Lab2_Final_2

Luke Profio

2024-04-22

Setup

Load libraries

```
library(ranger)
library(tidyverse)
library(tidymodels)
library(DataExplorer)
library(forcats)
library(conflicted)
library(vip)
conflicted::conflicts_prefer(yardstick::spec)
```

Read CSV

```
df <- read_csv("scholastic_travel.csv", show_col_types = FALSE) |>
  mutate(across(where(is.character), as.factor))
```

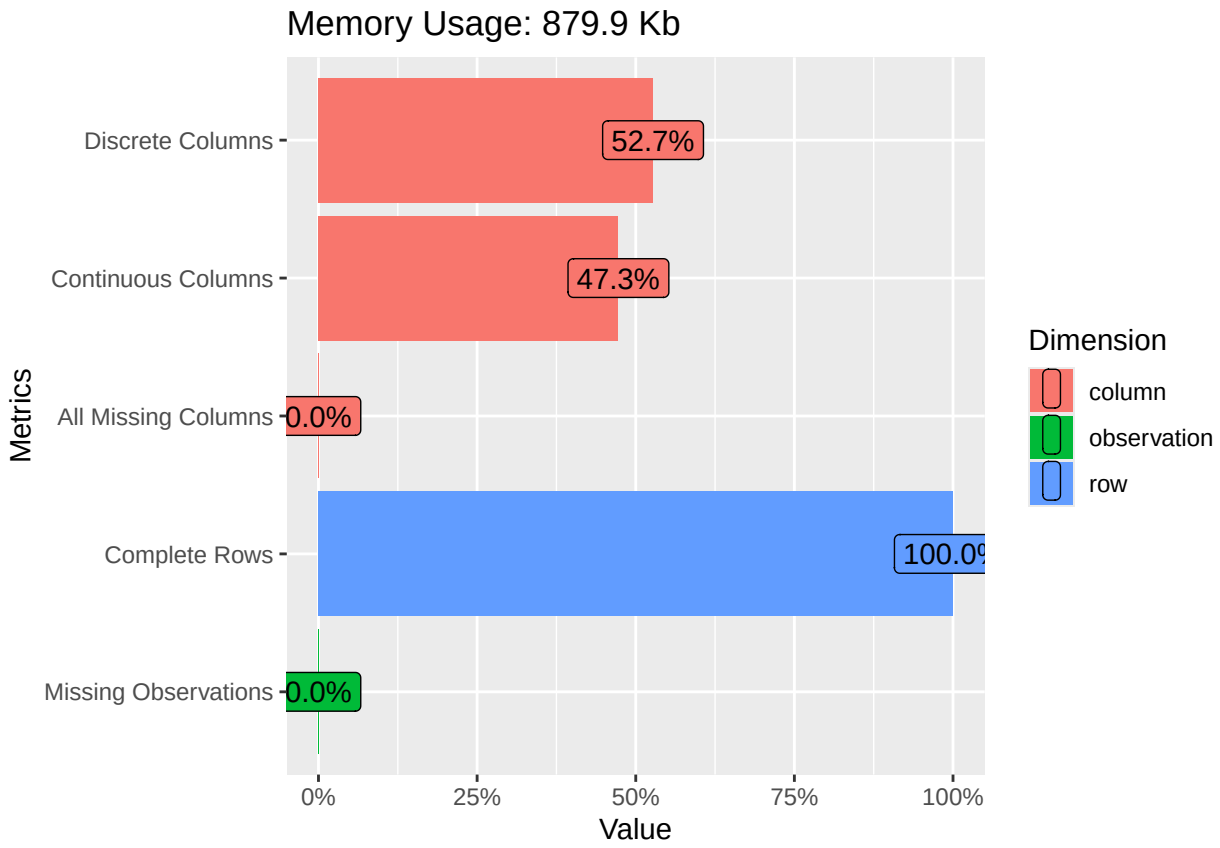
Set global variables to be used throughout the analysis

```
custom_metrics <- metric_set(accuracy, sens, spec, precision, recall)
set.seed(42)
```

Data Exploration

Missing values

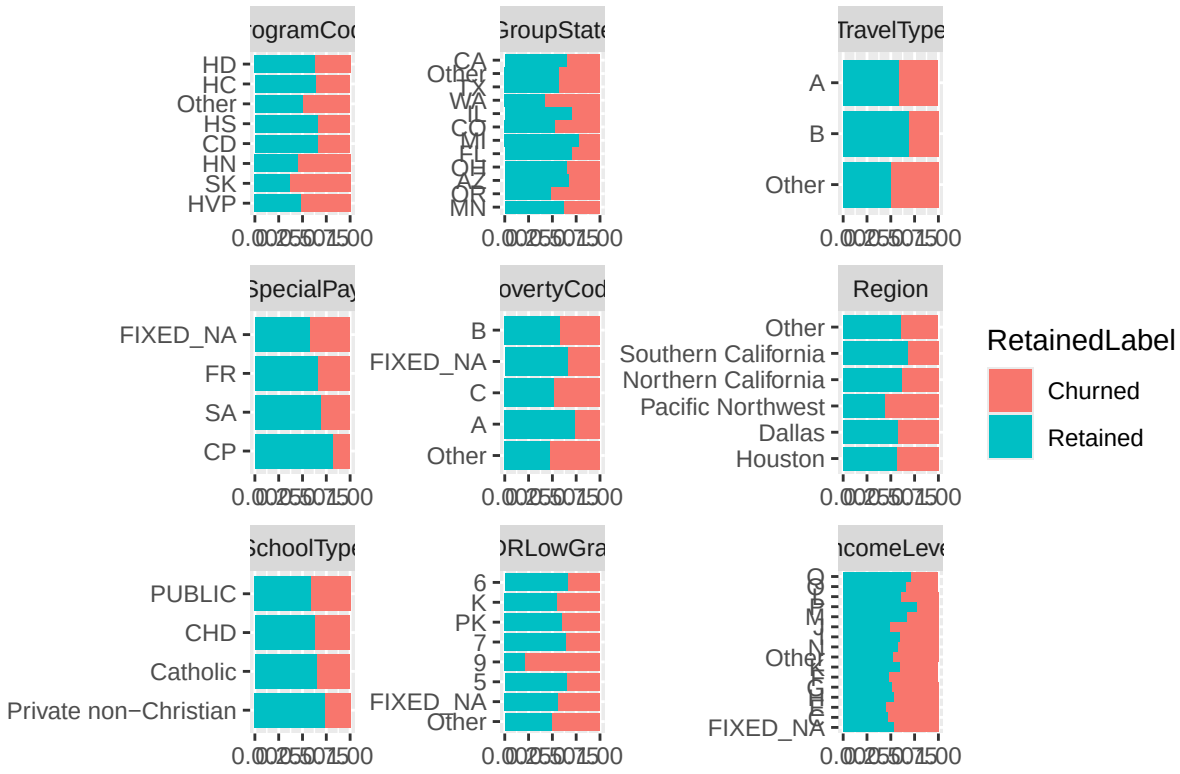
```
plot_intro(df)
```



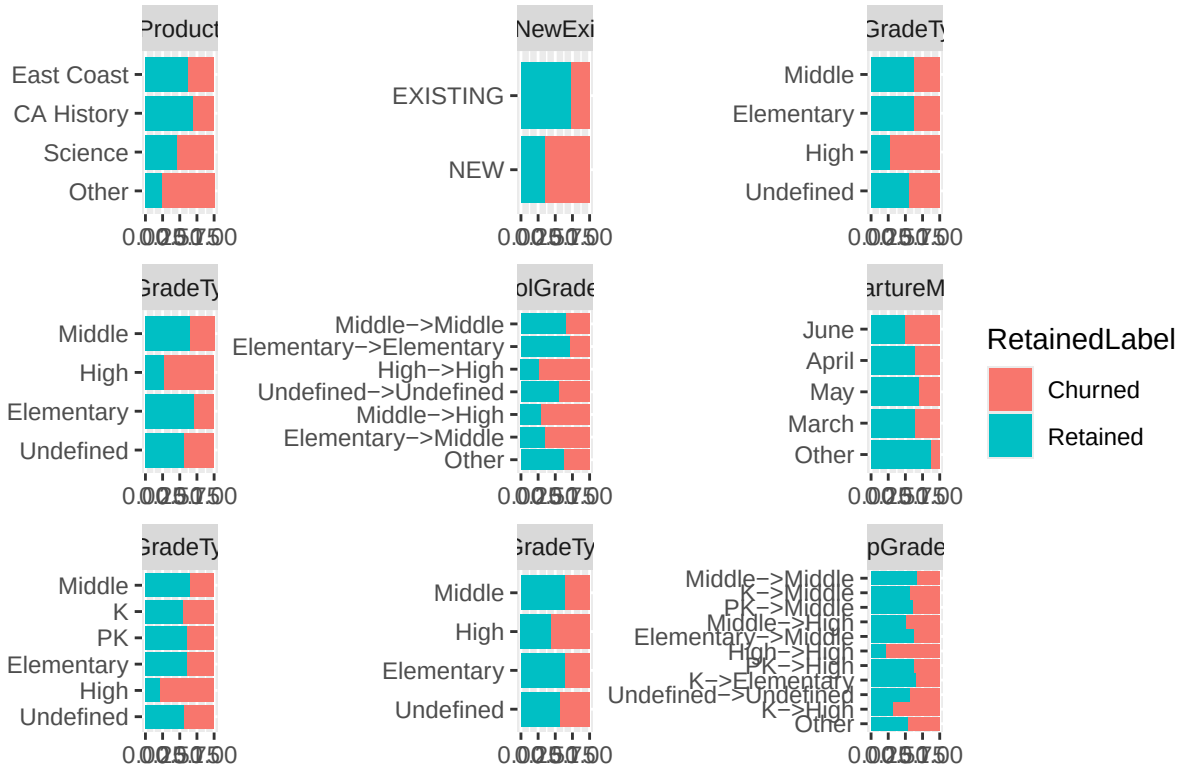
Retention by Categorical Variables

```
plot_bar(df, by = "RetainedLabel")
```

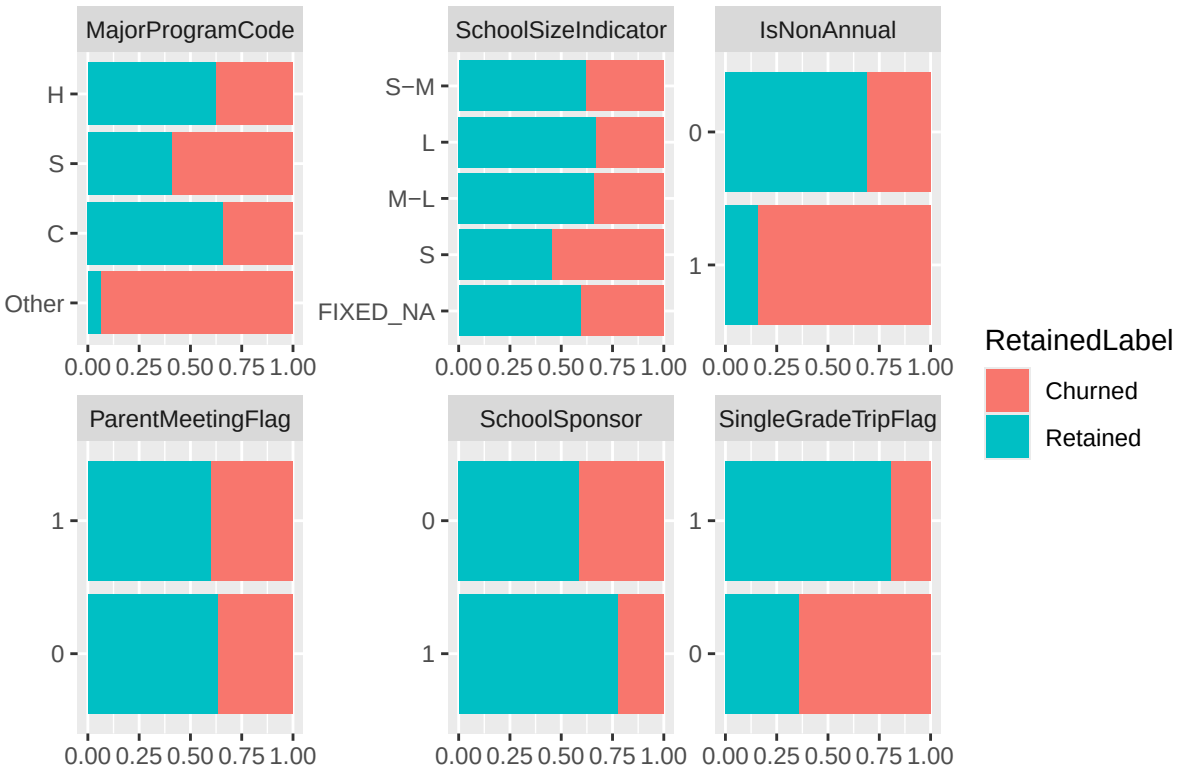
```
## 8 columns ignored with more than 50 categories.
## DepartureDate: 144 categories
## ReturnDate: 143 categories
## DepositDate: 135 categories
## EarlyRPL: 142 categories
## LatestRPL: 216 categories
## InitialSystemDate: 297 categories
## FirstMeeting: 208 categories
## LastMeeting: 173 categories
```


ogramCod
HD
HC
Other
HS
CD
HN
SK
HVP
GroupState
CA
Other
TX
WA
IL
CO
MI
FL
OH
AZ
OR
MN
TravelType
A
B
Other
SpecialPay
FIXED_NA
FR
SA
CP
overtyCod
B
FIXED_NA
C
A
Other
Region
Other
Southern California
Northern California
Pacific Northwest
Dallas
Houston
RetainedLabel
Churned
Retained
SchoolType
PUBLIC
CHD
Catholic
Private non−Christian
RLowGra
6
K
PK
7
9
5
FIXED_NA
Other
ncomeLeve
Other
FIXED_NA
0.00
0.25
0.50
0.75
1.00

Page 1


Product
East Coast
CA History
Science
Other
NewExi
EXISTING
NEW
GradeTy
Middle
Elementary
High
Undefined
GradeTy
Middle
High
Elementary
Undefined
olGrade
Middle−>Middle
Elementary−>Elementary
High−>High
Undefined−>Undefined
Middle−>High
Elementary−>Middle
Other
artureM
June
April
May
March
Other
RetainedLabel
Churned
Retained
GradeTy
Middle
K
PK
Elementary
High
Undefined
GradeTy
Middle
High
Elementary
Undefined
pGrade
Middle−>Middle
K−>Middle
PK−>Middle
Middle−>High
Elementary−>Middle
High−>High
PK−>High
K−>Elementary
Undefined−>Undefined
K−>High
Other
0.00
0.25
0.50
0.75
1.00

Page 2

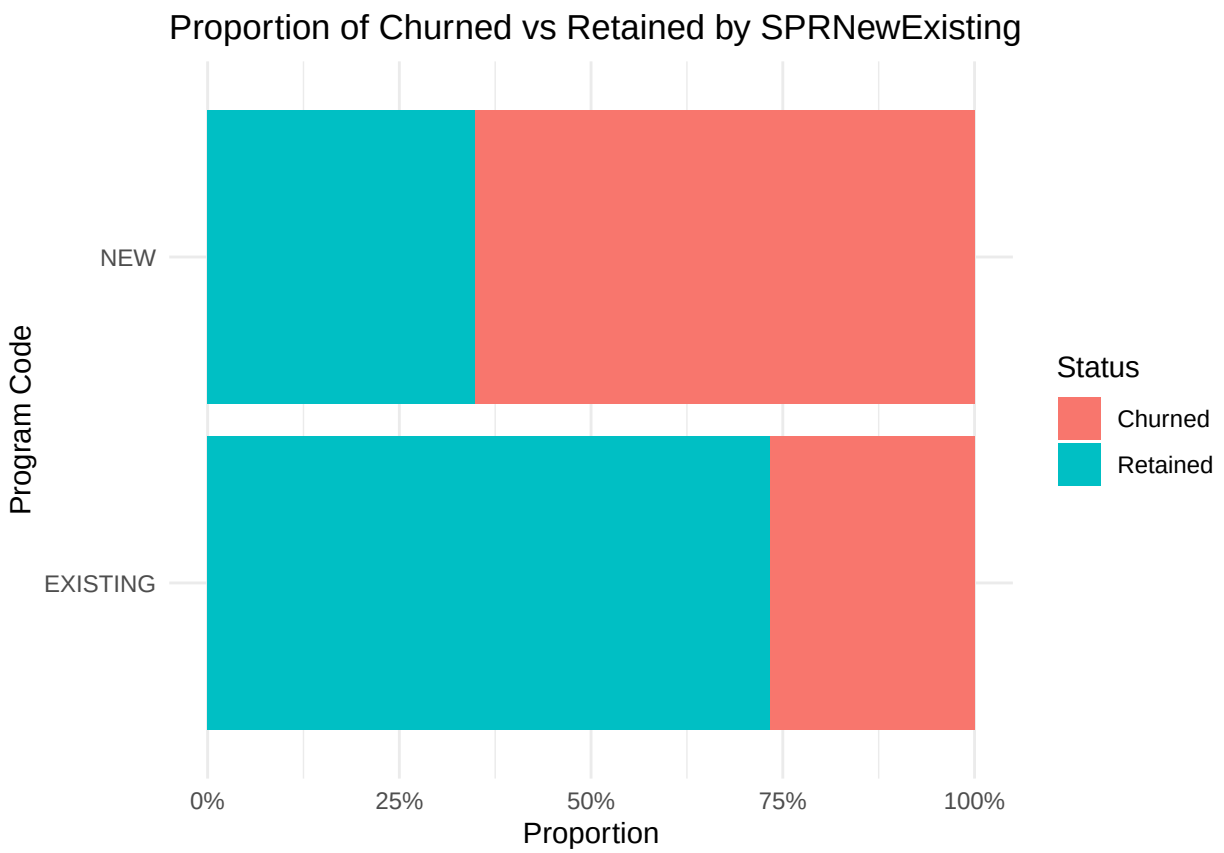


Retention by Categorical Variables (Zoomed In)

```
df_proportions <- df |>
  group_by(SPRNewExisting, RetainedLabel) |>
  summarise(Count = n()) |>
  mutate(Proportion = Count / sum(Count)) |>
  ungroup()
```

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

```
ggplot(df_proportions, aes(x = Proportion, y = SPRNewExisting, fill = RetainedLabel)) +
  geom_bar(stat = "identity", position = "fill", orientation = "y") +
  scale_x_continuous(labels = scales::percent) +
  labs(x = "Proportion", y = "Program Code", fill = "Status") +
  ggtitle("Proportion of Churned vs Retained by SPRNewExisting") +
  theme_minimal()
```

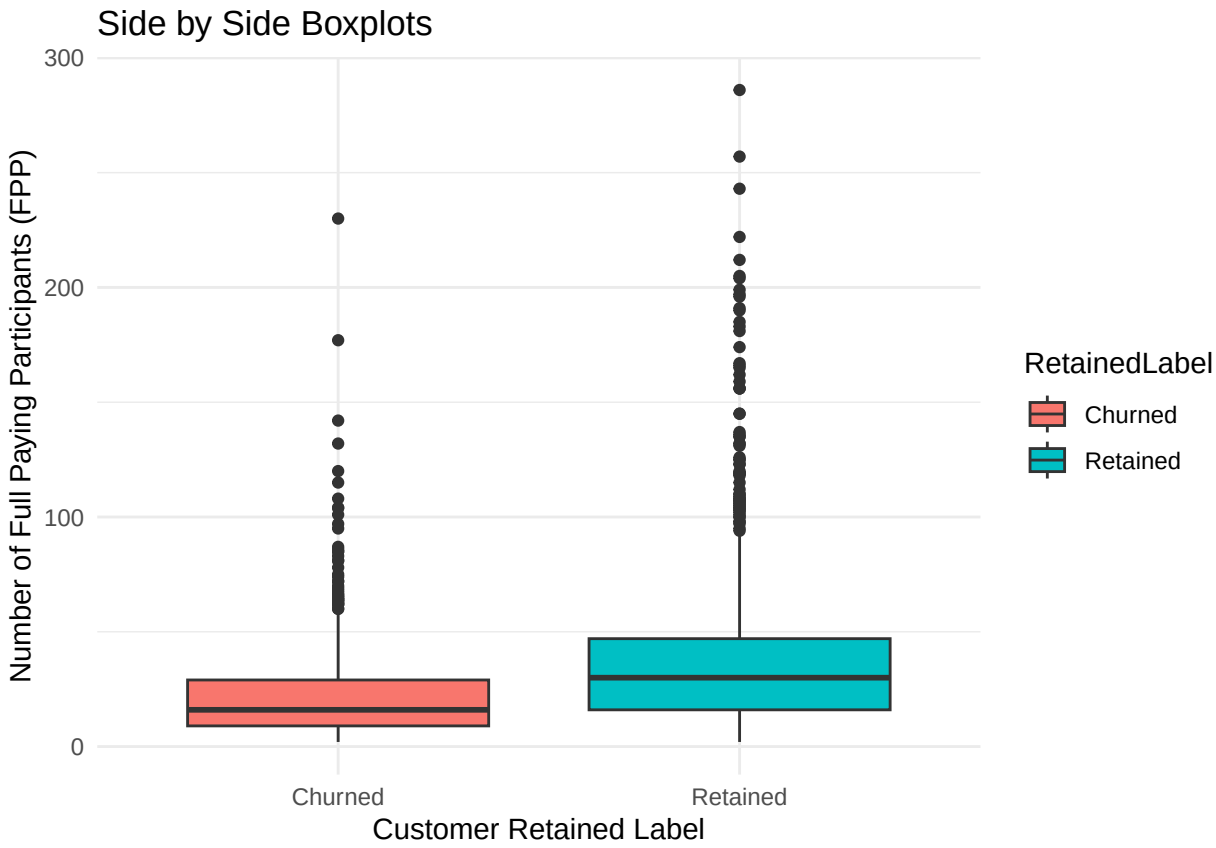



```
df_proportions
```

```
## # A tibble: 4 x 4
##   SPRNewExisting RetainedLabel Count Proportion
##   <fct>          <fct>         <int>      <dbl>
## 1 EXISTING       Churned         429      0.267
## 2 EXISTING       Retained       1178      0.733
## 3 NEW            Churned         509      0.651
## 4 NEW            Retained        273      0.349
```

Retention by Numeric Variables

```
ggplot(df, aes(x = RetainedLabel, y = FPP, fill = RetainedLabel)) +
  geom_boxplot() +
  labs(title = "Side by Side Boxplots", x = "Customer Retained Label", y = "Number of Full Paying Parti
  theme_minimal()
```




```
df |>
  group_by(RetainedLabel) |>
  summarise(count=n(), across(where(is.numeric), mean))
```

```
## # A tibble: 2 x 28
##   RetainedLabel count FromGrade ToGrade IsNonAnnual  Days Tuition FRPActive
##   <fct>         <dbl>     <dbl>   <dbl>       <dbl> <dbl>   <dbl>     <dbl>
## 1 Churned         938      6.69    7.74      0.329   4.66   1711.      11.6
## 2 Retained       1451      7.01    7.21      0.0407  4.52   1554.      20.3
## # i 20 more variables: FRPCancelled <dbl>, FRPTakeuppercent <dbl>,
## #   CancelledPax <dbl>, TotalDiscountPax <dbl>, CRMSegment <dbl>,
## #   ParentMeetingFlag <dbl>, MDRHighGrade <dbl>, TotalSchoolEnrollment <dbl>,
## #   EZPayTakeUpRate <dbl>, SchoolSponsor <dbl>, FPP <dbl>, TotalPax <dbl>,
## #   SPRGroupRevenue <dbl>, NumberOfMeetingswithParents <dbl>,
## #   DifferenceTraveltoFirstMeeting <dbl>, DifferenceTraveltoLastMeeting <dbl>,
## #   SingleGradeTripFlag <dbl>, FPPtoSchoolenrollment <dbl>, FPPtoPAX <dbl>, ...
```

Data Modeling

Split Data for Training and Testing

```
split <- initial_split(df,
                       prop = 0.80,
                       strata = RetainedLabel)

train <- training(split)
test <- testing(split)
```

K-Nearest Neighbor

```
library(kknn)
library(caret)
```

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

```
knn_recipe <-
  recipe(RetainedLabel ~ FromGrade + IsNonAnnual + Tuition + ToGrade + TotalSchoolEnrollment + FPP + Sc
  step_normalize(all_numeric_predictors()) |>
  step_dummy(all_nominal(), -all_outcomes())

knn_model <-
  nearest_neighbor(weight_func = "cos", neighbors = 20, dist_power = 2) |>
  set_engine("kknn") |>
  set_mode("classification")

knn_workflow <-
  workflow() |>
  add_model(knn_model) |>
  add_recipe(knn_recipe) |>
  fit(data = train)
```

Evaluate the Model

```
pred_class <- predict(knn_workflow,
                      new_data = test,
                      type = "class")

pred_probability <- predict(knn_workflow,
                      new_data = test,
                      type = "prob")

knn_results <- test |>
  bind_cols(pred_class, pred_probability)

custom_metrics(knn_results,
               truth = RetainedLabel,
               estimate = .pred_class)
```

```
## # A tibble: 5 x 3
##   .metric   .estimator .estimate
##   <chr>     <chr>          <dbl>
## 1 accuracy  binary         0.802
## 2 sens      binary         0.702
## 3 spec      binary         0.866
## 4 precision binary         0.772
## 5 recall    binary         0.702
```

Confusion Matrix

```
conf_mat(knn_results, truth = RetainedLabel,
         estimate = .pred_class)
```

```
##            Truth
## Prediction Churned Retained
##   Churned      132       39
##   Retained      56      252
```

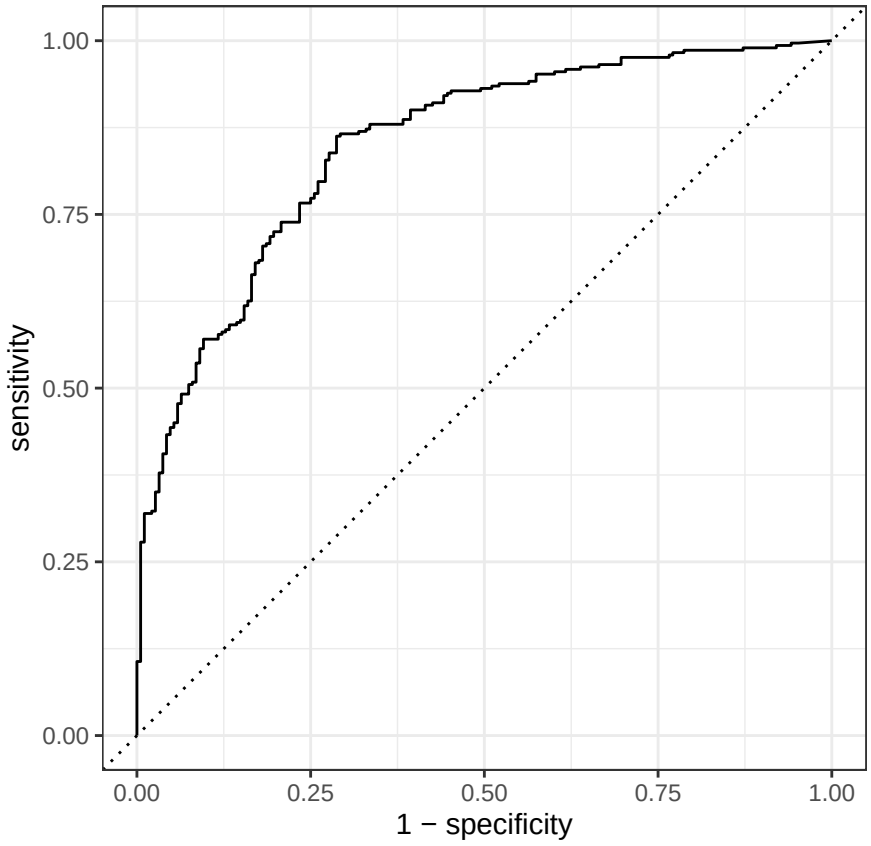
ROC Area Under the Curve

```
roc_auc(knn_results,
        truth = RetainedLabel,
        .pred_Retained,
        event_level = "second")
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.849
```

ROC Curve Visual

```
knn_results |>
  roc_curve(truth = RetainedLabel, .pred_Retained, event_level = "second") |>
  autoplot()
```



Random Forest

```
cv_set <- vfold_cv(train, strata = RetainedLabel, v = 5)

rf_recipe <-
  recipe(RetainedLabel ~ SPRNewExisting + FromGrade + InitialSystemDate + FRPActive + LatestRPL + FromG
```

```
rf_model <-
  rand_forest(mtry = tune(), min_n = tune(), trees = tune()) |>
  set_engine("ranger") |>
  set_mode("classification")

rf_workflow <-
  workflow() |>
  add_model(rf_model) |>
  add_recipe(rf_recipe)

rf_res <-
  rf_workflow |>
  tune_grid(cv_set,
            grid = 25,
            control = control_grid(),
            metrics = metric_set(accuracy))
```

```
## i Creating pre-processing data to finalize unknown parameter: mtry
```

```
rf_res |>
  show_best(metric = "accuracy")
```

```
## # A tibble: 5 x 9
##    mtry trees min_n .metric  .estimator  mean     n std_err .config
##   <int> <int> <int> <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1     3  1093    32 accuracy binary     0.797     5 0.00710 Preprocessor1_Model~
## 2     4  1752    27 accuracy binary     0.797     5 0.00565 Preprocessor1_Model~
## 3    10   829    28 accuracy binary     0.797     5 0.00634 Preprocessor1_Model~
## 4     7   212    30 accuracy binary     0.797     5 0.00665 Preprocessor1_Model~
## 5     4   648    36 accuracy binary     0.795     5 0.00577 Preprocessor1_Model~
```

Use the best parameters from the Random Forest model tuning to train the model

```
rf_model <-
  rand_forest(mtry = 3, min_n = 32, trees = 1093 ) |>
  set_engine("ranger", importance = 'impurity') |>
  set_mode("classification")

fit_workflow <-
  workflow() |>
  add_model(rf_model) |>
  add_recipe(rf_recipe) |>
  fit(data = train)


pred_class <- predict(fit_workflow,
                      new_data = test,
                      type = "class")
```

```
pred_probability <- predict(fit_workflow,
                      new_data = test,
                      type = "prob")

rf_results <- test |>
  bind_cols(pred_class, pred_probability)


custom_metrics(rf_results,
               truth = RetainedLabel,
               estimate = .pred_class)
```

```
## # A tibble: 5 x 3
##   .metric   .estimator .estimate
##   <chr>     <chr>          <dbl>
## 1 accuracy  binary         0.818
## 2 sens      binary         0.734
## 3 spec      binary         0.873
## 4 precision binary         0.789
## 5 recall    binary         0.734
```

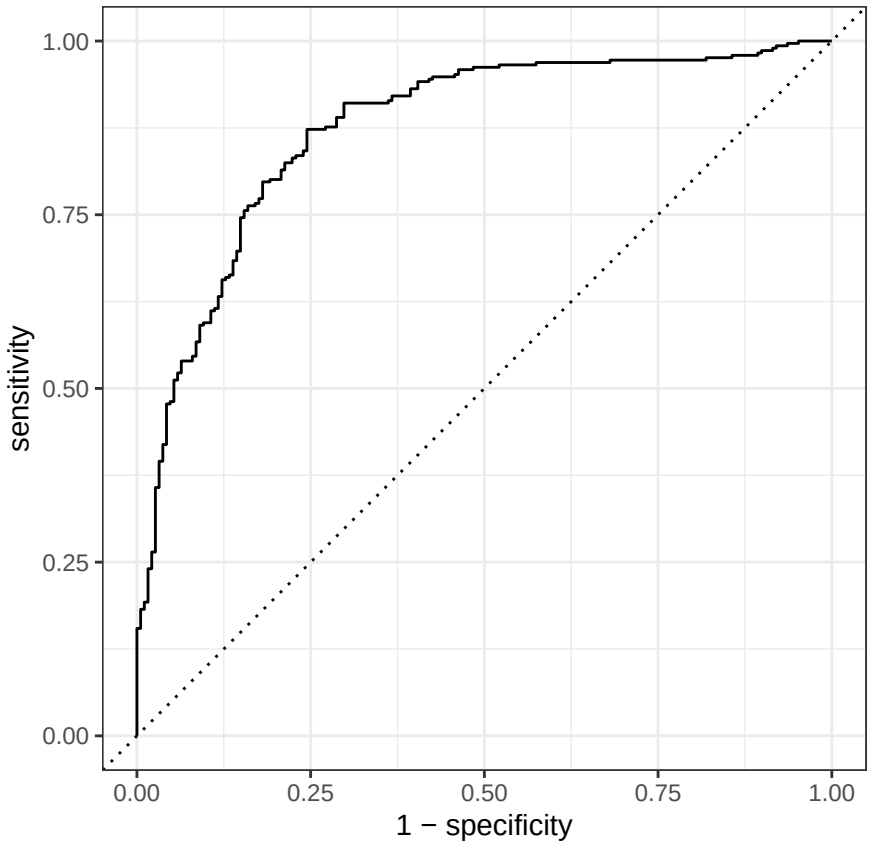
ROC Area Under the Curve

```
roc_auc(rf_results,
        truth = RetainedLabel,
        .pred_Retained,
        event_level = "second")
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.874
```

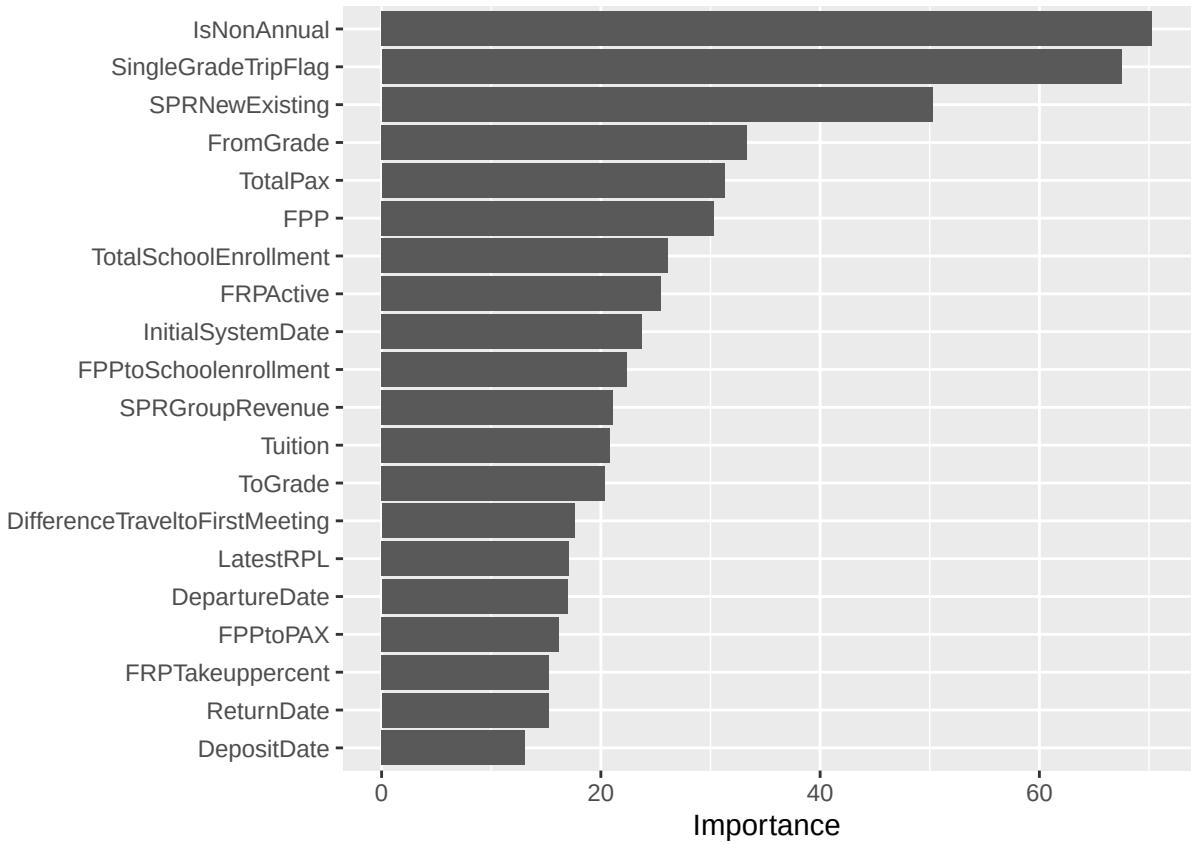
ROC Curve Visual

```
rf_results |>
  roc_curve(truth = RetainedLabel, .pred_Retained, event_level = "second") |>
  autoplot()
```



Variable Importance Plot

```
fit_workflow |>
  extract_fit_parsnip() |>
  vip(num_features = 20)
```




```
model_data <- read_csv("scholastic_travel_predictions.csv", show_col_types = FALSE)
```

```
pred_class <- predict(fit_workflow,
                      new_data = model_data,
                      type = "class")

model_data <- model_data |>
  mutate(RetainedLabel = pred_class$.pred_class)

write_csv(model_data, "model_data.csv")
```