Data Science Final Project

MODEL 1B: SVM

Kjay O. Coca

2022-12-14

```
# Helper packages
library(dplyr)    # for data wrangling
```

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

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

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

```
library(ggplot2)  # for awesome graphics
library(rsample)  # for data splitting
library(readr)

# Modeling packages
library(caret)    # for classification and regression training
```

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

```
library(kernlab)  # for fitting SVMs
```

```
##
## Attaching package: 'kernlab'

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

```
library(modeldata) #for Failure.binary data
library(forcats)

# Model interpretability packages
library(pdp)      # for partial dependence plots, etc.
library(vip)      # for variable importance plots
```

```
##
## Attaching package: 'vip'

## The following object is masked from 'package:utils':
##
##     vi
```

```
set.seed(123)
radiomics_data <- read_csv("D:/1 MASTERS/STAT225/FINAL PROJECT/STAT 325 _FINAL PROJECT_/Normalize Radio
```

```
## Rows: 197 Columns: 431
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (1): Institution
## dbl (430): Failure.binary, Failure, Entropy_cooc.W.ADC, GLNU_align.H.PET, Mi...
##
## 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.
```

Load Failure.binary data

```
radiomics_data$Failure.binary=as.factor(radiomics_data$Failure.binary)
```

SPLITTING FOR TRAINING AND TESTING

```
set.seed(123)  # for reproducibility
for_split <- initial_split(radiomics_data, prop = 0.8, strata = "Failure.binary")
radiomicsdata_train <- training(for_split)
radiomicsdata_test  <- testing(for_split)
```

In this case, I set 80 percent for training data and 20 percent for testing data. There are 39 observation for testing and 158 observation for training and both have 413 variables.

Linear

```
caret::getModelInfo("svmLinear")$svmLinear$parameters
```

```
##   parameter   class label
## 1         C numeric  Cost
```

Polynomial kernel

```
caret::getModelInfo("svmPoly")$svmPoly$parameters
```

```
##   parameter   class             label
## 1    degree numeric Polynomial Degree
## 2     scale numeric             Scale
## 3         C numeric              Cost
```

Radial basis kernel

```
caret::getModelInfo("svmRadial")$svmRadial$parameters
```

```
##   parameter   class label
## 1     sigma numeric Sigma
## 2         C numeric  Cost
```

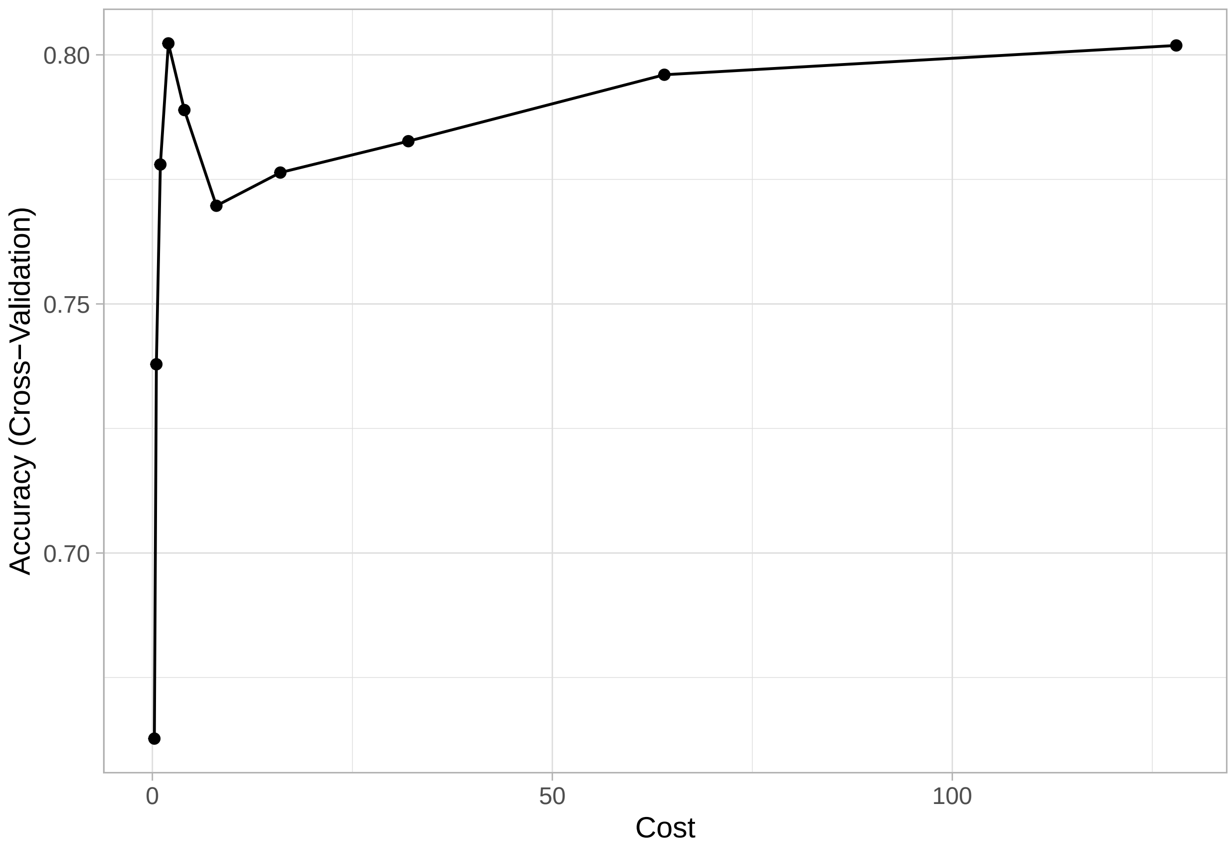
Run SVM Model in Training phase

```
set.seed(1854)
svm_split <- train(
  Failure.binary ~ .,
  data = radiomicsdata_train,
  method = "svmRadial",
  preProcess = c("center", "scale"),
  trControl = trainControl(method = "cv", number = 10),
  tuneLength = 10
)
```

Using `split_train()` function, tuning SVM model with radial basis kernel. Using `split_train()` function, we can tune an SVM model with radial basis kernel.

Plot results

```
ggplot(svm_split) + theme_light()
```



Print results

```
svm_split$results
```

```
##          sigma      C  Accuracy     Kappa AccuracySD   KappaSD
## 1  0.001998749   0.25 0.6627451 0.0000000 0.01891300 0.0000000
## 2  0.001998749   0.50 0.7378922 0.2715440 0.06418046 0.2198366
## 3  0.001998749   1.00 0.7779902 0.4565954 0.07142465 0.1608304
## 4  0.001998749   2.00 0.8023039 0.5196491 0.09057479 0.2186000
## 5  0.001998749   4.00 0.7889216 0.5030643 0.07639949 0.1942976
## 6  0.001998749   8.00 0.7697059 0.4653629 0.07092559 0.1830668
## 7  0.001998749  16.00 0.7763725 0.4861127 0.06283611 0.1498343
## 8  0.001998749  32.00 0.7826716 0.4985015 0.07602914 0.1806382
## 9  0.001998749  64.00 0.7960049 0.5248585 0.07147503 0.1670975
## 10 0.001998749 128.00 0.8018873 0.5429164 0.08701199 0.2010434
```

Control parameters for SVM

```
class.weights = c("No" = 1, "Yes" = 10)
ctrl <- trainControl(
  method = "cv",
  number = 10,
  classProbs = TRUE,
  summaryFunction = twoClassSummary
)

radiomicsdata_train$Failure.binary <- fct_recode(radiomicsdata_train$Failure.binary,No="0",Yes="1")
```

Print the AUC values during Training

```
# Tune an SVM
set.seed(123)  # for reproducibility
train_svm_auc <- train(
  Failure.binary ~ .,
  data = radiomicsdata_train,
  method = "svmRadial",
  preProcess = c("center", "scale"),
  metric = "ROC",  # area under ROC curve (AUC)
  trControl = ctrl,
  tuneLength = 10
)

# Print results
train_svm_auc$results
```

```
##          sigma      C       ROC      Sens      Spec     ROCSD     SensSD
## 1  0.001769054   0.25 0.8054545 0.8636364 0.5233333 0.1315912 0.13483997
## 2  0.001769054   0.50 0.8034545 0.8536364 0.5400000 0.1328426 0.12708861
## 3  0.001769054   1.00 0.8197879 0.8927273 0.5400000 0.1282824 0.09984377
## 4  0.001769054   2.00 0.8577576 0.9200000 0.5933333 0.1013829 0.11352924
## 5  0.001769054   4.00 0.8736061 0.9509091 0.6100000 0.1005268 0.05181730
## 6  0.001769054   8.00 0.8756061 0.9118182 0.6700000 0.1099644 0.10816526
## 7  0.001769054  16.00 0.8607879 0.8718182 0.6866667 0.1190518 0.13281234
## 8  0.001769054  32.00 0.8623636 0.9018182 0.6666667 0.1077055 0.10363459
## 9  0.001769054  64.00 0.8652424 0.9118182 0.6500000 0.1014671 0.09735246
## 10 0.001769054 128.00 0.8743939 0.9027273 0.6500000 0.1000568 0.07937427
##       SpecSD
## 1  0.3344611
## 2  0.3184841
## 3  0.2734327
## 4  0.2688797
## 5  0.2183270
## 6  0.2710064
## 7  0.2644351
## 8  0.2449490
## 9  0.2625845
## 10 0.2625845
```

```
confusionMatrix(train_svm_auc)
```

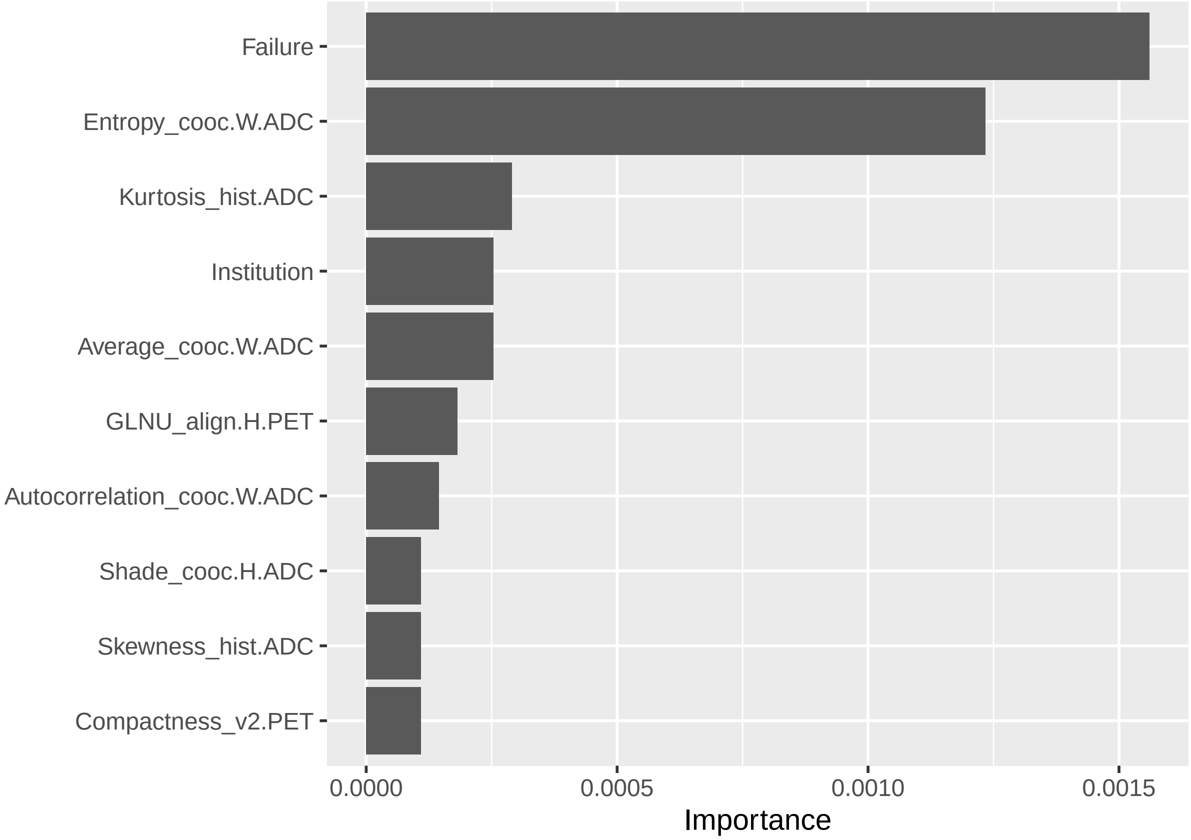
```
## Cross-Validated (10 fold) Confusion Matrix
```

```
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction   No  Yes
##        No  60.5 11.5
##        Yes  5.7 22.3
##
##  Accuracy (average) : 0.828
```

The average accuracy of the trained model is 0.828 or 82.8 percent.

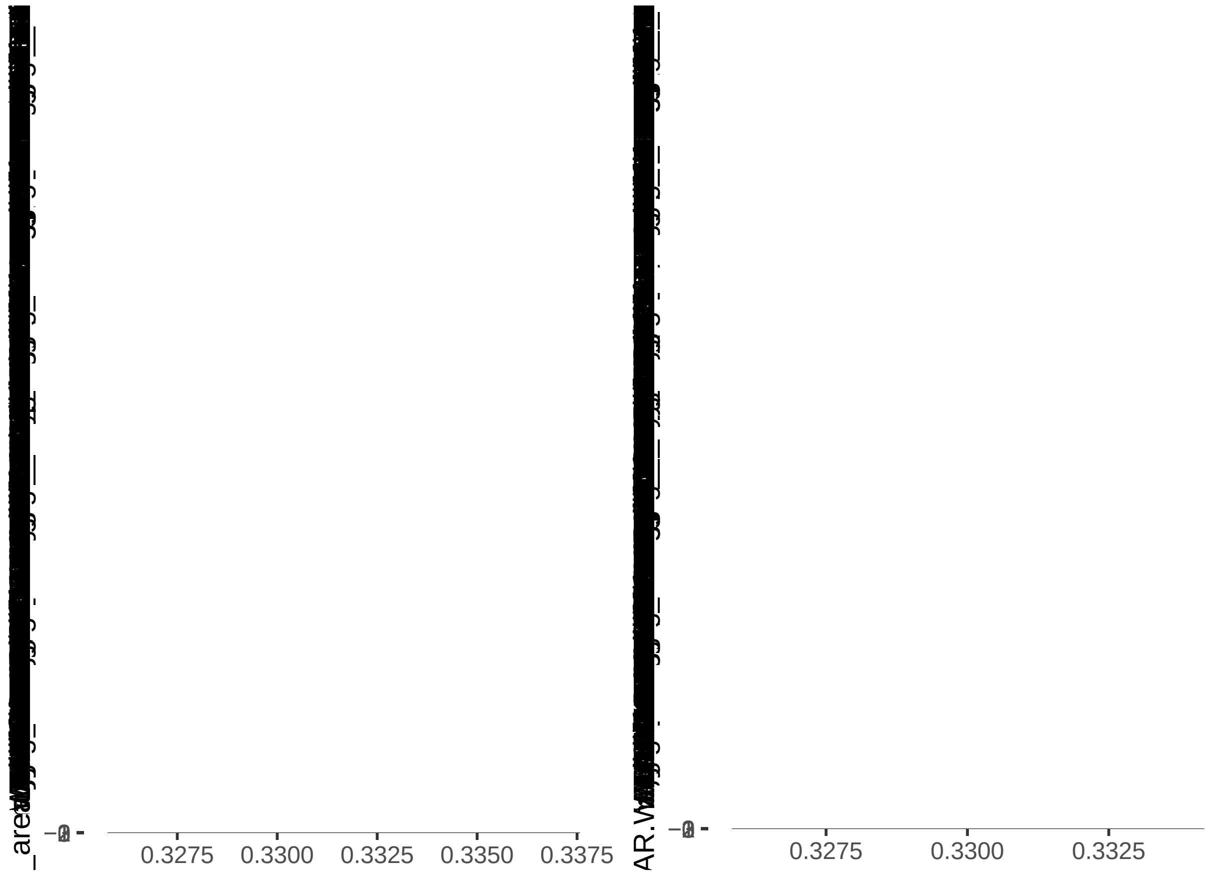
Print the Top 20 important features during Training

```
prob_yes <- function(object, newdata) {
  predict(object, newdata = newdata, type = "prob")[, "Yes"]
}

# Variable importance plot
set.seed(2827)  # for reproducibility
vip(train_svm_auc, method = "permute", nsim = 5, train = radiomicsdata_train,
    target = "Failure.binary", metric = "auc", reference_class = "Yes",
    pred_wrapper = prob_yes)
```



This are the top 20 important variable during Training. Failure variable is the most important. And next is Entrophy_cooc.W.ADC.

```
features <- setdiff(names(radiomics_data), names(radiomics_data)[c(1,2)])
pdps <- lapply(features, function(x) {
  partial(train_svm_auc, pred.var = x, which.class = 2,
          prob = TRUE, plot = TRUE, plot.engine = "ggplot2") +
    coord_flip()
})

grid.arrange(grobs = pdps,  ncol = 2)
```



Print the AUC values during Testing

```
radiomicsdata_test$Failure.binary=fct_recode(radiomicsdata_test$Failure.binary,No="0",Yes="1")

# Tune an SVM with radial
set.seed(5628)  # for reproducibility
test_svm_auc <- train(
  Failure.binary ~ .,
  data = radiomicsdata_test,
  method = "svmRadial",
  preProcess = c("center", "scale"),
  metric = "ROC",  # area under ROC curve (AUC)
  trControl = ctrl,
  tuneLength = 10
)

# Print results
test_svm_auc$results
```

```
##          sigma      C       ROC      Sens Spec     ROCSD    SensSD SpecSD
## 1  0.001959001   0.25 0.6750000 0.9666667    0 0.2872013 0.1054093      0
## 2  0.001959001   0.50 0.5750000 0.9333333    0 0.3320577 0.1405457      0
## 3  0.001959001   1.00 0.6250000 1.0000000    0 0.3148829 0.0000000      0
## 4  0.001959001   2.00 0.3083333 0.9000000    0 0.3168372 0.2249829      0
## 5  0.001959001   4.00 0.3500000 0.9000000    0 0.4021547 0.2249829      0
## 6  0.001959001   8.00 0.3916667 0.9000000    0 0.3889881 0.2249829      0
## 7  0.001959001  16.00 0.3083333 0.9000000    0 0.3514740 0.2249829      0
## 8  0.001959001  32.00 0.4250000 0.8333333    0 0.3976202 0.2832789      0
## 9  0.001959001  64.00 0.3750000 0.9333333    0 0.3833937 0.1405457      0
## 10 0.001959001 128.00 0.4083333 0.8666667    0 0.3937200 0.2810913      0
```

```
confusionMatrix(test_svm_auc)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction   No  Yes
##        No  62.5 35.0
##        Yes  2.5  0.0
##
##  Accuracy (average) : 0.625
```

The accuracy of test data in this model is 0.625 or just 62.5 percent. It smaller compare to trained data.