Assignment_3_Fox

Amanda Fox

2025-04-13

Introduction

This document contains all code for Assignment 3. It extends Assignment 2 by training two support vector machine (SVM) classifiers using both linear and radial basis function (RBF) kernels. Precision is prioritized as the primary evaluation metric reflecting the need to avoid false positives due to high cost of telemarketing calls, but all models are evaluated on multiple metrics, which are stored in a matrix for comparison.

Load libraries

```
library(tidyverse)
library(caret)
library(MLmetrics)
library(kernlab)
library(ggplot2)
library(pROC)
library(doParallel)
```

Load data

First the metrics from Assignment 2 are loaded and stored as a data frame for later comparison to the SVM models.

The balanced train and test datasets from Assignment 2 are then loaded and character columns are converted to factors, with the target variable Y explicitly ordered.

```
# Import matrix of metrics for each algorithm tested in Assignment 2 for comparison
df_metrics <- read_csv("https://raw.githubusercontent.com/AmandaSFox/DATA622/refs/heads/main/Assignment_
df_metrics
```

```
## # A tibble: 7 x 6
##   Model                   Accuracy Precision Recall    F1   AUC
##   <chr>                      <dbl>     <dbl>  <dbl> <dbl> <dbl>
## 1 1 Tree Unbalanced Y        0.898     0.903  0.992 0.945 0.683
## 2 2 Tree Balanced Y          0.854     0.833  0.887 0.859 0.894
## 3 3 Random Forest Default    0.897     0.892  0.904 0.898 0.960
## 4 4 Random Forest Tuned      0.897     0.887  0.912 0.899 0.960
## 5 5 Adaboost Tuned           0.892     0.873  0.917 0.894 0.956
## 6 6 xgBoost Tuned            0.894     0.875  0.920 0.897 0.958
## 7 7 xgBoost Tuned 2          0.896     0.876  0.924 0.899 0.960
```

```r
# Import balanced dataset and test/train split
df_bal <- read_csv("https://raw.githubusercontent.com/AmandaSFox/DATA622/refs/heads/main/Assignment_3/d
df_bal_test <- read_csv("https://raw.githubusercontent.com/AmandaSFox/DATA622/refs/heads/main/Assignmen
df_bal_train <- read_csv("https://raw.githubusercontent.com/AmandaSFox/DATA622/refs/heads/main/Assignme

# Factors: convert all char columns in train and test datasets to factors
df_bal_train <- df_bal_train %>%
  mutate(across(where(is.character), as.factor))

df_bal_test <- df_bal_test %>%
  mutate(across(where(is.character), as.factor))

# Explicitly set Y as a binary factor with controlled levels
df_bal_train$Y <- factor(df_bal_train$Y, levels = c("no", "yes"))
df_bal_test$Y  <- factor(df_bal_test$Y, levels = c("no", "yes"))

# STOP if factor levels do not match between train and test
stopifnot(all(names(df_bal_train) == names(df_bal_test)))

glimpse(df_bal_train)
```

```
## Rows: 28,833
## Columns: 16
## $ Age                          <dbl> 50.98393, 42.51871, 34.59952, 39.66418, ~
## $ Occupation                   <fct> retired, services, admin., blue-collar, ~
## $ Marital_Status               <fct> married, married, single, married, singl~
## $ Contact_Type                 <fct> cellular, cellular, cellular, telephone,~
## $ Month                        <fct> aug, may, jul, may, jul, nov, jun, may, ~
## $ Day                          <fct> mon, thu, thu, wed, mon, wed, mon, fri, ~
## $ Contacts_This_Campaign       <dbl> 0.57636387, 7.34684029, 0.05170396, 0.04~
## $ Days_Since_Last_Campaign     <dbl> 1011.1844, 985.2942, 992.9217, 1004.1043~
## $ Contacts_Before_This_Campaign <dbl> 0.034471843, 0.913504187, 0.214885597, -~
## $ Previous_Outcome             <fct> nonexistent, failure, nonexistent, nonex~
## $ cpi                          <dbl> 93.53622, 92.82878, 94.09862, 93.92475, ~
## $ cci                          <dbl> -36.55729, -42.17078, -44.16873, -35.017~
## $ euribor3m                    <dbl> 5.6326824, 1.6714123, 4.5837682, 4.53677~
## $ Y                            <fct> no, no, no, no, no, no, no, no, no, no, ~
## $ Education2                   <fct> less.than.hs, high.school, less.than.hs,~
## $ Loan_Profile                 <fct> No Loans - No Default, No Loans - No Def~
```

```r
glimpse(df_bal_test)
```

```
## Rows: 12,355
## Columns: 16
## $ Age                          <dbl> 25.68035, 35.05131, 29.16287, 40.83483, ~
## $ Occupation                   <fct> technician, admin., blue-collar, technic~
## $ Marital_Status               <fct> married, married, married, divorced, mar~
## $ Contact_Type                 <fct> telephone, telephone, telephone, telepho~
## $ Month                        <fct> may, may, jun, jul, aug, nov, mar, may, ~
## $ Day                          <fct> fri, mon, wed, wed, tue, wed, mon, tue, ~
## $ Contacts_This_Campaign       <dbl> 0.9885882, 1.6605531, 0.8857296, -0.1233~
## $ Days_Since_Last_Campaign     <dbl> 1034.51903, 1018.82153, 921.21337, 978.3~
## $ Contacts_Before_This_Campaign <dbl> -0.010129017, -0.223039509, 0.023908149,~
## $ Previous_Outcome              <fct> nonexistent, nonexistent, nonexistent, n~
## $ cpi                           <dbl> 94.06994, 93.97805, 94.32545, 93.91046, ~
## $ cci                           <dbl> -35.32341, -37.90812, -42.11269, -44.230~
## $ euribor3m                     <dbl> 3.8979938, 5.0188487, 4.9771177, 4.46368~
## $ Y                             <fct> no, no, no, no, no, no, no, no, no, no, ~
## $ Education2                    <fct> university.degree, less.than.hs, less.th~
## $ Loan_Profile                  <fct> No Loans - No Default, No Loans - No Def~
```

Prepare to store metrics

A new matrix is initialized to store the metrics associated with SVMs. Parallel processing is enabled.

```
# Initialize new matrix to store new metrics
matrix_metrics <- matrix(NA, nrow = 0, ncol = 6)
colnames(matrix_metrics) <- c("Model",
                              "Accuracy",
                              "Precision",
                              "Recall",
                              "F1",
                              "AUC")
```

Start parallel processing

```
# Stop if still running from last session
if (exists("cl")) {
  try(stopCluster(cl), silent = TRUE)
  rm(cl)
}

# Start
num_cores <- detectCores() - 1 # Use one less core
cl <- makePSOCKcluster(num_cores)
registerDoParallel(cl)
```

Data preparation

SVMs are sensitive to the scale of input features. Standardization is applied based on the training set, and the transformation is applied to both datasets.

```
preproc <- preProcess(df_bal_train,
                      method = c("center", "scale"))
train_svm <- predict(preproc, df_bal_train)
test_svm <- predict(preproc, df_bal_test)
```

SVM Linear Kernel: Train and Test

Using the caret package, an SVM with a linear kernel is trained with 5-fold cross-validation. The cost parameter is tuned over three values with precision used as the optimization metric.

```r
set.seed(123)

# train model
svm_linear <- train(
  Y ~ .,
  data = train_svm,
  method = "svmLinear",
  trControl = trainControl(
    method = "cv",
    number = 5,
    classProbs = TRUE,
    summaryFunction = prSummary
  ),
  metric = "Precision",
  tuneGrid = expand.grid(C = c(0.1, 1, 10))
)

save(svm_linear, file = "prob_svm_linear.RData")

# Keep only rows in test set where Loan_Profile exists in training set
test_svm <- test_svm[test_svm$Loan_Profile %in% levels(train_svm$Loan_Profile), ]

# Drop unused factor levels to avoid warnings
test_svm$Loan_Profile <- droplevels(test_svm$Loan_Profile)

# Predict on test set
pred_svm_linear <- predict(svm_linear, newdata = test_svm)
prob_svm_linear <- predict(svm_linear, newdata = test_svm, type = "prob")[, 2]

# Confusion matrix and AUC
cm_svm_linear <- confusionMatrix(pred_svm_linear, test_svm$Y)
auc_svm_linear <- auc(roc(test_svm$Y, prob_svm_linear))

# add metrics to the matrix
matrix_metrics <- rbind(matrix_metrics,
                        c("8 SVM Linear",
                          cm_svm_linear$overall["Accuracy"],
                          cm_svm_linear$byClass["Precision"],
                          cm_svm_linear$byClass["Recall"],
                          cm_svm_linear$byClass["F1"],
                          auc_svm_linear))
```

SVM RBF Kernel: Train and Test

An SVM with a radial basis function (RBF) kernel is also trained with 5-fold cross-validation.

The cost and sigma parameters are limited due to memory issues to two values for c (1, 10) and the median estimated sigma value (.51). Precision was again used as the optimization metric.

```r
# Set the same seed for all SVM models to ensure consistent cross-validation folds
set.seed(123)

# estimate sigma values: sigma = 1/(2*gamma) in caret
```

```r
sigest(Y ~ ., data = train_svm)

set.seed(123)

# Train on median value for sigma and two values for C due to memory issues
svm_radial <- train(
  Y ~ .,
  data = train_svm,
  method = "svmRadial",
  trControl = trainControl(
    method = "cv",
    number = 5,
    classProbs = TRUE,
    summaryFunction = prSummary,
    verboseIter = TRUE
  ),
  metric = "Precision",
  tuneGrid = expand.grid(
    C = c(1, 10),
    sigma = c(0.05139891)
  )
)

save(svm_radial, file = "prob_svm_radial.RData")

# Predict on test set
pred_svm_radial <- predict(svm_radial, newdata = test_svm)
prob_svm_radial <- predict(svm_radial, newdata = test_svm, type = "prob")[, 2]

# Confusion matrix and AUC
cm_svm_radial <- confusionMatrix(pred_svm_radial, test_svm$Y)
auc_svm_radial <- auc(roc(test_svm$Y, prob_svm_radial))

# add metrics to the matrix
matrix_metrics <- rbind(matrix_metrics,
                        c("9 SVM Radial",
                          cm_svm_radial$overall["Accuracy"],
                          cm_svm_radial$byClass["Precision"],
                          cm_svm_radial$byClass["Recall"],
                          cm_svm_radial$byClass["F1"],
                          auc_svm_radial))

# convert to df and change to numeric values
df_svm_metrics <- as.data.frame(matrix_metrics)

df_svm_metrics <- df_svm_metrics %>%
  mutate(
    Accuracy = as.numeric(Accuracy),
    Precision = as.numeric(Precision),
    Recall = as.numeric(Recall),
    F1 = as.numeric(F1),
    AUC = as.numeric(AUC)
  )
```

Compare Model Performance

The results of the two SVM models and all models from Assignment 2 are compared in a table and bar chart.

```
# Import SVM metrics from earlier sessions
df_svm_metrics <- read_csv("https://raw.githubusercontent.com/AmandaSFox/DATA622/refs/heads/main/Assign
df_svm_metrics
```

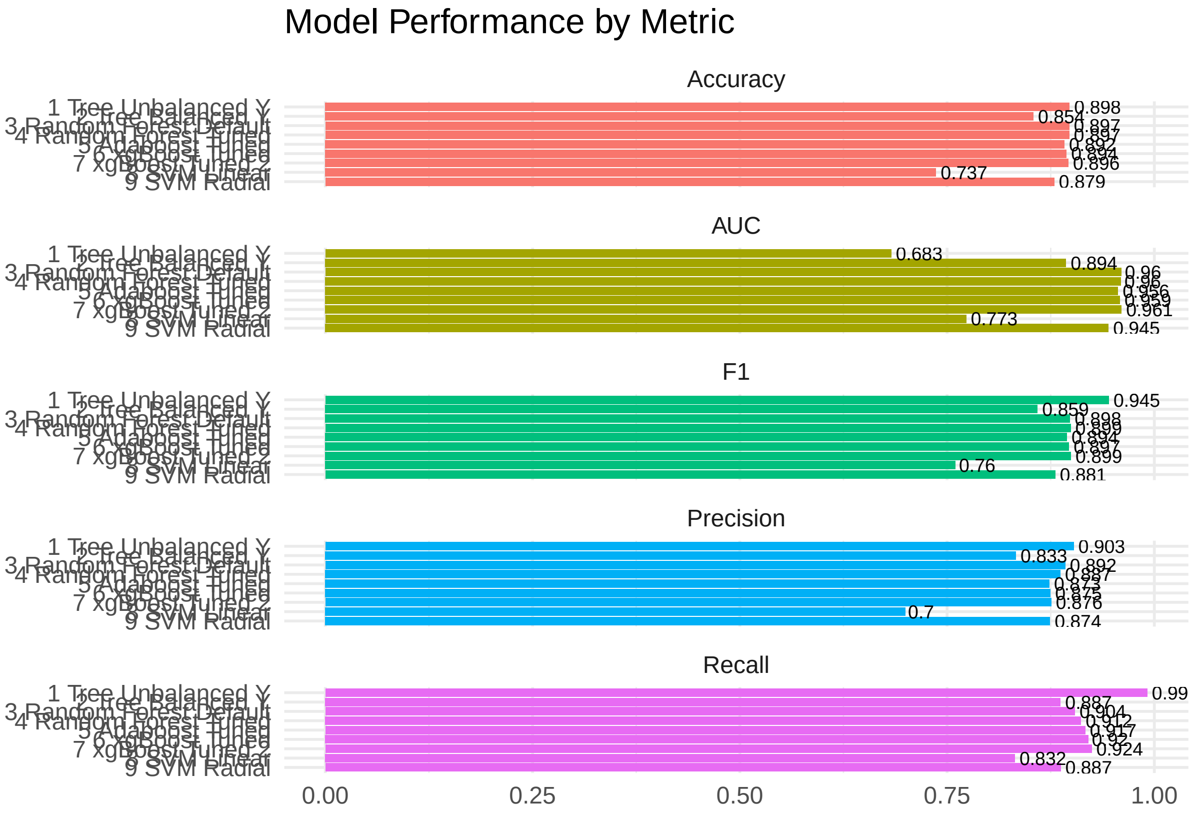
```
## # A tibble: 2 x 6
##   Model        Accuracy Precision Recall    F1   AUC
##   <chr>           <dbl>     <dbl>  <dbl> <dbl> <dbl>
## 1 8 SVM Linear    0.737     0.700  0.832 0.760 0.773
## 2 9 SVM Radial    0.879     0.874  0.887 0.881 0.945
```

```
# Combine with metrics from Assignment 2 to create all metrics table
df_all_metrics <- bind_rows(df_metrics, df_svm_metrics)
df_all_metrics
```

```
## # A tibble: 9 x 6
##   Model                   Accuracy Precision Recall    F1   AUC
##   <chr>                      <dbl>     <dbl>  <dbl> <dbl> <dbl>
## 1 1 Tree Unbalanced Y        0.898     0.903  0.992 0.945 0.683
## 2 2 Tree Balanced Y          0.854     0.833  0.887 0.859 0.894
## 3 3 Random Forest Default    0.897     0.892  0.904 0.898 0.960
## 4 4 Random Forest Tuned      0.897     0.887  0.912 0.899 0.960
## 5 5 Adaboost Tuned           0.892     0.873  0.917 0.894 0.956
## 6 6 xgBoost Tuned            0.894     0.875  0.920 0.897 0.958
## 7 7 xgBoost Tuned 2          0.896     0.876  0.924 0.899 0.960
## 8 8 SVM Linear               0.737     0.700  0.832 0.760 0.773
## 9 9 SVM Radial               0.879     0.874  0.887 0.881 0.945
```

```
# Pivot for ggplot
df_long <- df_all_metrics %>%
  pivot_longer(cols = c(-Model),
               names_to = "Metric")

# Create comparison bar chart
plot_compare <- df_long %>%
  ggplot(aes(x = reorder(Model, desc(Model)),
             y = value,
             fill = Metric)) +
  geom_bar(stat = "identity",
           show.legend = FALSE) +
  geom_text(aes(label = round(value, 3)),
            hjust = -0.1, size = 2.5) +
  facet_wrap(~ Metric,
             scales = "free_y",
             ncol = 1) +
  labs(title = "Model Performance by Metric",
       x = "",
       y = "")+
  coord_flip() +
  theme_minimal()
plot_compare
```



Cleanup

Parallel processing is stopped

```
stopCluster(cl)
registerDoSEQ()
```