SVM Analysis - Drug Response Prediction

2024-11-24

SVM Analysis of Drug Response Data

Data Loading and Initial Exploration

```
drug_data <- read.csv('https://raw.githubusercontent.com/Kingtilon1/MachineLearning-BigData/refs/heads/m

head(drug_data)
```

```
##   Age Sex     BP Cholesterol Na_to_K  Drug
## 1  23   F   HIGH        HIGH  25.355 drugY
## 2  47   M    LOW        HIGH  13.093 drugC
## 3  47   M    LOW        HIGH  10.114 drugC
## 4  28   F NORMAL        HIGH   7.798 drugX
## 5  61   F    LOW        HIGH  18.043 drugY
## 6  22   F NORMAL        HIGH   8.607 drugX
```

```
str(drug_data)
```

```
## 'data.frame':    200 obs. of  6 variables:
##  $ Age        : int  23 47 47 28 61 22 49 41 60 43 ...
##  $ Sex        : chr  "F" "M" "M" "F" ...
##  $ BP         : chr  "HIGH" "LOW" "LOW" "NORMAL" ...
##  $ Cholesterol: chr  "HIGH" "HIGH" "HIGH" "HIGH" ...
##  $ Na_to_K    : num  25.4 13.1 10.1 7.8 18 ...
##  $ Drug       : chr  "drugY" "drugC" "drugC" "drugX" ...
```

```
skim(drug_data)
```

Table 1: Data summary

Name	drug_data
Number of rows	200
Number of columns	6
_____	
Column type frequency:	
character	4
numeric	2
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Sex	0	1	1	1	0	2	0
BP	0	1	3	6	0	3	0
Cholesterol	0	1	4	6	0	2	0
Drug	0	1	5	5	0	5	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Age	0	1	44.31	16.54	15.00	31.00	45.00	58.00	74.00	
Na_to_K	0	1	16.08	7.22	6.27	10.45	13.94	19.38	38.25	

Data Preprocessing

```
drug_data$Sex <- as.factor(drug_data$Sex)
drug_data$BP <- as.factor(drug_data$BP)
drug_data$Cholesterol <- as.factor(drug_data$Cholesterol)
drug_data$Drug <- as.factor(drug_data$Drug)

set.seed(123)
train_index <- createDataPartition(drug_data$Drug, p = 0.8, list = FALSE)
train_data <- drug_data[train_index, ]
test_data <- drug_data[-train_index, ]

preproc <- preProcess(train_data[, c("Age", "Na_to_K")], method = c("center", "scale"))
train_data_scaled <- predict(preproc, train_data)
test_data_scaled <- predict(preproc, test_data)
```

SVM Model with Linear Kernel

```
svm_linear <- svm(Drug ~ ., data = train_data_scaled, kernel = "linear", cost = 1)

pred_linear <- predict(svm_linear, test_data_scaled)
conf_matrix_linear <- confusionMatrix(pred_linear, test_data_scaled$Drug)
print(conf_matrix_linear)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction drugA drugB drugC drugX drugY
##      drugA     4     0     0     0     0
##      drugB     0     2     0     0     0
##      drugC     0     0     3     0     0
##      drugX     0     0     0     9     0
##      drugY     0     1     0     1    18
```

```
##
## Overall Statistics
##
##                Accuracy : 0.9474
##                  95% CI : (0.8225, 0.9936)
##     No Information Rate : 0.4737
##     P-Value [Acc > NIR] : 4.248e-10
##
##                   Kappa : 0.9211
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: drugA Class: drugB Class: drugC Class: drugX
## Sensitivity                1.0000      0.66667      1.00000       0.9000
## Specificity                1.0000      1.00000      1.00000       1.0000
## Pos Pred Value             1.0000      1.00000      1.00000       1.0000
## Neg Pred Value             1.0000      0.97222      1.00000       0.9655
## Prevalence                 0.1053      0.07895      0.07895       0.2632
## Detection Rate             0.1053      0.05263      0.07895       0.2368
## Detection Prevalence       0.1053      0.05263      0.07895       0.2368
## Balanced Accuracy          1.0000      0.83333      1.00000       0.9500
##                      Class: drugY
## Sensitivity                1.0000
## Specificity                0.9000
## Pos Pred Value             0.9000
## Neg Pred Value             1.0000
## Prevalence                 0.4737
## Detection Rate             0.4737
## Detection Prevalence       0.5263
## Balanced Accuracy          0.9500
```

SVM Model with Radial Kernel

```
svm_radial <- svm(Drug ~ ., data = train_data_scaled, kernel = "radial", cost = 1)

pred_radial <- predict(svm_radial, test_data_scaled)
conf_matrix_radial <- confusionMatrix(pred_radial, test_data_scaled$Drug)
print(conf_matrix_radial)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction drugA drugB drugC drugX drugY
##      drugA     4     0     0     0     0
##      drugB     0     1     0     0     0
##      drugC     0     0     3     0     0
##      drugX     0     0     0    10     0
##      drugY     0     2     0     0    18
##
## Overall Statistics
```

```
##
##                Accuracy : 0.9474
##                  95% CI : (0.8225, 0.9936)
##     No Information Rate : 0.4737
##     P-Value [Acc > NIR] : 4.248e-10
##
##                   Kappa : 0.9205
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: drugA Class: drugB Class: drugC Class: drugX
## Sensitivity                1.0000      0.33333      1.00000       1.0000
## Specificity                1.0000      1.00000      1.00000       1.0000
## Pos Pred Value             1.0000      1.00000      1.00000       1.0000
## Neg Pred Value             1.0000      0.94595      1.00000       1.0000
## Prevalence                 0.1053      0.07895      0.07895       0.2632
## Detection Rate             0.1053      0.02632      0.07895       0.2632
## Detection Prevalence       0.1053      0.02632      0.07895       0.2632
## Balanced Accuracy          1.0000      0.66667      1.00000       1.0000
##                      Class: drugY
## Sensitivity                1.0000
## Specificity                0.9000
## Pos Pred Value             0.9000
## Neg Pred Value             1.0000
## Prevalence                 0.4737
## Detection Rate             0.4737
## Detection Prevalence       0.5263
## Balanced Accuracy          0.9500
```

Tuning SVM Parameters

```r
tuning_grid <- expand.grid(
  cost = c(0.1, 1, 10),
  gamma = c(0.1, 1, 10)
)

svm_tune <- tune.svm(
  Drug ~ .,
  data = train_data_scaled,
  kernel = "radial",
  cost = c(0.1, 1, 10),
  gamma = c(0.1, 1, 10)
)
print(svm_tune$best.parameters)
```

```
##   gamma cost
## 7   0.1   10
```

```r
svm_final <- svm(
  Drug ~ .,
  data = train_data_scaled,
  kernel = "radial",
  cost = svm_tune$best.parameters$cost,
  gamma = svm_tune$best.parameters$gamma
)

pred_final <- predict(svm_final, test_data_scaled)
conf_matrix_final <- confusionMatrix(pred_final, test_data_scaled$Drug)
print(conf_matrix_final)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction drugA drugB drugC drugX drugY
##      drugA     4     0     0     0     0
##      drugB     0     3     0     0     0
##      drugC     0     0     3     0     0
##      drugX     0     0     0    10     0
##      drugY     0     0     0     0    18
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9075, 1)
##     No Information Rate : 0.4737
##     P-Value [Acc > NIR] : 4.662e-13
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: drugA Class: drugB Class: drugC Class: drugX
## Sensitivity                1.0000      1.00000      1.00000       1.0000
## Specificity                1.0000      1.00000      1.00000       1.0000
## Pos Pred Value             1.0000      1.00000      1.00000       1.0000
## Neg Pred Value             1.0000      1.00000      1.00000       1.0000
## Prevalence                 0.1053      0.07895      0.07895       0.2632
## Detection Rate             0.1053      0.07895      0.07895       0.2632
## Detection Prevalence       0.1053      0.07895      0.07895       0.2632
## Balanced Accuracy          1.0000      1.00000      1.00000       1.0000
##                      Class: drugY
## Sensitivity                1.0000
## Specificity                1.0000
## Pos Pred Value             1.0000
## Neg Pred Value             1.0000
## Prevalence                 0.4737
## Detection Rate             0.4737
## Detection Prevalence       0.4737
## Balanced Accuracy          1.0000
```

Variable Importance Analysis

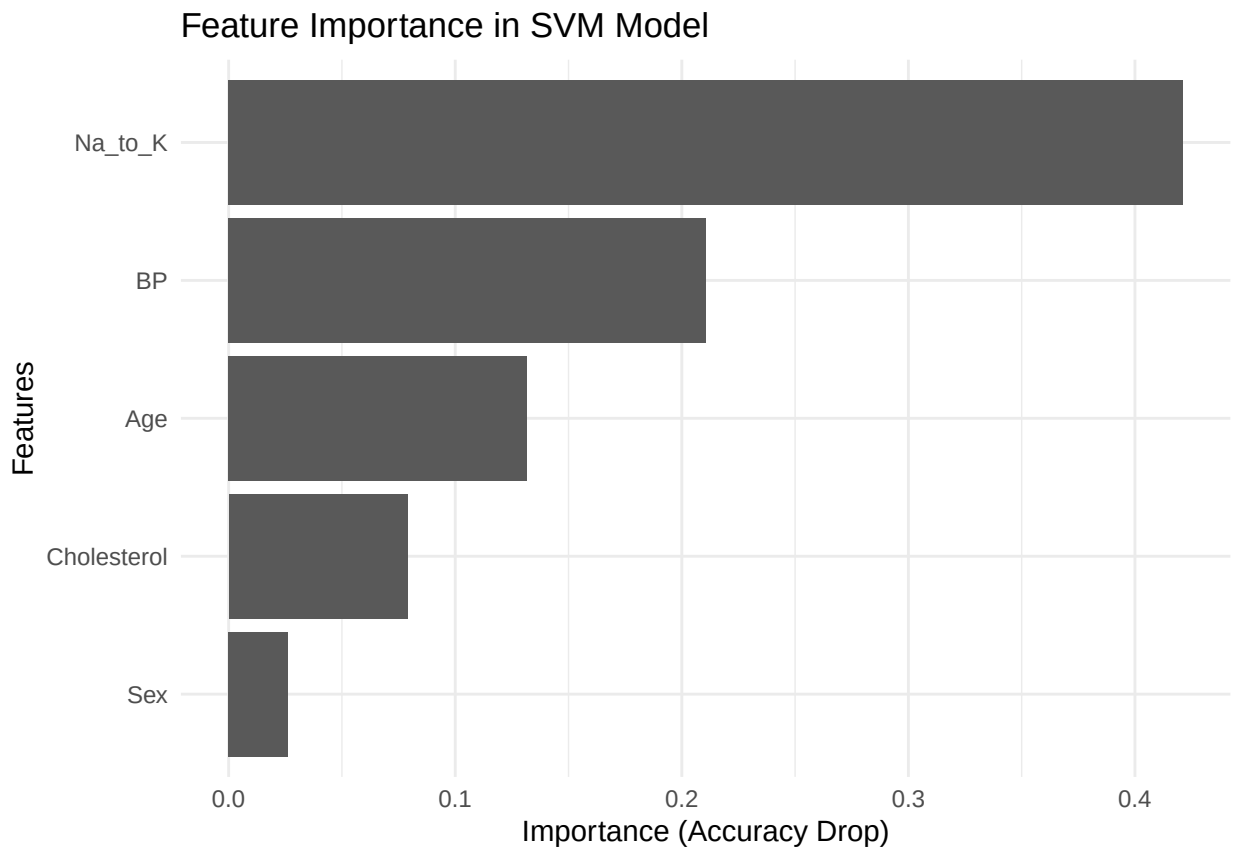
```
importance <- data.frame(
  Feature = names(train_data_scaled)[-which(names(train_data_scaled) == "Drug")],
  Importance = 0
)

for(feature in importance$Feature) {
  test_permuted <- test_data_scaled
  test_permuted[,feature] <- sample(test_permuted[,feature])

  pred_permuted <- predict(svm_final, test_permuted)

  importance$Importance[importance$Feature == feature] <-
    mean(pred_final == test_data_scaled$Drug) - mean(pred_permuted == test_data_scaled$Drug)
}

importance <- importance[order(-importance$Importance),]
ggplot(importance, aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_minimal() +
  labs(title = "Feature Importance in SVM Model",
       x = "Features",
       y = "Importance (Accuracy Drop)")
```



Comparison with Decision Tree Results

```
results_comparison <- data.frame(
  Model = c("Decision Tree", "SVM Linear", "SVM Radial", "SVM Tuned"),
  Accuracy = c(0.97,
               conf_matrix_linear$overall["Accuracy"],
               conf_matrix_radial$overall["Accuracy"],
               conf_matrix_final$overall["Accuracy"]),
  Kappa = c(0.96,
            conf_matrix_linear$overall["Kappa"],
            conf_matrix_radial$overall["Kappa"],
            conf_matrix_final$overall["Kappa"])
)

print(results_comparison)
```

```
##           Model  Accuracy     Kappa
## 1 Decision Tree 0.9700000 0.9600000
## 2    SVM Linear 0.9473684 0.9210800
## 3    SVM Radial 0.9473684 0.9205021
## 4     SVM Tuned 1.0000000 1.0000000
```