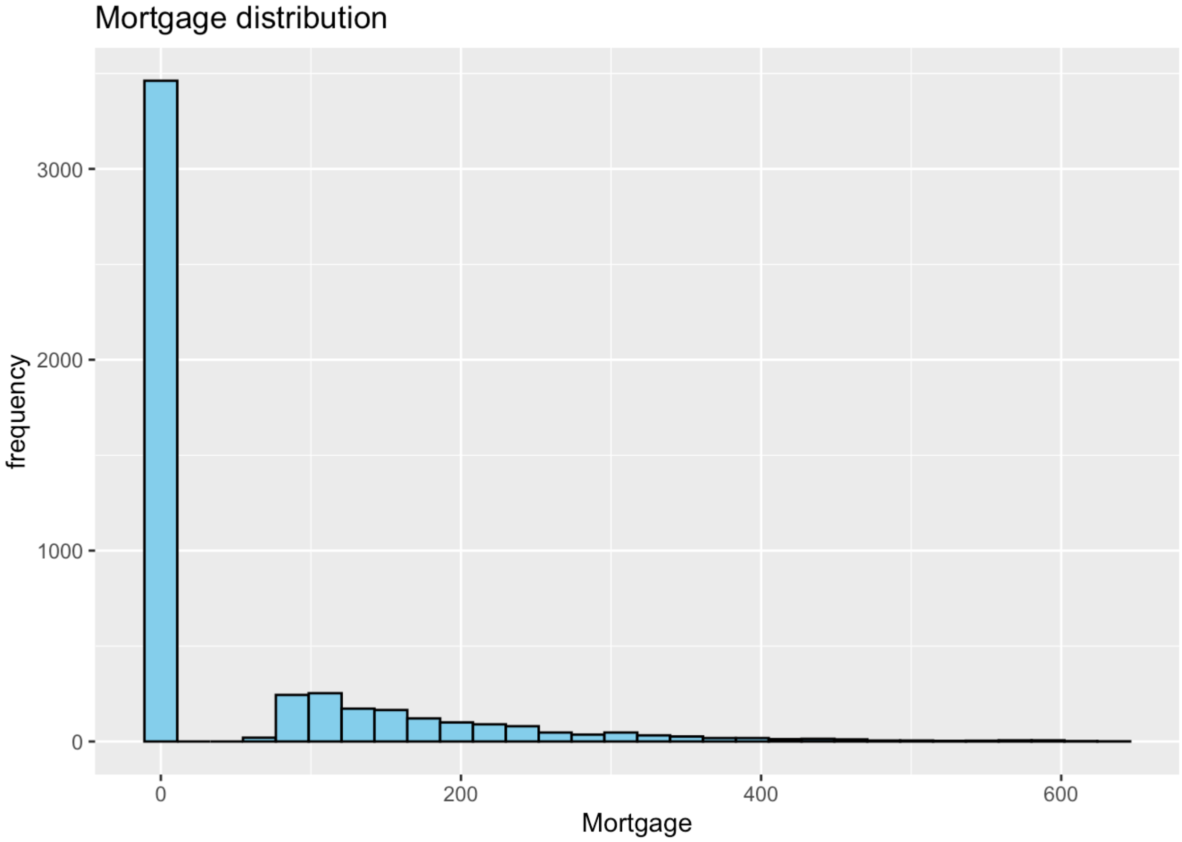

Income distribution
frequency
0
100
200
300
400
0
50
100
150
200
Income

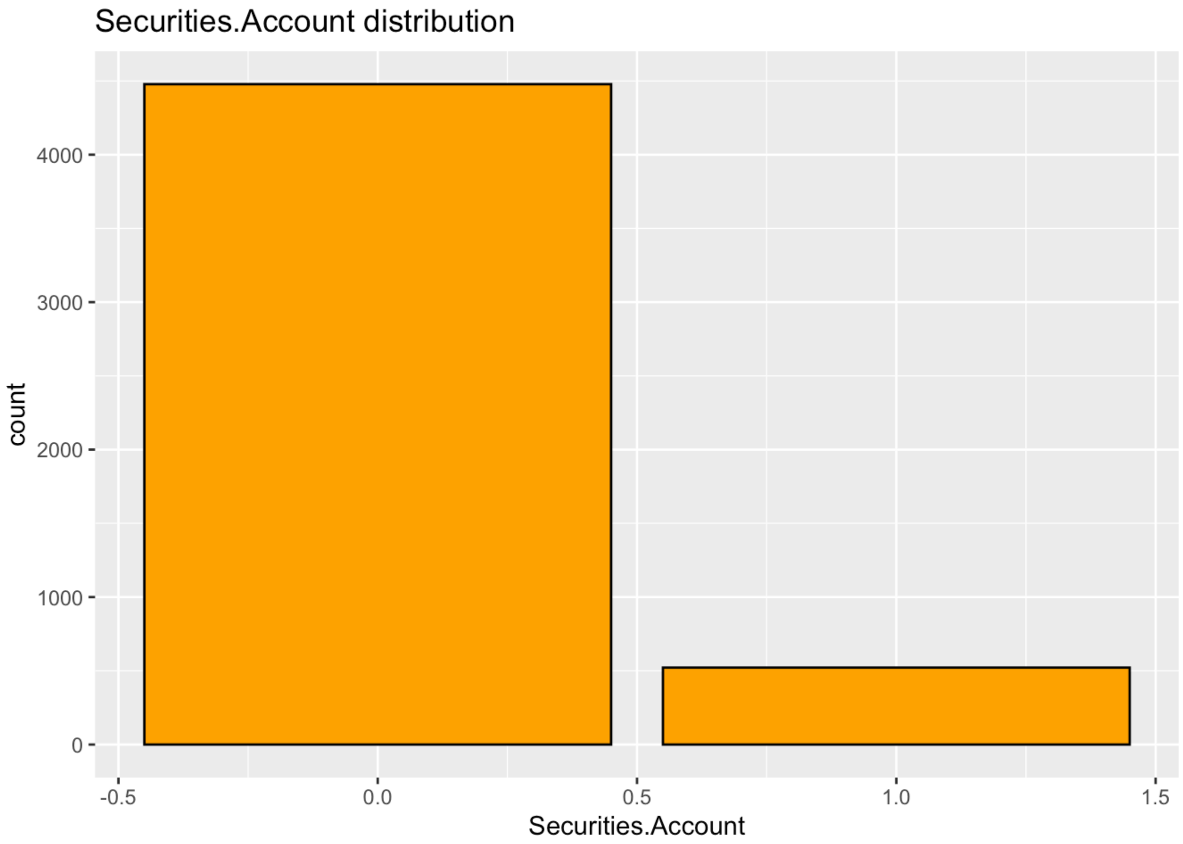

ZIP.Code distribution
frequency
0
1000
2000
25000
50000
75000
100000
ZIP.Code

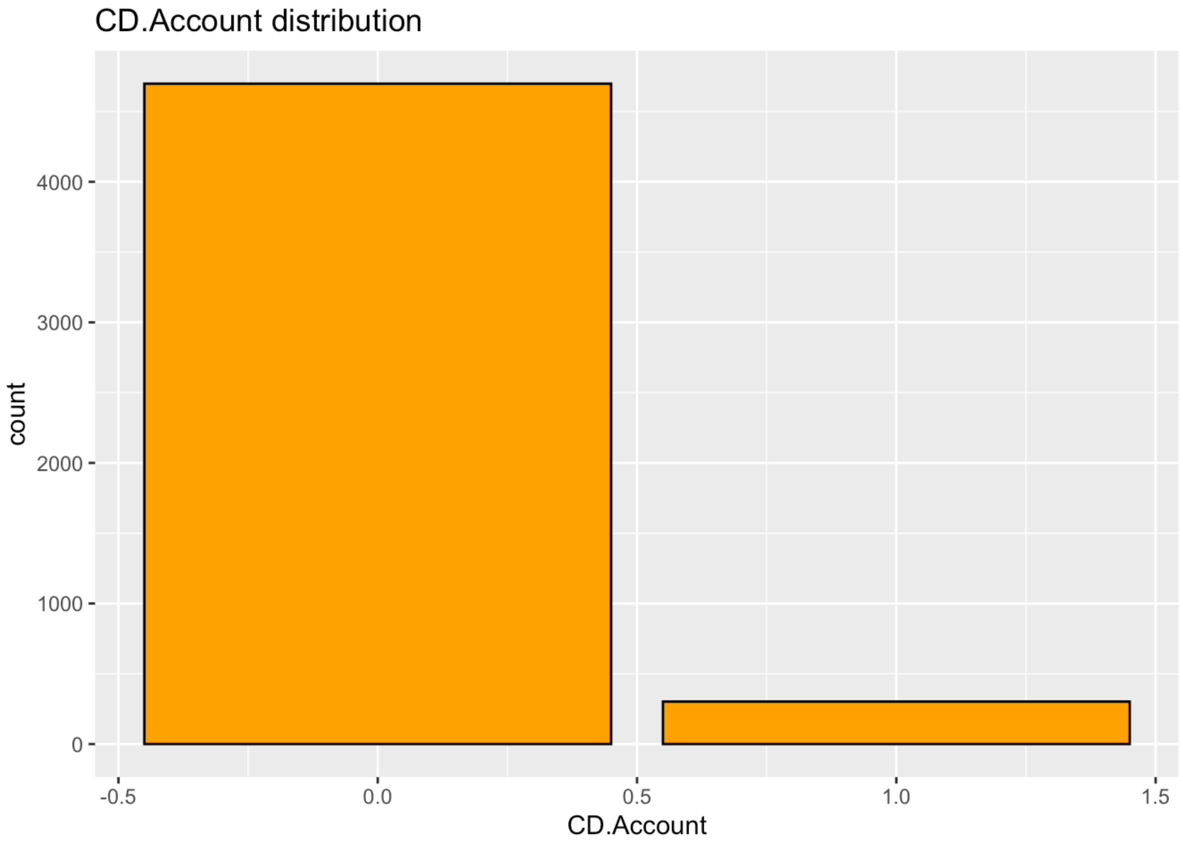

Family distribution
1500
1000
500
0
count
1
2
3
4
Family

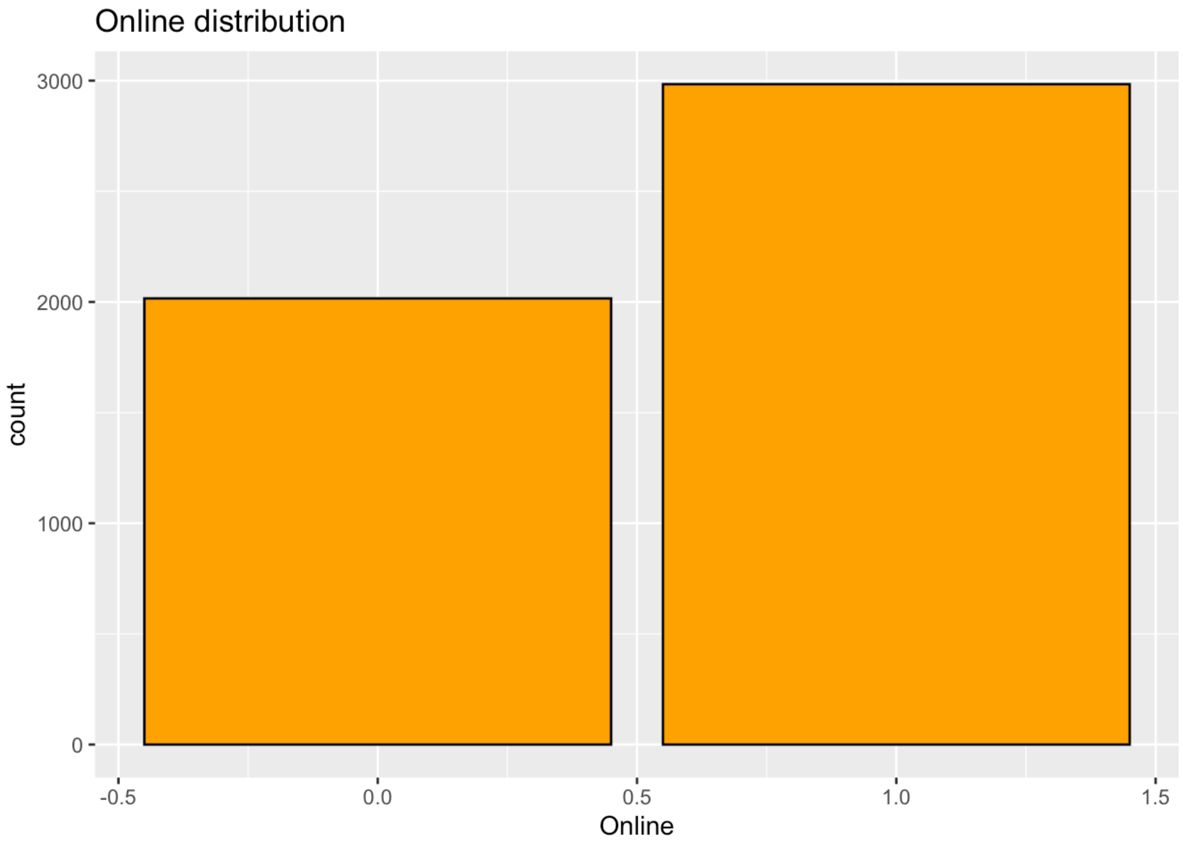

CCAvg distribution
800
600
400
200
0
frequency
0.0
2.5
5.0
7.5
10.0
CCAvg

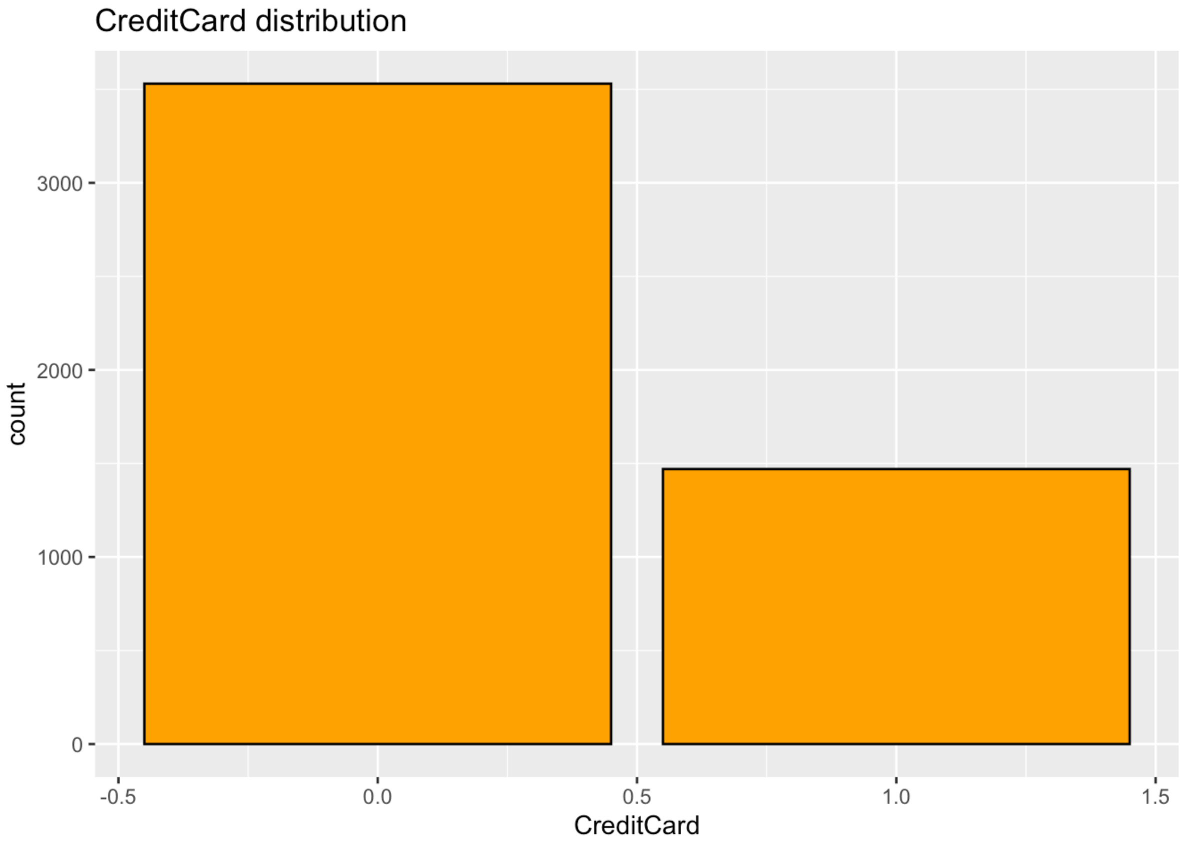

Education distribution
count
2000
1500
1000
500
0
1
2
3
Education


Mortgage distribution
frequency
3000
2000
1000
0
0
200
400
600
Mortgage


Securities.Account distribution
count
4000
3000
2000
1000
0
-0.5
0.0
0.5
1.0
1.5
Securities.Account


CD.Account distribution
count
4000
3000
2000
1000
0
-0.5
0.0
0.5
1.0
1.5
CD.Account



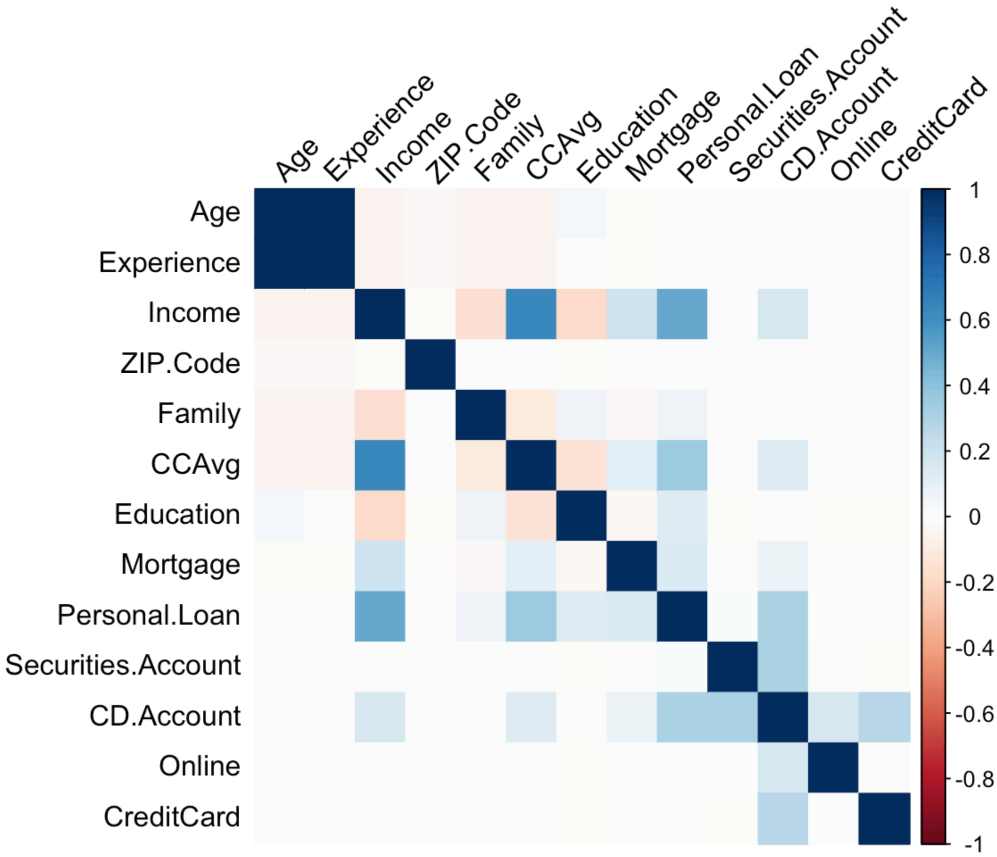


```
personal_loan_data = readr::read_csv("https://www.louisaslett.com/Courses/MISCADA/bank_personal_
loan.csv")
```

```
## Rows: 5000 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (13): Age, Experience, Income, ZIP.Code, Family, CCAvg, Education, Mortg...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

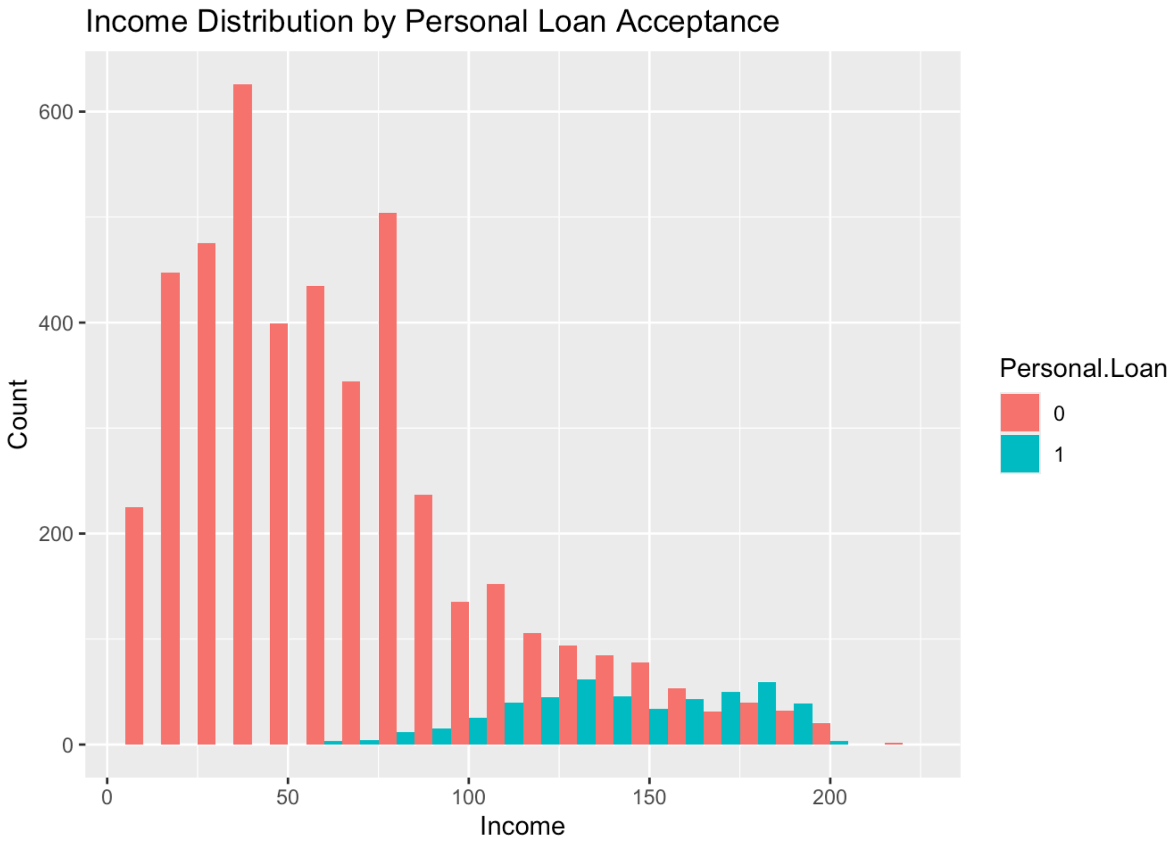
```
personal_loan_data$Education <- as.numeric(personal_loan_data$Education)
personal_loan_data$Personal.Loan <- as.numeric(personal_loan_data$Personal.Loan)
personal_loan_data$Securities.Account <- as.numeric(personal_loan_data$Securities.Account)
personal_loan_data$CD.Account <- as.numeric(personal_loan_data$CD.Account)
personal_loan_data$Online <- as.numeric(personal_loan_data$Online)
personal_loan_data$CreditCard <- as.numeric(personal_loan_data$CreditCard)
```

```
cor_matrix <- cor(personal_loan_data[, sapply(personal_loan_data, is.numeric)], use="complete.ob
s")
corrplot(cor_matrix, method="color", tl.col="black", tl.srt=45)
```

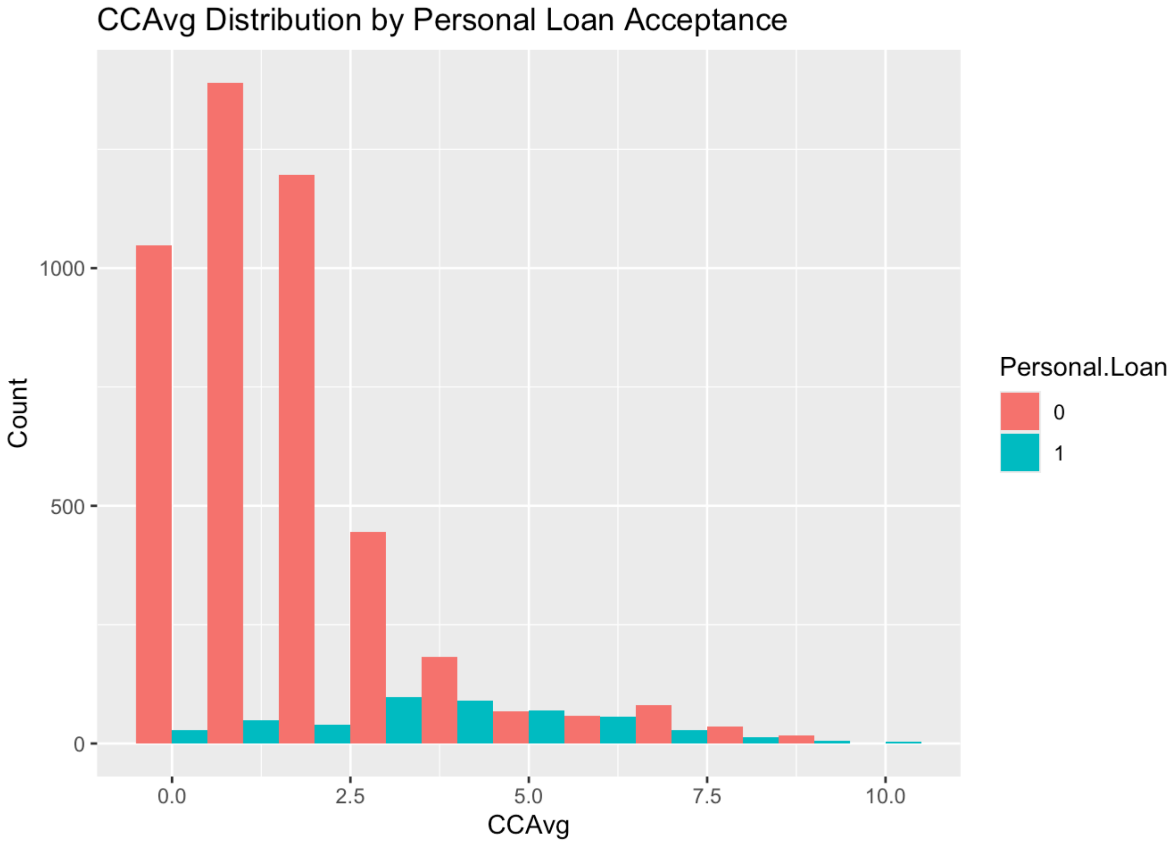


```
personal_loan_data$Education <- as.factor(personal_loan_data$Education)
personal_loan_data$CD.Account <- as.factor(personal_loan_data$CD.Account)
personal_loan_data$Personal.Loan <- as.factor(personal_loan_data$Personal.Loan)
```

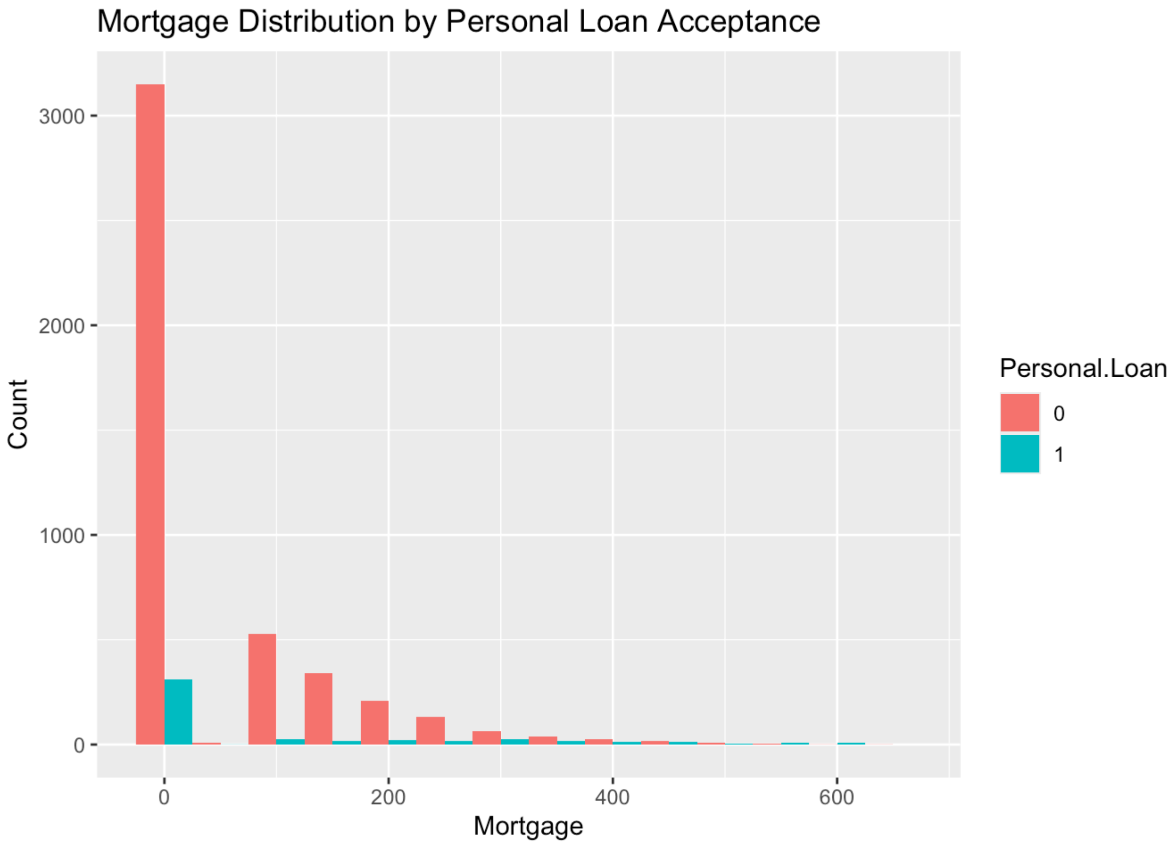
```
ggplot(personal_loan_data, aes(x = Income, fill = Personal.Loan)) +
  geom_histogram(binwidth = 10, position = "dodge") +
  labs(title = "Income Distribution by Personal Loan Acceptance",
       x = "Income",
       y = "Count")
```



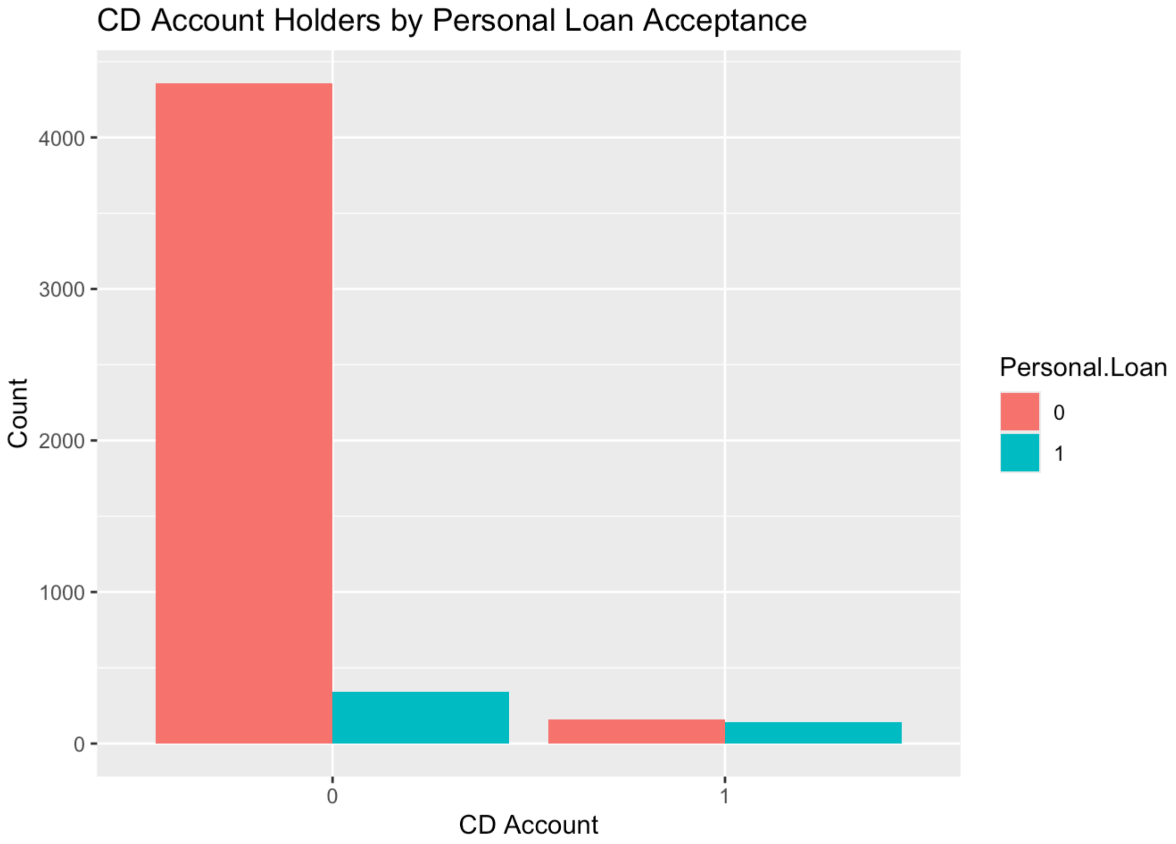
```
ggplot(personal_loan_data, aes(x = CCAvg, fill = Personal.Loan)) +
  geom_histogram(binwidth = 1, position = "dodge") +
  labs(title = "CCAvg Distribution by Personal Loan Acceptance", x = "CCAvg", y = "Count")
```



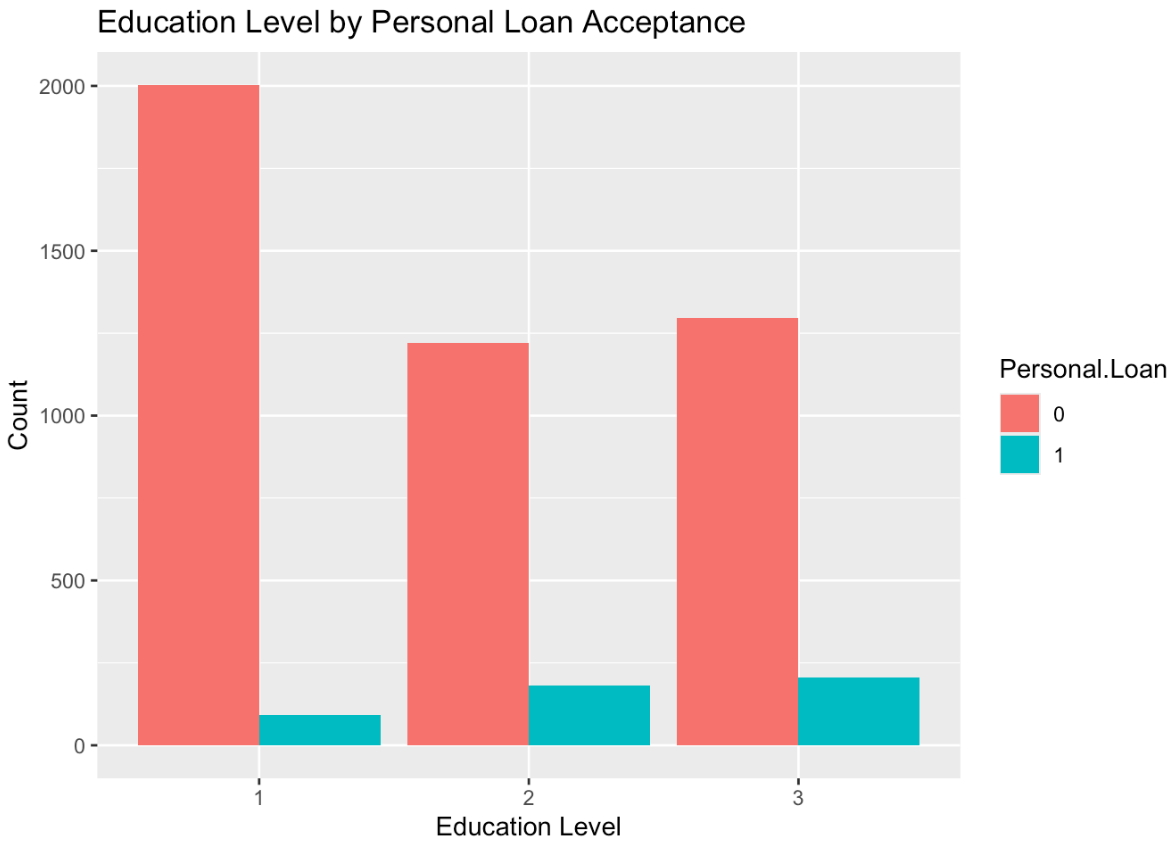
```
ggplot(personal_loan_data, aes(x = Mortgage, fill = Personal.Loan)) +
     geom_histogram(binwidth = 50, position = "dodge") +
     labs(title = "Mortgage Distribution by Personal Loan Acceptance", x = "Mortgage", y = "Coun
t")
```



```
ggplot(personal_loan_data, aes(x = CD.Account, fill = Personal.Loan)) +
     geom_bar(position = "dodge") +
     labs(title = "CD Account Holders by Personal Loan Acceptance", x = "CD Account", y = "Coun
t")
```

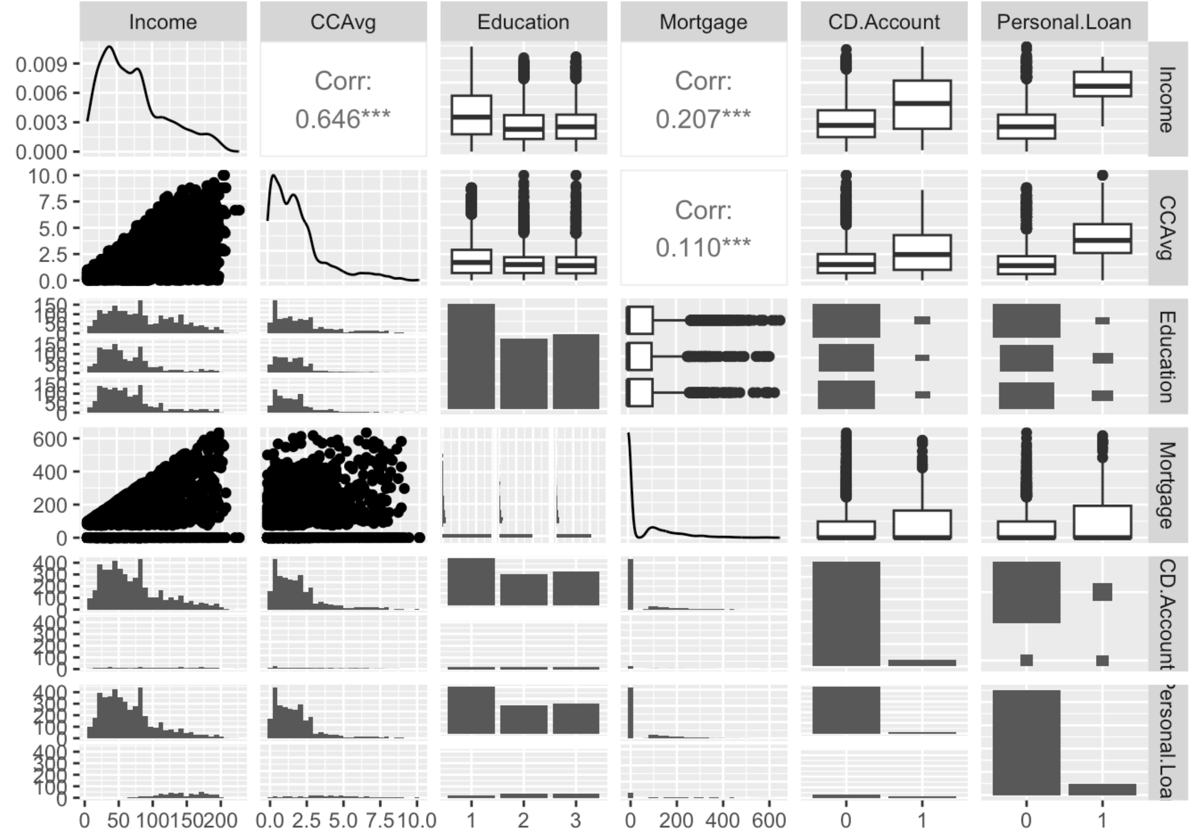


```
ggplot(personal_loan_data, aes(x = Education, fill = Personal.Loan)) +
     geom_bar(position = "dodge") +
     labs(title = "Education Level by Personal Loan Acceptance", x = "Education Level", y = "Cou
nt")
```



```
selected_vars <- personal_loan_data %>% select(Income, CCAvg, Education, Mortgage, CD.Account, P
ersonal.Loan)
ggpairs(selected_vars)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Part1 Use caret to fit the model and evaluate the performance.

```
# Set seed value to ensure reproducibility of results.
set.seed(123)

# Data preprocessing.
# Make sure the levels of the factor variables are valid R variable names.
data$Personal.Loan <- factor(data$Personal.Loan, levels = c("0", "1"), labels = c("No", "Yes"))
data$Securities.Account <- factor(data$Securities.Account, levels = c("0", "1"), labels = c("N
o", "Yes"))
data$CD.Account <- factor(data$CD.Account, levels = c("0", "1"), labels = c("No", "Yes"))
data$Online <- factor(data$Online, levels = c("0", "1"), labels = c("No", "Yes"))
data$CreditCard <- factor(data$CreditCard, levels = c("0", "1"), labels = c("No", "Yes"))
```

```
# Divide training set and test set
index <- createDataPartition(data$Personal.Loan, p = 0.8, list = FALSE)
train_data <- data[index, ]
test_data <- data[-index, ]

# Set cross-validation method
control <- trainControl(method="cv", number=10, classProbs=TRUE, summaryFunction=twoClassSummar
y)
```

```
# Train logistic regression model
logit_model <- train(Personal.Loan ~ ., data=train_data, method="glm", family="binomial", trCont
rol=control, metric="ROC")

print(logit_model)
```

```
## Generalized Linear Model
##
## 4000 samples
##   12 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 3600, 3599, 3599, 3601, 3599, 3600, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.9605702  0.9839687  0.6404184
```

```
summary(logit_model)
```

```
##
## Call:
## NULL
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -1.329e+01  4.707e+00  -2.824 0.004744 **
## Age                    -4.790e-02  7.127e-02  -0.672 0.501555
## Experience              5.632e-02  7.062e-02   0.797 0.425162
## Income                  5.571e-02  3.009e-03  18.516  < 2e-16 ***
## ZIP.Code                6.579e-06  4.612e-05   0.143 0.886581
## Family                  7.961e-01  8.630e-02   9.225  < 2e-16 ***
## CCAvg                   1.520e-01  4.572e-02   3.325 0.000883 ***
## Education               1.714e+00  1.282e-01  13.364  < 2e-16 ***
## Mortgage                8.090e-04  6.206e-04   1.304 0.192373
## Securities.AccountYes -8.175e-01  3.200e-01  -2.554 0.010635 *
## CD.AccountYes           3.921e+00  3.737e-01  10.492  < 2e-16 ***
## OnlineYes              -8.233e-01  1.793e-01  -4.591 4.41e-06 ***
## CreditCardYes          -1.225e+00  2.361e-01  -5.186 2.14e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2529.63  on 3999  degrees of freedom
## Residual deviance:  995.01  on 3987  degrees of freedom
## AIC: 1021
##
## Number of Fisher Scoring iterations: 8
```

```
# Train support vector machine model
svm_model <- train(Personal.Loan ~ ., data=train_data, method="svmRadial", trControl=control, me
tric="ROC", preProcess=c("center", "scale"))

print(svm_model)
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 4000 samples
##   12 predictor
##    2 classes: 'No', 'Yes'
##
## Pre-processing: centered (12), scaled (12)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 3601, 3599, 3600, 3601, 3599, 3601, ...
## Resampling results across tuning parameters:
##
##   C     ROC        Sens       Spec
##   0.25  0.9812739  0.9892158  0.7916329
##   0.50  0.9838530  0.9908733  0.8123482
##   1.00  0.9852311  0.9919790  0.8307018
##
## Tuning parameter 'sigma' was held constant at a value of 0.07016594
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.07016594 and C = 1.
```

```
summary(svm_model)
```

```
## Length  Class   Mode
##      1   ksvm     S4
```

```
# Train random forest model
rf_model <- train(Personal.Loan ~ ., data=train_data, method="rf", trControl=control, metric="ROC", ntree=100)

print(rf_model)
```

```
## Random Forest
##
## 4000 samples
##   12 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 3599, 3599, 3600, 3601, 3601, 3600, ...
## Resampling results across tuning parameters:
##
##   mtry  ROC        Sens       Spec
##    2    0.9960061  0.9997230  0.8466937
##    7    0.9973462  0.9983402  0.9011471
##   12    0.9976981  0.9969575  0.9010121
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 12.
```

```
summary(rf_model)
```

```
##                 Length Class      Mode     
## call               5   -none-     call     
## type               1   -none-     character
## predicted       4000   factor     numeric  
## err.rate         300   -none-     numeric  
## confusion          6   -none-     numeric  
## votes           8000   matrix     numeric  
## oob.times       4000   -none-     numeric  
## classes            2   -none-     character
## importance        12   -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               4000   factor     numeric  
## test               0   -none-     NULL     
## inbag              0   -none-     NULL     
## xNames            12   -none-     character
## problemType        1   -none-     character
## tuneValue          1   data.frame list     
## obsLevels          2   -none-     character
## param              1   -none-     list
```

```
# Predict and evaluate the model
predictions <- list(
  logit = predict(logit_model, test_data, type="prob")[, "Yes"],
  svm = predict(svm_model, test_data, type="prob")[, "Yes"],
  rf = predict(rf_model, test_data, type="prob")[, "Yes"]
)

# Calculate ROC curve and AUC
roc_results <- lapply(predictions, function(pred) roc(response = as.numeric(test_data$Personal.L
oan) - 1, predictor = pred))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
# Calculate and format AUC values
auc_values <- sapply(roc_results, function(x) round(auc(x), 4))

# Format and print AUC value
cat("\nAUC Values for Models:\n")
```

```
## 
## AUC Values for Models:
```

```
cat("Logistic Regression: ", auc_values["logit"], "\n")
```

```
## Logistic Regression:  0.9483
```

```
cat("SVM:                 ", auc_values["svm"], "\n")
```

```
## SVM:                  0.983
```

```
cat("Random Forest:       ", auc_values["rf"], "\n")
```

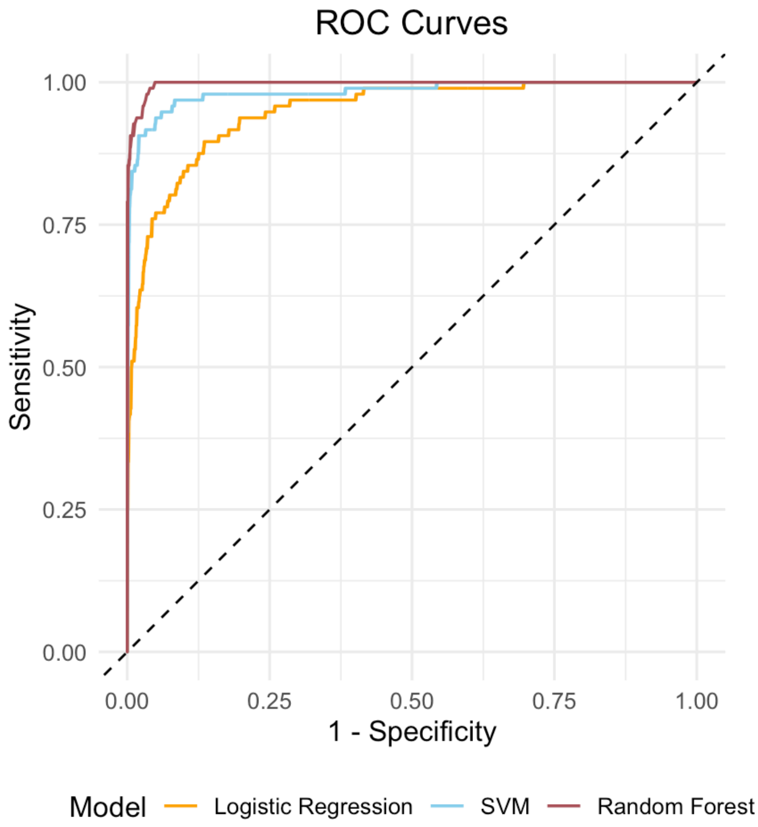
```
## Random Forest:        0.9972
```

```
# Prepare ROC data
roc_data <- data.frame(
  model = factor(c(rep("Logistic Regression", length(roc_results$logit$sensitivities)),
                   rep("SVM", length(roc_results$svm$sensitivities)),
                   rep("Random Forest", length(roc_results$rf$sensitivities))),
                 levels = c("Logistic Regression", "SVM", "Random Forest")),
  specificity = c(1 - roc_results$logit$specificities,
                  1 - roc_results$svm$specificities,
                  1 - roc_results$rf$specificities),
  sensitivity = c(roc_results$logit$sensitivities,
                  roc_results$svm$sensitivities,
                  roc_results$rf$sensitivities)
)

# Draw ROC curve
roc_plot <- ggplot(roc_data, aes(x = specificity, y = sensitivity, color = model)) +
  geom_line(size = 0.6) +
  scale_color_manual(values = c("orange", "skyblue", "#AB545A")) +
  labs(x = "1 - Specificity", y = "Sensitivity", color = "Model", title = "ROC Curves") +
  theme_minimal() +
  theme(
    text = element_text(size = 12),
    legend.position = "bottom",
    legend.title.align = 0.5,
    plot.title = element_text(hjust = 0.5)
  ) +
  coord_fixed(ratio = 1) +
  geom_abline(linetype = "dashed")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `legend.title.align` argument of `theme()` is deprecated as of ggplot2
## 3.5.0.
## ℹ Please use theme(legend.title = element_text(hjust)) instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
# Print ROC curve graph
print(roc_plot)
```



Part2 Use mlr3 for model fitting

```
if (!requireNamespace("mlr3", quietly = TRUE)) install.packages("mlr3")
if (!requireNamespace("mlr3learners", quietly = TRUE)) install.packages("mlr3learners")
if (!requireNamespace("mlr3tuning", quietly = TRUE)) install.packages("mlr3tuning")
if (!requireNamespace("mlr3viz", quietly = TRUE)) install.packages("mlr3viz")
if (!requireNamespace("data.table", quietly = TRUE)) install.packages("data.table")
if (!requireNamespace("ggplot2", quietly = TRUE)) install.packages("ggplot2")
if (!requireNamespace("precrec", quietly = TRUE)) install.packages("precrec")

library("mlr3")
```

```
##
## Attaching package: 'mlr3'
```

```
## The following object is masked from 'package:skimr':
##
##     partition
```

```
library("mlr3learners")
library("mlr3tuning")
```

```
## Loading required package: paradox
```

```
##
## Attaching package: 'mlr3tuning'
```

```
## The following object is masked from 'package:e1071':
##
##     tune
```

```
library("mlr3viz")
library("data.table")
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library("ggplot2")
library("precrec")
```

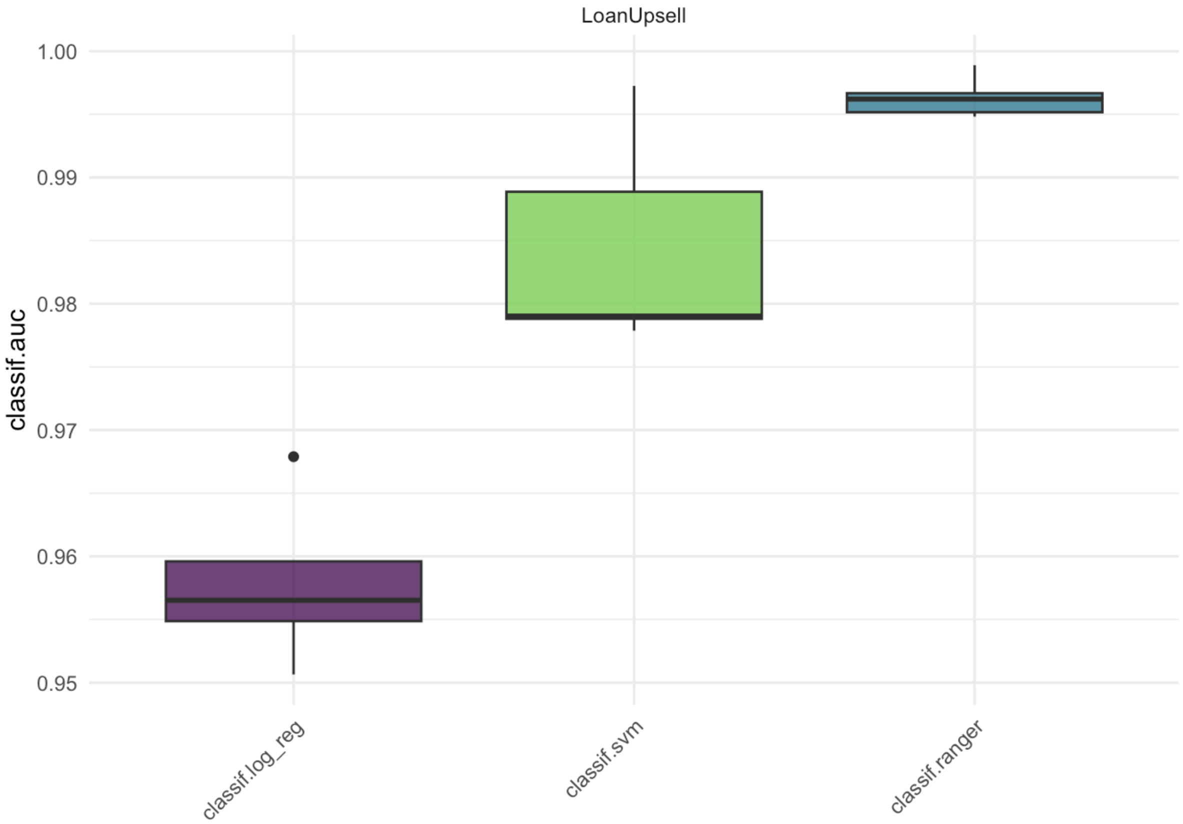
```
##
## Attaching package: 'precrec'
```

```
## The following object is masked from 'package:pROC':
##
##     auc
```

```
# new data
data <- fread("https://www.louisaslett.com/Courses/MISCADA/bank_personal_loan.csv")

# Data preprocessing: Convert the target variable Personal.Loan to factor type
data$Personal.Loan <- as.factor(data$Personal.Loan)

# Create a task
task <- TaskClassif$new(id = "LoanUpsell", backend = data, target = "Personal.Loan")

# Three different classification algorithms (logistic regression, SVM, random forest) were selected for model training and comparison.
learners <- list(
  lrn("classif.log_reg", predict_type = "prob"),
  lrn("classif.svm", predict_type = "prob"),
  lrn("classif.ranger", predict_type = "prob")
)

# Define 10-fold cross-validation to evaluate model performance.
resampling <- rsmp("cv", folds = 5)

# Define evaluation design
design <- benchmark_grid(
  tasks = task,
  learners = learners,
  resamplings = resampling
)

if (!is.data.frame(design)) {
  stop("The design is not a data frame structure.")
}

# Function trains and compares models, aggregating results based on the AUC metric.
bmr <- benchmark(design)
```

```
## INFO  [05:48:00.256] [mlr3] Running benchmark with 15 resampling iterations
## INFO  [05:48:00.301] [mlr3] Applying learner 'classif.log_reg' on task 'LoanUpsell' (iter 1/
5)
## INFO  [05:48:00.345] [mlr3] Applying learner 'classif.log_reg' on task 'LoanUpsell' (iter 2/
5)
## INFO  [05:48:00.377] [mlr3] Applying learner 'classif.log_reg' on task 'LoanUpsell' (iter 3/
5)
## INFO  [05:48:00.423] [mlr3] Applying learner 'classif.log_reg' on task 'LoanUpsell' (iter 4/
5)
## INFO  [05:48:00.451] [mlr3] Applying learner 'classif.log_reg' on task 'LoanUpsell' (iter 5/
5)
## INFO  [05:48:00.483] [mlr3] Applying learner 'classif.svm' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:00.936] [mlr3] Applying learner 'classif.svm' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:01.393] [mlr3] Applying learner 'classif.svm' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:01.870] [mlr3] Applying learner 'classif.svm' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:02.330] [mlr3] Applying learner 'classif.svm' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:02.779] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:03.429] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:04.027] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:04.614] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:05.217] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:05.807] [mlr3] Finished benchmark
```

```
bmr_results <- bmr$aggregate(msr("classif.auc"))

# Visualize the AUC performance of different models.
autoplot(bmr, measure = msr("classif.auc"))
```



```
# Create a new classification task task_rf_optimization
task_rf_optimization <- TaskClassif$new(id = "LoanUpsell", backend = data, target = "Personal.Loan")

# Create a learner for a random forest model
# Set predict_type to prob to predict probability
learner_rf <- lrn("classif.ranger", predict_type = "prob")

# Use ParamSet$new to define the parameter set param_set for model tuning
# mtry: Number of variables considered in each split, ranging from sqrt(ncol(data)/3) to sqrt(ncol(data)).
# min.node.size: The minimum number of samples of leaf nodes of the tree, ranging from 1 to 10.
# num.trees: Number of trees in the forest, ranging from 100 to 1000.
param_set <- ParamSet$new(params = list(
  ParamInt$new("mtry", lower = as.integer(sqrt(ncol(data)/3)), upper = as.integer(sqrt(ncol(data)))),
  ParamInt$new("min.node.size", lower = 1, upper = 10),
  ParamInt$new("num.trees", lower = 100, upper = 1000)
))

# A random search tuner is defined, and 10 parameter combinations are randomly selected for evaluation in each batch.
tuner <- tnr("random_search", batch_size = 10)

# Create an automatic tuner at
at <- AutoTuner$new(
  learner = learner_rf,                    # random forest learner
  resampling = rsmp("cv", folds = 5),     # 10-fold cross validation
  measure = msr("classif.auc"),            # AUC performance metric
  tuner = tuner,                           # Random search tuner
  search_space = param_set,                # Parameter search space
  terminator = trm("evals", n_evals = 50)  # Stopping criterion is set to 50 evaluations
)

# Training the autotuner
at$train(task_rf_optimization)
```

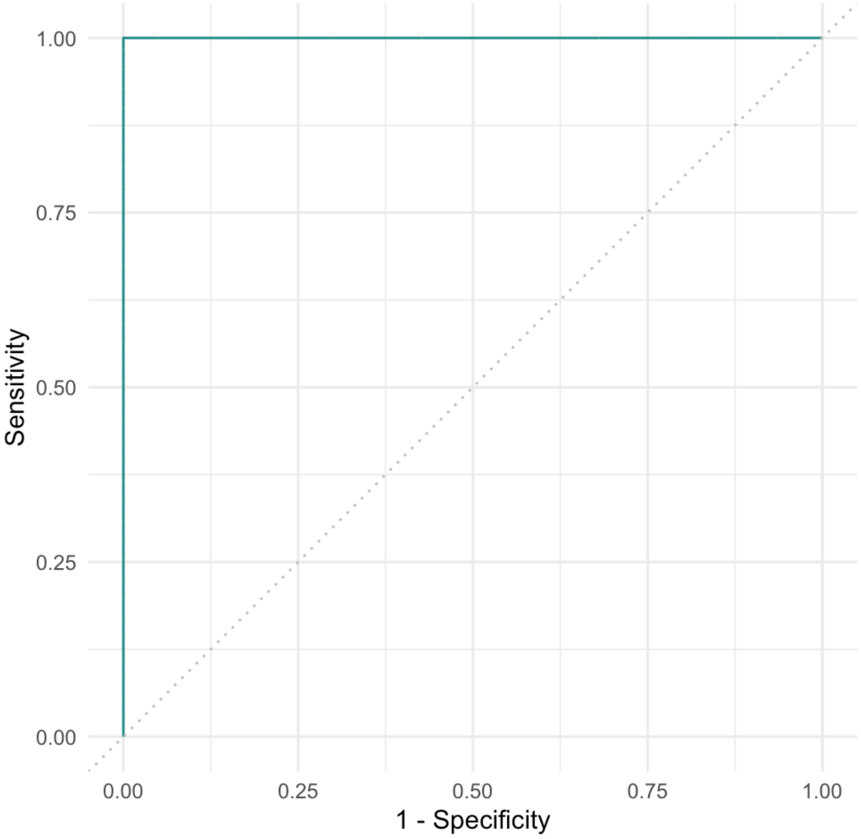
```
## INFO  [05:48:06.385] [bbotk] Starting to optimize 3 parameter(s) with '<OptimizerRandomSearch
>' and '<TerminatorEvals> [n_evals=50, k=0]'
## INFO  [05:48:06.400] [bbotk] Evaluating 10 configuration(s)
## INFO  [05:48:06.409] [mlr3] Running benchmark with 50 resampling iterations
## INFO  [05:48:06.414] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:07.259] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:08.098] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:08.933] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:09.790] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:10.644] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:10.853] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:11.053] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:11.260] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:11.469] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:11.680] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:12.620] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:13.551] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:14.500] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:15.431] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:16.367] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:17.046] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:17.712] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:18.387] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:19.127] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:19.806] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:20.402] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:20.992] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:21.592] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:22.191] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:22.785] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:23.839] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:24.885] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:25.943] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:26.988] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:28.040] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:28.412] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:28.784] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:29.166] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:29.533] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:29.909] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:31.098] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:32.278] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:33.488] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:34.664] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:35.856] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:36.306] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:36.755] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:37.214] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:37.661] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:38.120] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:39.212] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:40.294] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:41.412] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:42.506] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:43.607] [mlr3] Finished benchmark
## INFO  [05:48:43.709] [bbotk] Result of batch 1:
## INFO  [05:48:43.712] [bbotk]  mtry min.node.size num.trees classif.auc warnings errors runtim
e_learners
## INFO  [05:48:43.712] [bbotk]     2             9       692   0.9959392        0      0
4.179
## INFO  [05:48:43.712] [bbotk]     3             8       161   0.9970142        0      0
0.996
```

```
## INFO  [05:48:43.712] [bbotk]     2             1       731   0.9965932        0      0
4.648
## INFO  [05:48:43.712] [bbotk]     3             4       552   0.9971873        0      0
3.392
## INFO  [05:48:43.712] [bbotk]     2             2       461   0.9965770        0      0
2.940
## INFO  [05:48:43.712] [bbotk]     3             5       870   0.9971987        0      0
5.213
## INFO  [05:48:43.712] [bbotk]     3             7       305   0.9970579        0      0
1.831
## INFO  [05:48:43.712] [bbotk]     3             4       988   0.9972232        0      0
5.907
## INFO  [05:48:43.712] [bbotk]     3             1       357   0.9973317        0      0
2.220
## INFO  [05:48:43.712] [bbotk]     2             9       902   0.9960873        0      0
5.441
## INFO  [05:48:43.712] [bbotk]                                 uhash
## INFO  [05:48:43.712] [bbotk]  daaa911d-e2bf-4b77-9e7c-ad2381f60f7b
## INFO  [05:48:43.712] [bbotk]  706e242f-05b1-401a-ab78-8351550e0dfa
## INFO  [05:48:43.712] [bbotk]  f276c8ca-ff7e-4fd2-8557-6d7652810a14
## INFO  [05:48:43.712] [bbotk]  65dda3a4-a83f-4f94-9636-67f983ed8925
## INFO  [05:48:43.712] [bbotk]  52e83c75-2de5-4772-ba16-94775b81ed54
## INFO  [05:48:43.712] [bbotk]  a9bf7662-676e-4c69-8e68-050028be284a
## INFO  [05:48:43.712] [bbotk]  a906b5c7-676f-4309-b963-d189148d1f56
## INFO  [05:48:43.712] [bbotk]  83e79562-6956-46de-8533-d81824e8064c
## INFO  [05:48:43.712] [bbotk]  961aced0-5a85-44d5-9900-0d0d85092c94
## INFO  [05:48:43.712] [bbotk]  6fbdd132-be4b-4772-b0f5-120b82df6e4e
## INFO  [05:48:43.715] [bbotk] Evaluating 10 configuration(s)
## INFO  [05:48:43.721] [mlr3] Running benchmark with 50 resampling iterations
## INFO  [05:48:43.725] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:44.736] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:45.765] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:46.746] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:47.727] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:48.724] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:49.709] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:50.691] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:51.697] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:52.689] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:53.679] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:48:54.831] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:48:55.968] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:48:57.131] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:48:58.300] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:48:59.454] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:00.126] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:00.800] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:01.465] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:02.133] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:02.811] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:03.565] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:04.314] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:05.076] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:05.821] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:06.589] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:07.263] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:07.921] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:08.591] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:09.257] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:10.023] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:11.054] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:12.058] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:13.097] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:14.128] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:15.153] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:16.409] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:17.625] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:18.843] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:20.059] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:21.276] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:21.593] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:21.888] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:22.189] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:22.487] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:22.781] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:23.212] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:23.636] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:24.071] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:24.513] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:24.964] [mlr3] Finished benchmark
## INFO  [05:49:25.059] [bbotk] Result of batch 2:
## INFO  [05:49:25.061] [bbotk]  mtry min.node.size num.trees classif.auc warnings errors runtim
e_learners
## INFO  [05:49:25.061] [bbotk]     2             8       800   0.9960886        0      0
4.959
## INFO  [05:49:25.061] [bbotk]     3             7       830   0.9971512        0      0
4.915
## INFO  [05:49:25.061] [bbotk]     2            10       951   0.9961352        0      0
5.736
## INFO  [05:49:25.061] [bbotk]     2            10       547   0.9961177        0      0
3.317
## INFO  [05:49:25.061] [bbotk]     3             8       636   0.9970311        0      0
3.736
## INFO  [05:49:25.061] [bbotk]     2             9       542   0.9958664        0      0
3.396
## INFO  [05:49:25.061] [bbotk]     3             2       830   0.9972349        0      0
5.089
## INFO  [05:49:25.061] [bbotk]     2             4       970   0.9966713        0      0
6.083
## INFO  [05:49:25.061] [bbotk]     2             1       223   0.9967423        0      0
1.464
## INFO  [05:49:25.061] [bbotk]     3             1       340   0.9973056        0      0
2.136
## INFO  [05:49:25.061] [bbotk]                                 uhash
## INFO  [05:49:25.061] [bbotk]  642dff33-6fe6-4b5d-b2f3-a5fab278e9fb
## INFO  [05:49:25.061] [bbotk]  96ba016d-09e7-40fe-b84b-40c8ce51409a
## INFO  [05:49:25.061] [bbotk]  f91c536c-f559-41c4-a791-e0703c9fb2cc
## INFO  [05:49:25.061] [bbotk]  08e37df9-f69d-4aca-8916-13158c12318d
## INFO  [05:49:25.061] [bbotk]  3acd3ece-7980-492d-90ce-2fce8a18b5b8
## INFO  [05:49:25.061] [bbotk]  c9ff9718-c284-4707-9b9c-71709884ed78
## INFO  [05:49:25.061] [bbotk]  e161764d-7dd9-4e17-abe8-1a9068ed1cfb
## INFO  [05:49:25.061] [bbotk]  ab50b49d-fd7d-4af5-8e1b-3be0d2136d6b
## INFO  [05:49:25.061] [bbotk]  9adae21a-3f9f-4117-9b31-2724ef142e44
## INFO  [05:49:25.061] [bbotk]  c1d69944-1937-47b7-b5dd-5a66e066d4bb
## INFO  [05:49:25.063] [bbotk] Evaluating 10 configuration(s)
## INFO  [05:49:25.070] [mlr3] Running benchmark with 50 resampling iterations
## INFO  [05:49:25.074] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:25.731] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:26.405] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:27.068] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:27.720] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:28.388] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:29.113] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:29.841] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:30.577] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:31.321] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:32.067] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:33.081] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:34.070] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:35.075] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:36.085] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:37.104] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:37.269] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:37.427] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:37.581] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:37.741] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:37.899] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:38.530] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:39.155] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:39.787] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:40.414] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:41.059] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:41.757] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:42.450] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:43.143] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:43.833] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:44.522] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:45.135] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:45.743] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:46.369] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:46.989] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:47.607] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:48.380] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:49.148] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:49.923] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:50.700] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:51.506] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:52.057] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:52.601] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:53.147] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:53.690] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:54.239] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:55.192] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:49:56.117] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:49:57.055] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:49:57.999] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:49:58.966] [mlr3] Finished benchmark
## INFO  [05:49:59.057] [bbotk] Result of batch 3:
## INFO  [05:49:59.059] [bbotk]  mtry min.node.size num.trees classif.auc warnings errors runtim
e_learners
## INFO  [05:49:59.059] [bbotk]     2             7       525   0.9961797        0      0
3.273
## INFO  [05:49:59.059] [bbotk]     3             6       606   0.9971688        0      0
3.630
## INFO  [05:49:59.059] [bbotk]     2             9       834   0.9960897        0      0
4.996
## INFO  [05:49:59.059] [bbotk]     2             9       115   0.9960621        0      0
0.755
## INFO  [05:49:59.059] [bbotk]     2             5       494   0.9963101        0      0
3.114
## INFO  [05:49:59.059] [bbotk]     3             3       563   0.9973102        0      0
3.426
## INFO  [05:49:59.059] [bbotk]     3             7       515   0.9970990        0      0
3.048
## INFO  [05:49:59.059] [bbotk]     3             2       626   0.9972493        0      0
3.856
## INFO  [05:49:59.059] [bbotk]     2             6       434   0.9961827        0      0
2.693
## INFO  [05:49:59.059] [bbotk]     2             5       755   0.9963319        0      0
```

```
4.666
## INFO  [05:49:59.059] [bbotk]                                  uhash
## INFO  [05:49:59.059] [bbotk]  169845fb-bed4-4754-9bcf-fbfdd34e4a65
## INFO  [05:49:59.059] [bbotk]  c8f2ba6b-f277-4a36-a1af-8b08388fec25
## INFO  [05:49:59.059] [bbotk]  c3ea0cbf-a3d3-4925-a081-8a86b726197a
## INFO  [05:49:59.059] [bbotk]  40b734ac-f37e-4400-90b5-7fa81869a514
## INFO  [05:49:59.059] [bbotk]  f7f34ded-dd94-465b-aa70-fe9c1c553dcb
## INFO  [05:49:59.059] [bbotk]  46c64f76-8281-42b1-9806-dab3b57f99f1
## INFO  [05:49:59.059] [bbotk]  3ff49198-8eaa-4af1-b49f-abcbfacd6082
## INFO  [05:49:59.059] [bbotk]  19bab87e-9cf5-4387-957a-209b7aa03be8
## INFO  [05:49:59.059] [bbotk]  8521ee1c-4818-4ad2-8cd8-ad8da4b3eb5b
## INFO  [05:49:59.059] [bbotk]  a901aa1d-bf0b-4a8e-9ad3-f4aee51e7476
## INFO  [05:49:59.062] [bbotk] Evaluating 10 configuration(s)
## INFO  [05:49:59.068] [mlr3] Running benchmark with 50 resampling iterations
## INFO  [05:49:59.072] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:49:59.747] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:00.426] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:01.095] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:01.795] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:02.482] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:02.682] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:02.891] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:03.101] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:03.313] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:03.520] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:04.010] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:04.488] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:04.969] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:05.458] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:05.937] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:06.984] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:08.027] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:09.074] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:10.118] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:11.189] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:12.099] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:13.008] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:13.923] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:14.853] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:15.789] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:16.492] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:17.175] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:17.877] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:18.566] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:19.259] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:19.936] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:20.620] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:21.310] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:21.995] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:22.675] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:23.304] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:23.922] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:24.538] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:25.157] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:25.782] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:26.640] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:27.486] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:28.341] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:29.189] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:30.054] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:30.455] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:30.854] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:31.259] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:31.659] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:32.069] [mlr3] Finished benchmark
## INFO  [05:50:32.170] [bbotk] Result of batch 4:
## INFO  [05:50:32.172] [bbotk]  mtry min.node.size num.trees classif.auc warnings errors runtim
e_learners
## INFO  [05:50:32.172] [bbotk]     2            10       554   0.9961312        0      0
3.369
## INFO  [05:50:32.172] [bbotk]     3             7       157   0.9971578        0      0
0.997
## INFO  [05:50:32.172] [bbotk]     2             7       386   0.9960109        0      0
2.369
## INFO  [05:50:32.172] [bbotk]     2             2       818   0.9964884        0      0
5.214
## INFO  [05:50:32.172] [bbotk]     2             2       715   0.9965607        0      0
4.561
## INFO  [05:50:32.172] [bbotk]     2             2       531   0.9963149        0      0
3.420
## INFO  [05:50:32.172] [bbotk]     3             8       572   0.9968995        0      0
3.376
## INFO  [05:50:32.172] [bbotk]     2             8       502   0.9962038        0      0
3.067
## INFO  [05:50:32.172] [bbotk]     3             4       701   0.9973064        0      0
4.232
## INFO  [05:50:32.172] [bbotk]     3             6       329   0.9968816        0      0
1.970
## INFO  [05:50:32.172] [bbotk]                                  uhash
## INFO  [05:50:32.172] [bbotk]  fa51911f-7043-45cb-b497-14492cbd3e53
## INFO  [05:50:32.172] [bbotk]  4621f24f-0f0e-49ad-be2b-5bd39d2985fa
## INFO  [05:50:32.172] [bbotk]  e57092f0-2f31-46e6-a3aa-c609a3333c38
## INFO  [05:50:32.172] [bbotk]  f7e2528d-1cfb-4398-8ca0-c204c049be24
## INFO  [05:50:32.172] [bbotk]  9ded94c6-5b65-4ac7-8c6a-368bda91728b
## INFO  [05:50:32.172] [bbotk]  1b074b9b-675f-4cbb-aeb8-38868ccb0c17
## INFO  [05:50:32.172] [bbotk]  4456cb56-56ac-4e56-aa33-934da63948dd
## INFO  [05:50:32.172] [bbotk]  17f626bd-4f96-4a87-86a6-ecca2975f7e4
## INFO  [05:50:32.172] [bbotk]  8316aa3e-288a-47e4-a24f-bc24b0c6d13d
## INFO  [05:50:32.172] [bbotk]  62128835-b7d5-4253-8b85-87cb37f7773c
## INFO  [05:50:32.174] [bbotk] Evaluating 10 configuration(s)
## INFO  [05:50:32.181] [mlr3] Running benchmark with 50 resampling iterations
## INFO  [05:50:32.185] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:32.596] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:32.996] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:33.410] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:33.833] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:34.247] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:35.181] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:36.103] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:37.052] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:37.985] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:38.924] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:39.767] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:40.621] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:41.467] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:42.305] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:43.172] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:43.869] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:44.555] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:45.461] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:46.149] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:46.853] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:47.709] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:48.557] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:49.423] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:50.287] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:51.167] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:52.360] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:53.519] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:50:54.720] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:50:55.901] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:50:57.082] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:50:58.238] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:50:59.368] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:51:00.511] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:51:01.649] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:51:02.793] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:51:03.153] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:51:03.507] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:51:03.880] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:51:04.239] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:51:04.606] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:51:04.803] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:51:04.996] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:51:05.193] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:51:05.388] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:51:05.585] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 1/5)
## INFO  [05:51:06.454] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 2/5)
## INFO  [05:51:07.301] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 3/5)
## INFO  [05:51:08.159] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 4/5)
## INFO  [05:51:09.008] [mlr3] Applying learner 'classif.ranger' on task 'LoanUpsell' (iter 5/5)
## INFO  [05:51:09.878] [mlr3] Finished benchmark
## INFO  [05:51:09.975] [bbotk] Result of batch 5:
## INFO  [05:51:09.976] [bbotk]  mtry min.node.size num.trees classif.auc warnings errors runtim
e_learners
## INFO  [05:51:09.976] [bbotk]     2             4       320   0.9962484        0      0
2.023
## INFO  [05:51:09.976] [bbotk]     3             5       779   0.9972839        0      0
4.638
## INFO  [05:51:09.976] [bbotk]     3             6       703   0.9971525        0      0
4.208
## INFO  [05:51:09.976] [bbotk]     3            10       586   0.9970841        0      0
3.642
## INFO  [05:51:09.976] [bbotk]     2             8       703   0.9962877        0      0
4.273
## INFO  [05:51:09.976] [bbotk]     3             5       987   0.9973160        0      0
5.873
## INFO  [05:51:09.976] [bbotk]     3            10       984   0.9970161        0      0
5.670
## INFO  [05:51:09.976] [bbotk]     3             1       281   0.9973471        0      0
1.765
## INFO  [05:51:09.976] [bbotk]     2             6       144   0.9962746        0      0
0.941
## INFO  [05:51:09.976] [bbotk]     2            10       710   0.9961833        0      0
4.248
## INFO  [05:51:09.976] [bbotk]                                  uhash
## INFO  [05:51:09.976] [bbotk]  bc69ab34-2778-4008-94fb-89efa568c2ee
## INFO  [05:51:09.976] [bbotk]  579db6be-ed34-40b9-85b7-dd21b3e079df
## INFO  [05:51:09.976] [bbotk]  c2712de1-dd7e-45ac-b5e2-f9c62b115d44
## INFO  [05:51:09.976] [bbotk]  d6014ee5-0355-4d57-888a-39fc00a31f03
## INFO  [05:51:09.976] [bbotk]  b6a77d5f-1491-49d3-8a87-dbe2a15d7d24
## INFO  [05:51:09.976] [bbotk]  e745a23a-f8ac-4000-bbef-7d8e7e6a46dc
## INFO  [05:51:09.976] [bbotk]  8df54427-f214-4c70-8814-c5def32a7262
## INFO  [05:51:09.976] [bbotk]  0b54afca-2a18-43cd-8897-9c7e96e1208e
## INFO  [05:51:09.976] [bbotk]  0ddf6a09-ecc7-4d77-abe9-51fa1f9d3664
## INFO  [05:51:09.976] [bbotk]  9b617a41-c00d-4a94-8d1a-a7c27bcb5bce
## INFO  [05:51:09.985] [bbotk] Finished optimizing after 50 evaluation(s)
## INFO  [05:51:09.985] [bbotk] Result:
## INFO  [05:51:09.987] [bbotk]   mtry min.node.size num.trees learner_param_vals  x_domain clas
```

```
sif.auc
## INFO  [05:51:09.987] [bbotk]  <int>          <int>     <int>             <list>    <list>    
<num>
## INFO  [05:51:09.987] [bbotk]      3              1       281          <list[4]> <list[3]>   0.
9973471
```

```
# Performance evaluation and plot roc
prediction <- at$predict(task_rf_optimization)
autoplot(prediction, type = "roc")
```



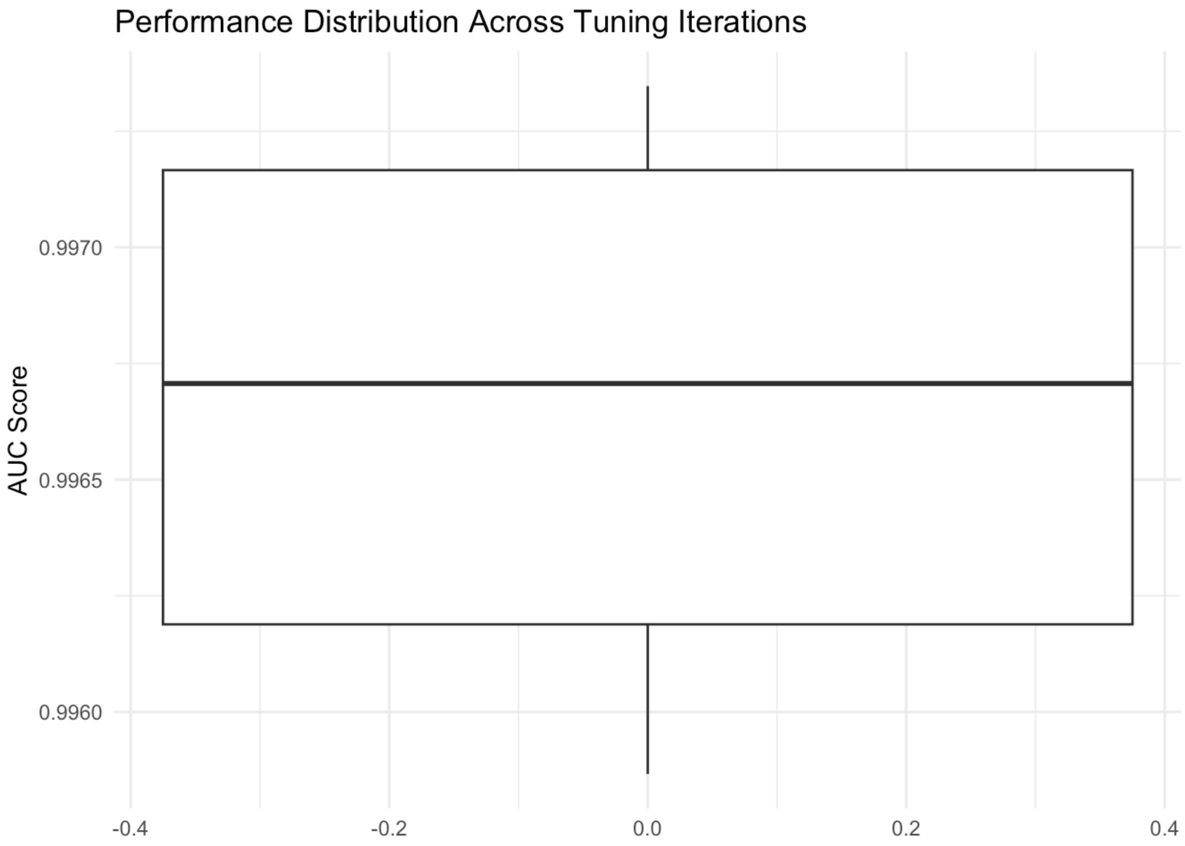
```
# Print confusion matrix conf_mat
conf_mat <- prediction$confusion
print(conf_mat)
```

```
##         truth
## response    0    1
##        0 4520    0
##        1    0  480
```

```
# The archive variable stores the detailed results of each evaluation during the tuning process.
archive <- at$archive
print(names(archive$data))
```

```
##  [1] "mtry"             "min.node.size"    "num.trees"        "classif.auc"     
##  [5] "warnings"         "errors"           "runtime_learners" "uhash"           
##  [9] "x_domain"         "timestamp"        "batch_nr"        
```

```
ggplot(archive$data, aes(y = classif.auc)) +
  geom_boxplot() +
  labs(title = "Performance Distribution Across Tuning Iterations",
       y = "AUC Score") +
  theme_minimal()
```



```
ggplot(archive$data, aes(x = mtry, y = classif.auc)) +
  geom_point() +
  geom_smooth() +
  labs(title = "mtry vs. AUC", x = "mtry", y = "AUC Score") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at 1.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 1.005
```

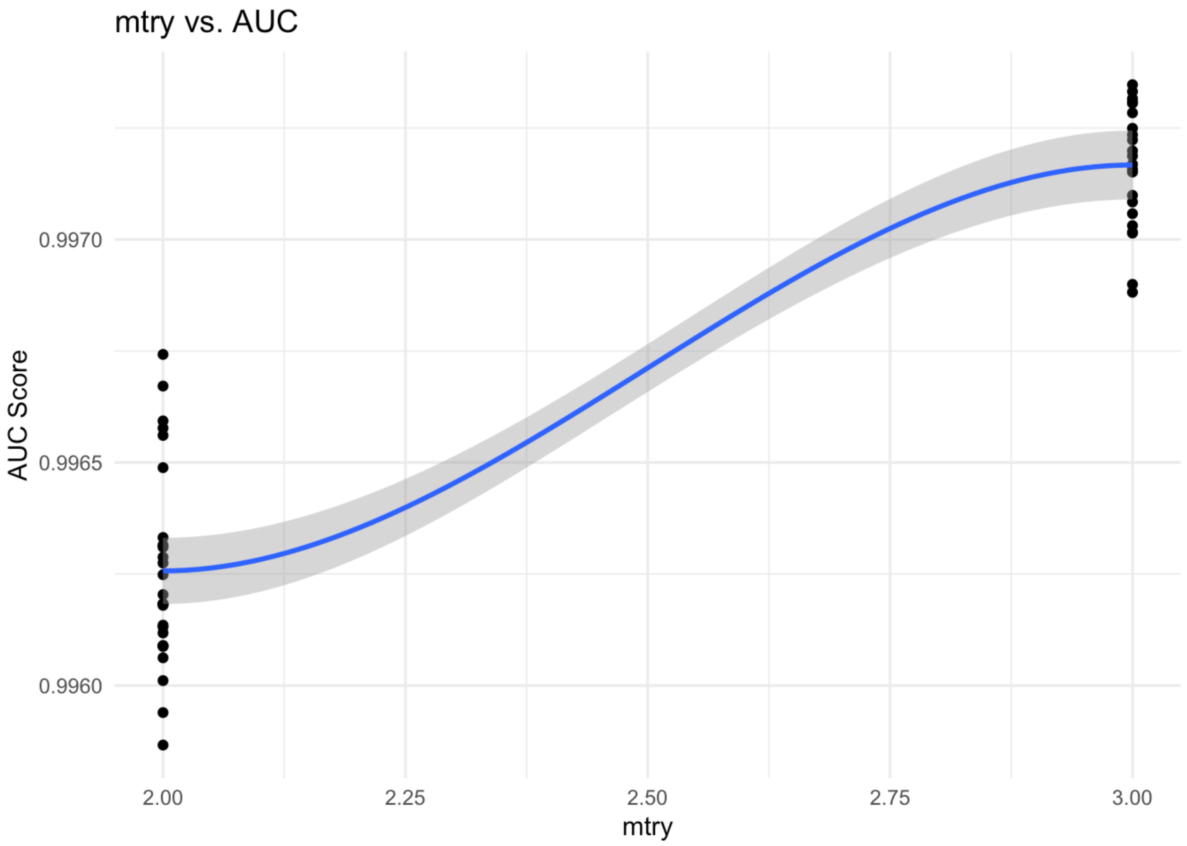
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 1.01
```

```
## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at
## 1.995
## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 1.005
```

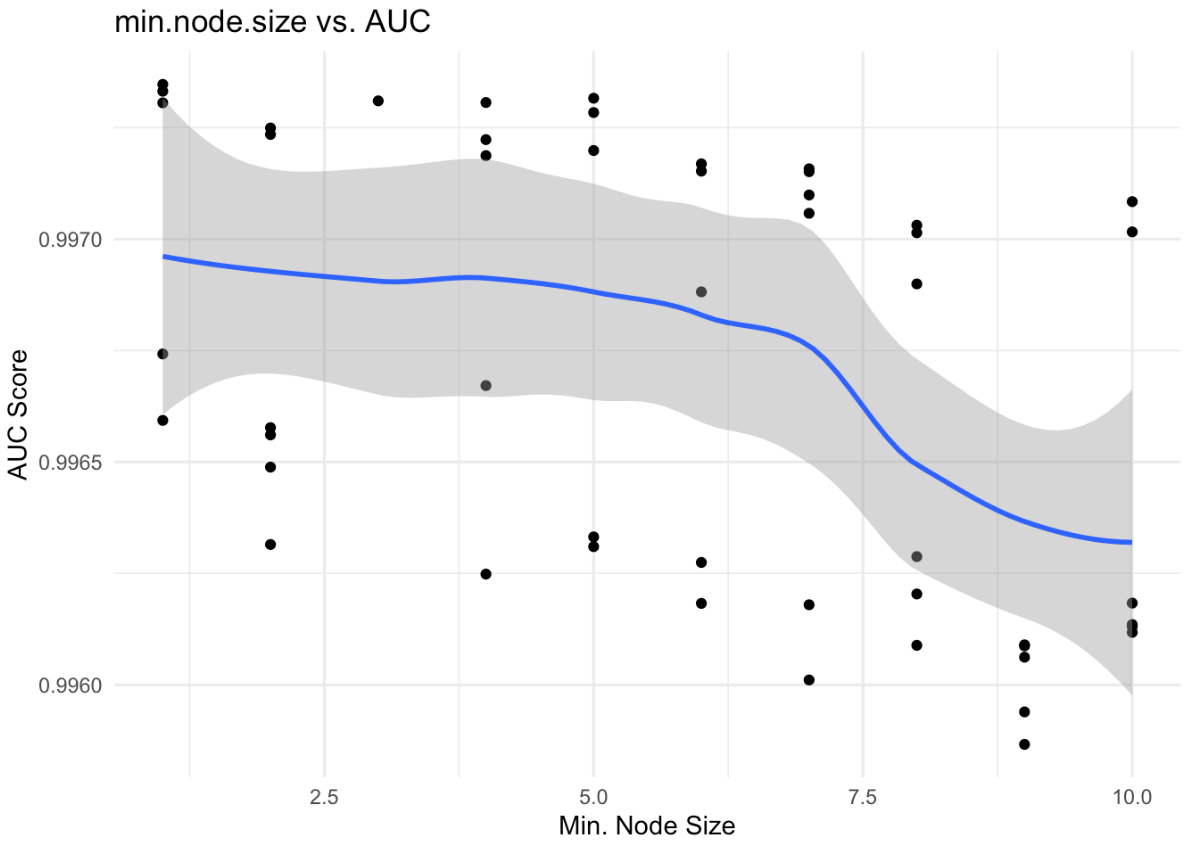
```
## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
## number 0
```

```
## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : There are other near
## singularities as well. 1.01
```



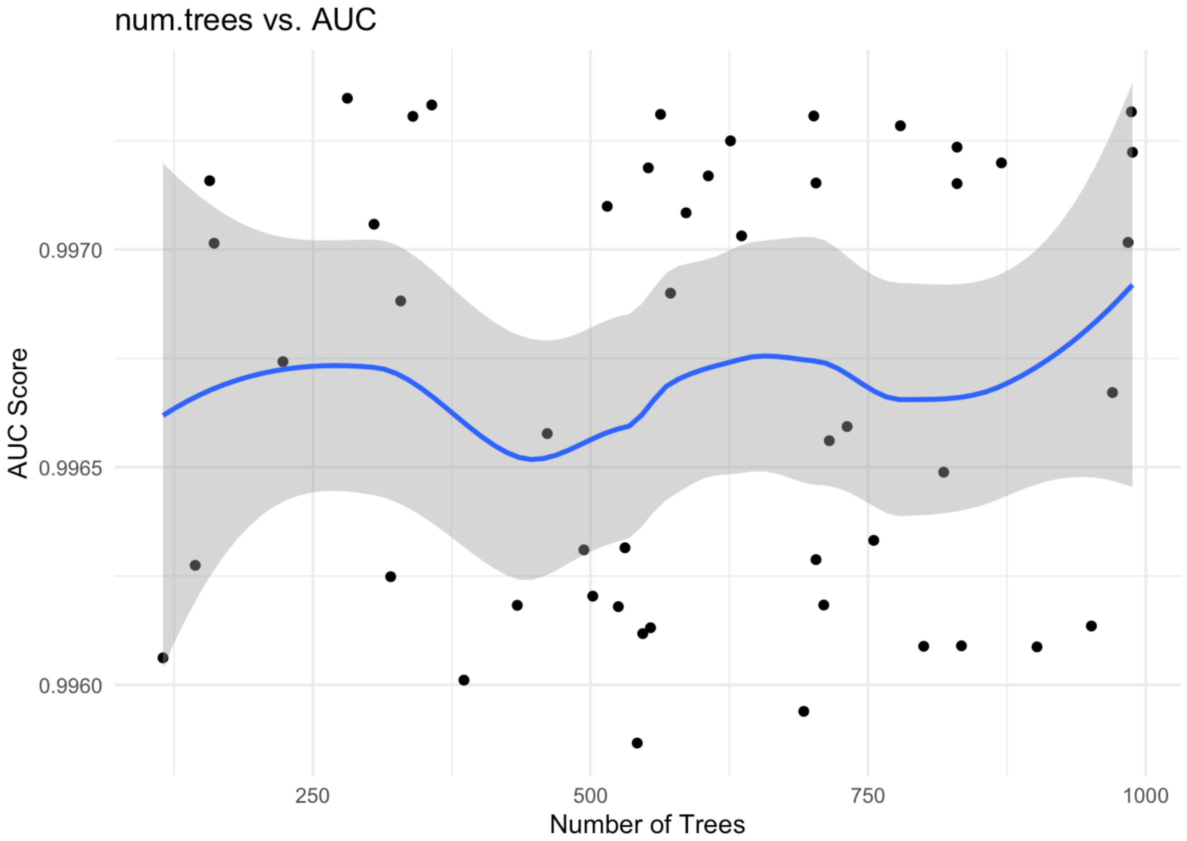
```
ggplot(archive$data, aes(x = min.node.size, y = classif.auc)) +
  geom_point() +
  geom_smooth() +
  labs(title = "min.node.size vs. AUC", x = "Min. Node Size", y = "AUC Score") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
ggplot(archive$data, aes(x = num.trees, y = classif.auc)) +
  geom_point() +
  geom_smooth() +
  labs(title = "num.trees vs. AUC", x = "Number of Trees", y = "AUC Score") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Part3 Deep learning using keras

```
if (!requireNamespace("keras", quietly = TRUE)) install.packages("keras")
if (!requireNamespace("tidyverse", quietly = TRUE)) install.packages("tidyverse")
if (!requireNamespace("data.table", quietly = TRUE)) install.packages("data.table")

library("keras")
library("tidyverse")
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ lubridate 1.9.3     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.1
## ✔ readr     2.1.5
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ data.table::between()  masks dplyr::between()
## ✖ dplyr::combine()       masks randomForest::combine()
## ✖ dplyr::filter()        masks stats::filter()
## ✖ data.table::first()    masks dplyr::first()
## ✖ lubridate::hour()      masks data.table::hour()
## ✖ lubridate::isoweek()   masks data.table::isoweek()
## ✖ dplyr::lag()           masks stats::lag()
## ✖ data.table::last()     masks dplyr::last()
## ✖ purrr::lift()          masks caret::lift()
## ✖ randomForest::margin() masks ggplot2::margin()
## ✖ lubridate::mday()      masks data.table::mday()
## ✖ lubridate::minute()    masks data.table::minute()
## ✖ lubridate::month()     masks data.table::month()
## ✖ lubridate::quarter()   masks data.table::quarter()
## ✖ lubridate::second()    masks data.table::second()
## ✖ purrr::transpose()     masks data.table::transpose()
## ✖ lubridate::wday()      masks data.table::wday()
## ✖ lubridate::week()      masks data.table::week()
## ✖ lubridate::yday()      masks data.table::yday()
## ✖ lubridate::year()      masks data.table::year()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library("data.table")
```

```
#Convert it to a factor, and then convert it to a numerical type to adapt to the needs of the de
ep learning model for processing binary classification problems.
data$Personal.Loan <- as.factor(data$Personal.Loan)
target_variable <- as.numeric(as.character(data$Personal.Loan))

# Select features
features <- data %>% select(-Personal.Loan) %>% as.matrix()
# Standardize features
features_scaled <- scale(features)

set.seed(123)
```

```
# Randomly divide the data set into training set and test set.
# 80% of the data is used as the training set.
# Split the training set and test set of features and target variables according to the index.
index <- sample(1:nrow(data), round(0.8 * nrow(data)))
x_train <- features_scaled[index, ]
y_train <- target_variable[index]
x_test <- features_scaled[-index, ]
y_test <- target_variable[-index]
```

```
# Build a sequential model
# One input layer and two hidden layers, each hidden layer is followed by a Dropout layer to reduce overfitting.
# Each hidden layer uses the ReLU activation function, and the last output layer uses the Sigmoid activation function for binary classification.
# Use L2 regularizer (regularizer_l2) to reduce overfitting.
model <- keras_model_sequential() %>%
  layer_dense(units = 32, activation = 'relu',
              kernel_regularizer = regularizer_l2(0.001),
              input_shape = dim(x_train)[2]) %>%
  layer_dropout(rate = 0.5) %>%
  layer_dense(units = 16, activation = 'relu',
              kernel_regularizer = regularizer_l2(0.001)) %>%
  layer_dropout(rate = 0.5) %>%
  layer_dense(units = 1, activation = 'sigmoid')

# Compile the model, set the Adam optimizer (learning rate 0.0005), the loss function is binary cross-entropy, and the performance measure is accuracy.
model %>% compile(
  optimizer = optimizer_adam(lr = 0.0005),
  loss = 'binary_crossentropy',
  metrics = c('accuracy')
)
```

```
model
```

```
## Model: "sequential"
## ________________________________________________________________________________
##  Layer (type)                       Output Shape                    Param #
## ================================================================================
##  dense_2 (Dense)                    (None, 32)                      416
##  dropout_1 (Dropout)                (None, 32)                      0
##  dense_1 (Dense)                    (None, 16)                      528
##  dropout (Dropout)                  (None, 16)                      0
##  dense (Dense)                      (None, 1)                       17
## ================================================================================
## Total params: 961 (3.75 KB)
## Trainable params: 961 (3.75 KB)
## Non-trainable params: 0 (0.00 Byte)
## ________________________________________________________________________________
```

```
# Use early stopping (callback_early_stopping) to prevent overfitting and end training early if
the loss on the validation set does not improve after a certain number of iterations.
early_stop <- callback_early_stopping(monitor = "val_loss", patience = 10)

# Training model
history <- model %>% fit(
  x_train,
  y_train,
  epochs = 50,
  batch_size = 32,
  validation_split = 0.2,
  callbacks = list(early_stop)
)
```

```
## Epoch 1/50
## 100/100 - 5s - loss: 0.7393 - accuracy: 0.6228 - val_loss: 0.4812 - val_accuracy: 0.8862 - 5
s/epoch - 54ms/step
## Epoch 2/50
## 100/100 - 1s - loss: 0.4950 - accuracy: 0.8025 - val_loss: 0.3507 - val_accuracy: 0.9175 - 1
s/epoch - 14ms/step
## Epoch 3/50
## 100/100 - 1s - loss: 0.3786 - accuracy: 0.8662 - val_loss: 0.2802 - val_accuracy: 0.9237 - 1
s/epoch - 12ms/step
## Epoch 4/50
## 100/100 - 1s - loss: 0.3005 - accuracy: 0.9047 - val_loss: 0.2408 - val_accuracy: 0.9275 - 1
s/epoch - 12ms/step
## Epoch 5/50
## 100/100 - 1s - loss: 0.2766 - accuracy: 0.9187 - val_loss: 0.2191 - val_accuracy: 0.9337 - 1
s/epoch - 12ms/step
## Epoch 6/50
## 100/100 - 1s - loss: 0.2447 - accuracy: 0.9266 - val_loss: 0.2082 - val_accuracy: 0.9350 - 1
s/epoch - 12ms/step
## Epoch 7/50
## 100/100 - 1s - loss: 0.2361 - accuracy: 0.9262 - val_loss: 0.2027 - val_accuracy: 0.9300 - 1
s/epoch - 12ms/step
## Epoch 8/50
## 100/100 - 1s - loss: 0.2180 - accuracy: 0.9297 - val_loss: 0.1968 - val_accuracy: 0.9375 - 1
s/epoch - 12ms/step
## Epoch 9/50
## 100/100 - 1s - loss: 0.2153 - accuracy: 0.9319 - val_loss: 0.1971 - val_accuracy: 0.9325 - 1
s/epoch - 12ms/step
## Epoch 10/50
## 100/100 - 1s - loss: 0.2073 - accuracy: 0.9337 - val_loss: 0.1943 - val_accuracy: 0.9287 - 1
s/epoch - 13ms/step
## Epoch 11/50
## 100/100 - 1s - loss: 0.1963 - accuracy: 0.9387 - val_loss: 0.1914 - val_accuracy: 0.9275 - 1
s/epoch - 13ms/step
## Epoch 12/50
## 100/100 - 1s - loss: 0.1991 - accuracy: 0.9350 - val_loss: 0.1895 - val_accuracy: 0.9375 - 1
s/epoch - 13ms/step
## Epoch 13/50
## 100/100 - 1s - loss: 0.1896 - accuracy: 0.9366 - val_loss: 0.1898 - val_accuracy: 0.9287 - 1
s/epoch - 13ms/step
## Epoch 14/50
## 100/100 - 1s - loss: 0.1926 - accuracy: 0.9378 - val_loss: 0.1921 - val_accuracy: 0.9262 - 1
s/epoch - 13ms/step
## Epoch 15/50
## 100/100 - 1s - loss: 0.1862 - accuracy: 0.9425 - val_loss: 0.1890 - val_accuracy: 0.9250 - 1
s/epoch - 13ms/step
## Epoch 16/50
## 100/100 - 1s - loss: 0.1932 - accuracy: 0.9372 - val_loss: 0.1886 - val_accuracy: 0.9287 - 1
s/epoch - 13ms/step
## Epoch 17/50
## 100/100 - 1s - loss: 0.1909 - accuracy: 0.9378 - val_loss: 0.1891 - val_accuracy: 0.9300 - 1
s/epoch - 13ms/step
## Epoch 18/50
## 100/100 - 1s - loss: 0.1933 - accuracy: 0.9387 - val_loss: 0.1885 - val_accuracy: 0.9287 - 1
s/epoch - 13ms/step
## Epoch 19/50
## 100/100 - 1s - loss: 0.1938 - accuracy: 0.9369 - val_loss: 0.1946 - val_accuracy: 0.9237 - 1
s/epoch - 13ms/step
## Epoch 20/50
## 100/100 - 1s - loss: 0.1880 - accuracy: 0.9409 - val_loss: 0.1878 - val_accuracy: 0.9300 - 1
s/epoch - 13ms/step
## Epoch 21/50
## 100/100 - 1s - loss: 0.1838 - accuracy: 0.9397 - val_loss: 0.1819 - val_accuracy: 0.9462 - 1
```

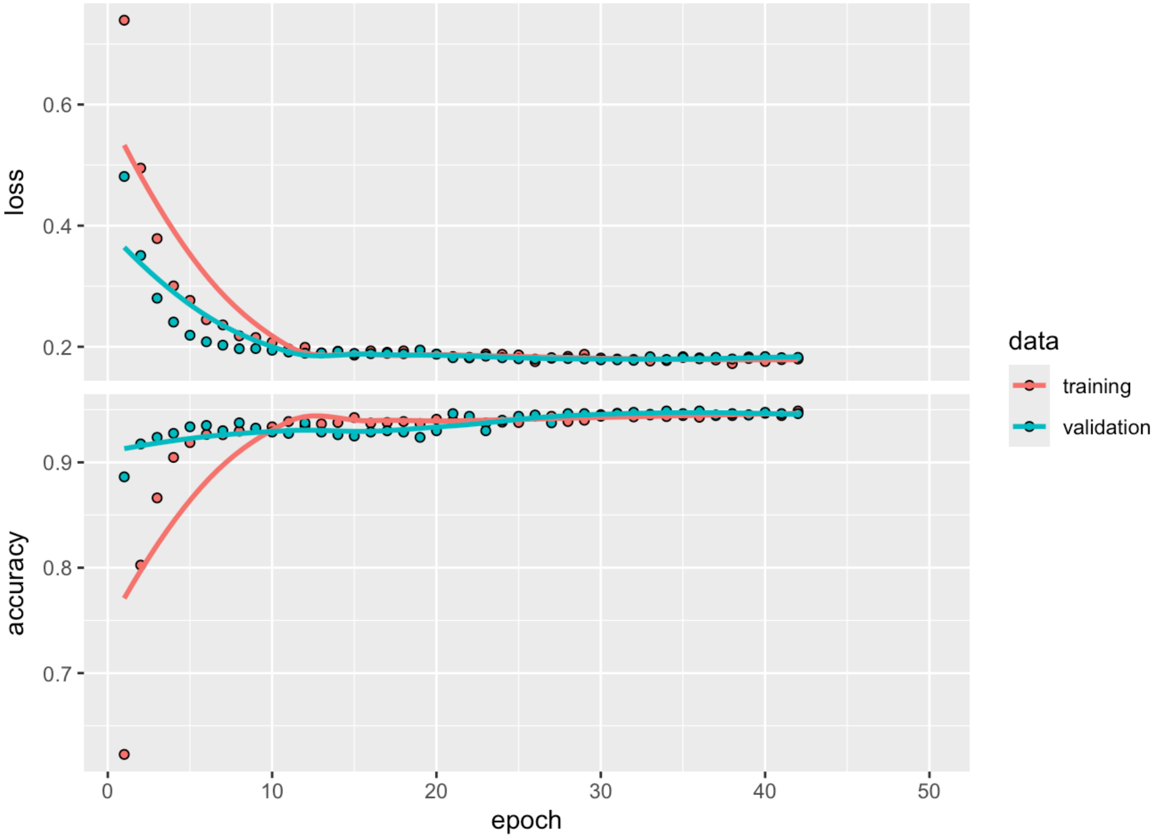
```
s/epoch - 13ms/step
## Epoch 22/50
## 100/100 - 1s - loss: 0.1816 - accuracy: 0.9416 - val_loss: 0.1826 - val_accuracy: 0.9437 - 1
s/epoch - 13ms/step
## Epoch 23/50
## 100/100 - 1s - loss: 0.1882 - accuracy: 0.9372 - val_loss: 0.1845 - val_accuracy: 0.9300 - 1
s/epoch - 13ms/step
## Epoch 24/50
## 100/100 - 1s - loss: 0.1875 - accuracy: 0.9381 - val_loss: 0.1819 - val_accuracy: 0.9400 - 1
s/epoch - 13ms/step
## Epoch 25/50
## 100/100 - 1s - loss: 0.1865 - accuracy: 0.9378 - val_loss: 0.1801 - val_accuracy: 0.9437 - 1
s/epoch - 13ms/step
## Epoch 26/50
## 100/100 - 1s - loss: 0.1754 - accuracy: 0.9441 - val_loss: 0.1795 - val_accuracy: 0.9450 - 1
s/epoch - 13ms/step
## Epoch 27/50
## 100/100 - 1s - loss: 0.1810 - accuracy: 0.9437 - val_loss: 0.1820 - val_accuracy: 0.9375 - 1
s/epoch - 13ms/step
## Epoch 28/50
## 100/100 - 1s - loss: 0.1841 - accuracy: 0.9387 - val_loss: 0.1805 - val_accuracy: 0.9462 - 1
s/epoch - 13ms/step
## Epoch 29/50
## 100/100 - 1s - loss: 0.1876 - accuracy: 0.9400 - val_loss: 0.1799 - val_accuracy: 0.9462 - 1
s/epoch - 13ms/step
## Epoch 30/50
## 100/100 - 1s - loss: 0.1815 - accuracy: 0.9453 - val_loss: 0.1784 - val_accuracy: 0.9437 - 1
s/epoch - 13ms/step
## Epoch 31/50
## 100/100 - 1s - loss: 0.1797 - accuracy: 0.9466 - val_loss: 0.1784 - val_accuracy: 0.9462 - 1
s/epoch - 13ms/step
## Epoch 32/50
## 100/100 - 1s - loss: 0.1791 - accuracy: 0.9431 - val_loss: 0.1776 - val_accuracy: 0.9475 - 1
s/epoch - 13ms/step
## Epoch 33/50
## 100/100 - 1s - loss: 0.1765 - accuracy: 0.9456 - val_loss: 0.1836 - val_accuracy: 0.9450 - 1
s/epoch - 13ms/step
## Epoch 34/50
## 100/100 - 1s - loss: 0.1772 - accuracy: 0.9434 - val_loss: 0.1791 - val_accuracy: 0.9487 - 1
s/epoch - 13ms/step
## Epoch 35/50
## 100/100 - 1s - loss: 0.1837 - accuracy: 0.9444 - val_loss: 0.1823 - val_accuracy: 0.9462 - 1
s/epoch - 13ms/step
## Epoch 36/50
## 100/100 - 1s - loss: 0.1818 - accuracy: 0.9428 - val_loss: 0.1801 - val_accuracy: 0.9487 - 1
s/epoch - 13ms/step
## Epoch 37/50
## 100/100 - 1s - loss: 0.1782 - accuracy: 0.9444 - val_loss: 0.1826 - val_accuracy: 0.9450 - 1
s/epoch - 13ms/step
## Epoch 38/50
## 100/100 - 1s - loss: 0.1722 - accuracy: 0.9444 - val_loss: 0.1798 - val_accuracy: 0.9462 - 1
s/epoch - 13ms/step
## Epoch 39/50
## 100/100 - 1s - loss: 0.1832 - accuracy: 0.9450 - val_loss: 0.1806 - val_accuracy: 0.9450 - 1
s/epoch - 13ms/step
## Epoch 40/50
## 100/100 - 1s - loss: 0.1756 - accuracy: 0.9466 - val_loss: 0.1838 - val_accuracy: 0.9475 - 1
s/epoch - 13ms/step
## Epoch 41/50
## 100/100 - 1s - loss: 0.1788 - accuracy: 0.9444 - val_loss: 0.1820 - val_accuracy: 0.9462 - 1
s/epoch - 13ms/step
## Epoch 42/50
## 100/100 - 1s - loss: 0.1798 - accuracy: 0.9487 - val_loss: 0.1825 - val_accuracy: 0.9462 - 1
s/epoch - 13ms/step
```

```
# Evaluate model performance and obtain loss values and accuracy.
model %>% evaluate(x_test, y_test)
```

```
## 32/32 - 1s - loss: 0.1569 - accuracy: 0.9550 - 513ms/epoch - 16ms/step
```

```
##      loss  accuracy
## 0.1569153 0.9550000
```

```
# Draw training history objects to show the loss and accuracy changes during training and verifi
cation, helping to analyze the learning process of the model, such as whether it is overfitting
or underfitting.
plot(history)
```


loss
accuracy
epoch
data
training
validation