Support Vector Machine Analysis - Student Dropout Prediction

Understanding the Pathways to Academic Achievement

2025-11-23

Contents

1 Support Vector Machine Analysis

1.1 Feature Selection

The results indicate that grades and the amount of credits approved have the highest F-Values, suggesting they are the most significant predictors for predicting the success of graduating or not.

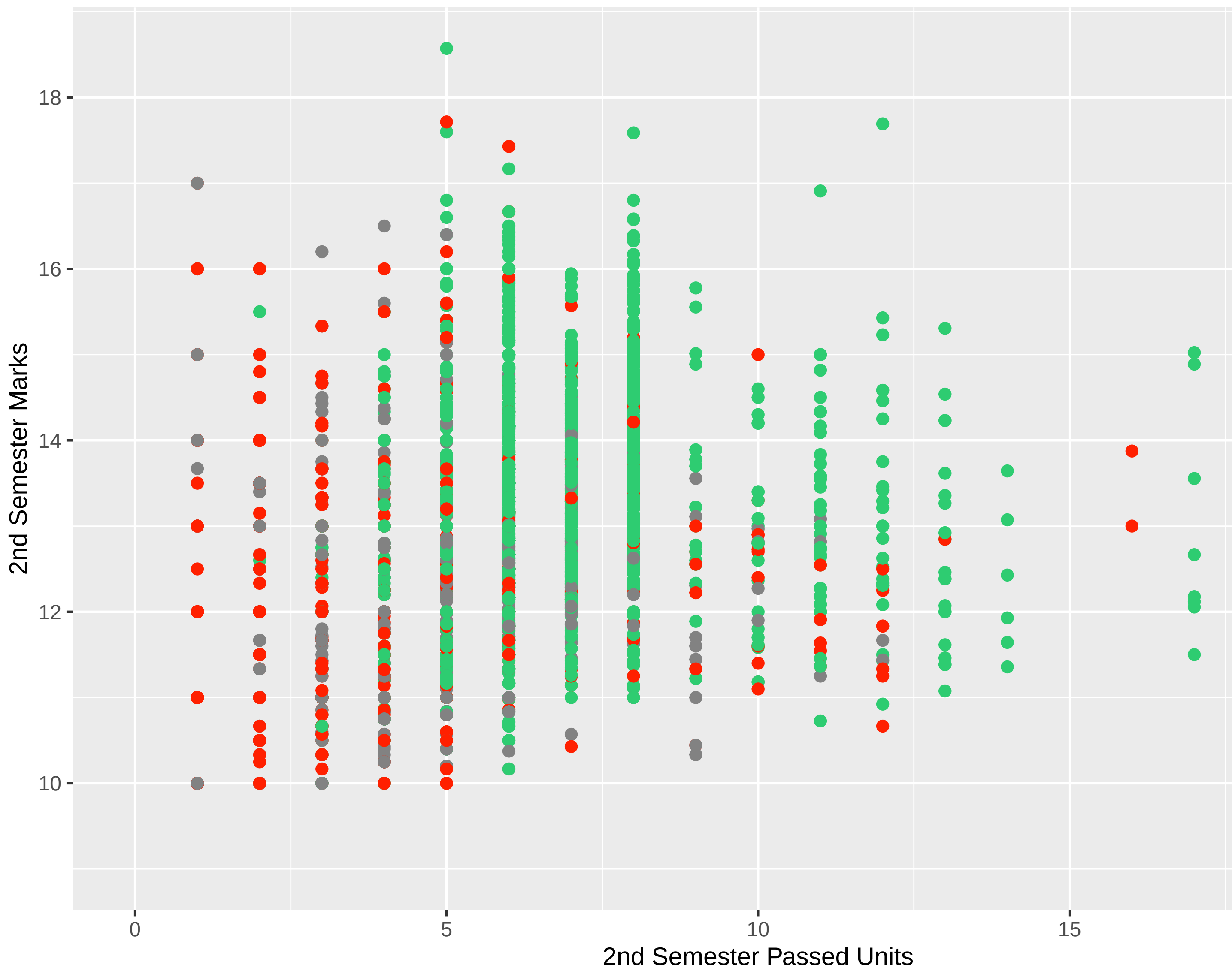
In this case, I want to choose the features of second semester marks and the amount of units passed to plot my model.

1.2 Data Loading & Preparation

```r
data <- read.csv("../../data/preprocessed_data.csv")

data$Target <- as.factor(data$Target)

# Prepare the data frame
marks <- data.frame(
  x.1 = data$Curricular.units.2nd.sem..grade.,
  x.2 = data$Curricular.units.2nd.sem..approved.,
  y = as.factor(data$Target)
)

marks <- marks[complete.cases(marks), ]
```

1.3 Exploratory Visualization

```r
# Plot
ggplot(data = marks, aes(x = x.2, y = x.1, color = y)) +
  geom_point(size = 2) +
  scale_color_manual(values = c("Dropout" = "#ff2000",
                                "Enrolled" = "#828282",
                                "Graduate" = "#2ecc71")) +
  scale_y_continuous(limits = c(9, NA)) +
  labs(
    x = "2nd Semester Passed Units",
    y = "2nd Semester Marks"
  )
```



1.4 Train/Test Split

Now, I will split the data into training and testing sets, and then fit several SVM models to classify the students based on their marks.

```
set.seed(123)
trainIndex <- createDataPartition(marks$y, p = 0.8, list = FALSE)
train_data <- marks[trainIndex, ]
test_data <- marks[-trainIndex, ]

cat("Training set size:", nrow(train_data), "\n")
```

```
## Training set size: 3541
```

```
cat("Test set size:", nrow(test_data), "\n")
```

```
## Test set size: 883
```

1.5 Model Training

Now, I will fit the SVM model with a linear, a radial and a polynomial kernel.

```
svm_linear <- svm(y ~ x.1 + x.2, data = train_data, kernel = "linear", cost = 1)
svm_radial <- svm(y ~ x.1 + x.2, data = train_data, kernel = "radial", cost = 10, gamma = 0.1)
svm_poly <- svm(y ~ x.1 + x.2, data = train_data, kernel = "polynomial", cost = 1, degree = 3)
```

1.6 Model Evaluation

Predict and compare the models

```
pred_linear <- predict(svm_linear, test_data)
pred_radial <- predict(svm_radial, test_data)
pred_poly <- predict(svm_poly, test_data)
```

1.6.1 Linear Kernel Results

```
cat("\n Linear Kernel Result:\n")
```

```
##
##  Linear Kernel Result:
```

```
cm_linear <- confusionMatrix(pred_linear, test_data$y)
print(cm_linear)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Dropout Enrolled Graduate
##   Dropout      220       65       29
##   Enrolled       0        0        0
##   Graduate      64       93      412
##
## Overall Statistics
##
##                Accuracy : 0.7157
##                  95% CI : (0.6847, 0.7453)
##     No Information Rate : 0.4994
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 0.4958
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: Dropout Class: Enrolled Class: Graduate
## Sensitivity                  0.7746          0.0000          0.9342
## Specificity                  0.8431          1.0000          0.6448
## Pos Pred Value               0.7006             NaN          0.7241
## Neg Pred Value               0.8875          0.8211          0.9076
## Prevalence                   0.3216          0.1789          0.4994
## Detection Rate               0.2492          0.0000          0.4666
## Detection Prevalence         0.3556          0.0000          0.6444
## Balanced Accuracy            0.8089          0.5000          0.7895
```

1.6.2 Radial Kernel Results

```r
cat("\n Radial Kernel Result:\n")
```

```
##
##  Radial Kernel Result:
```

```r
cm_radial <- confusionMatrix(pred_radial, test_data$y)
print(cm_radial)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Dropout Enrolled Graduate
##   Dropout      199       42       26
##   Enrolled      21       23        3
##   Graduate      64       93      412
##
## Overall Statistics
##
##                Accuracy : 0.718
##                  95% CI : (0.6871, 0.7475)
##     No Information Rate : 0.4994
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5065
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: Dropout Class: Enrolled Class: Graduate
## Sensitivity                  0.7007         0.14557          0.9342
## Specificity                  0.8865         0.96690          0.6448
## Pos Pred Value               0.7453         0.48936          0.7241
## Neg Pred Value               0.8620         0.83852          0.9076
## Prevalence                   0.3216         0.17894          0.4994
## Detection Rate               0.2254         0.02605          0.4666
```

```
## Detection Prevalence         0.3024         0.05323           0.6444
## Balanced Accuracy            0.7936         0.55623           0.7895
```

1.6.3 Polynomial Kernel Results

```r
cat("\n Polynomial Kernel Result:\n")
```

```
##
##  Polynomial Kernel Result:
```

```r
cm_poly <- confusionMatrix(pred_poly, test_data$y)
print(cm_poly)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Dropout Enrolled Graduate
##   Dropout      173       22       25
##   Enrolled      12        5        0
##   Graduate      99      131      416
##
## Overall Statistics
##
##                Accuracy : 0.6727
##                  95% CI : (0.6407, 0.7036)
##     No Information Rate : 0.4994
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.406
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: Dropout Class: Enrolled Class: Graduate
## Sensitivity                  0.6092        0.031646          0.9433
## Specificity                  0.9215        0.983448          0.4796
## Pos Pred Value               0.7864        0.294118          0.6440
## Neg Pred Value               0.8326        0.823326          0.8945
## Prevalence                   0.3216        0.178935          0.4994
## Detection Rate               0.1959        0.005663          0.4711
## Detection Prevalence         0.2492        0.019253          0.7316
## Balanced Accuracy            0.7653        0.507547          0.7115
```

1.7 Performance Comparison

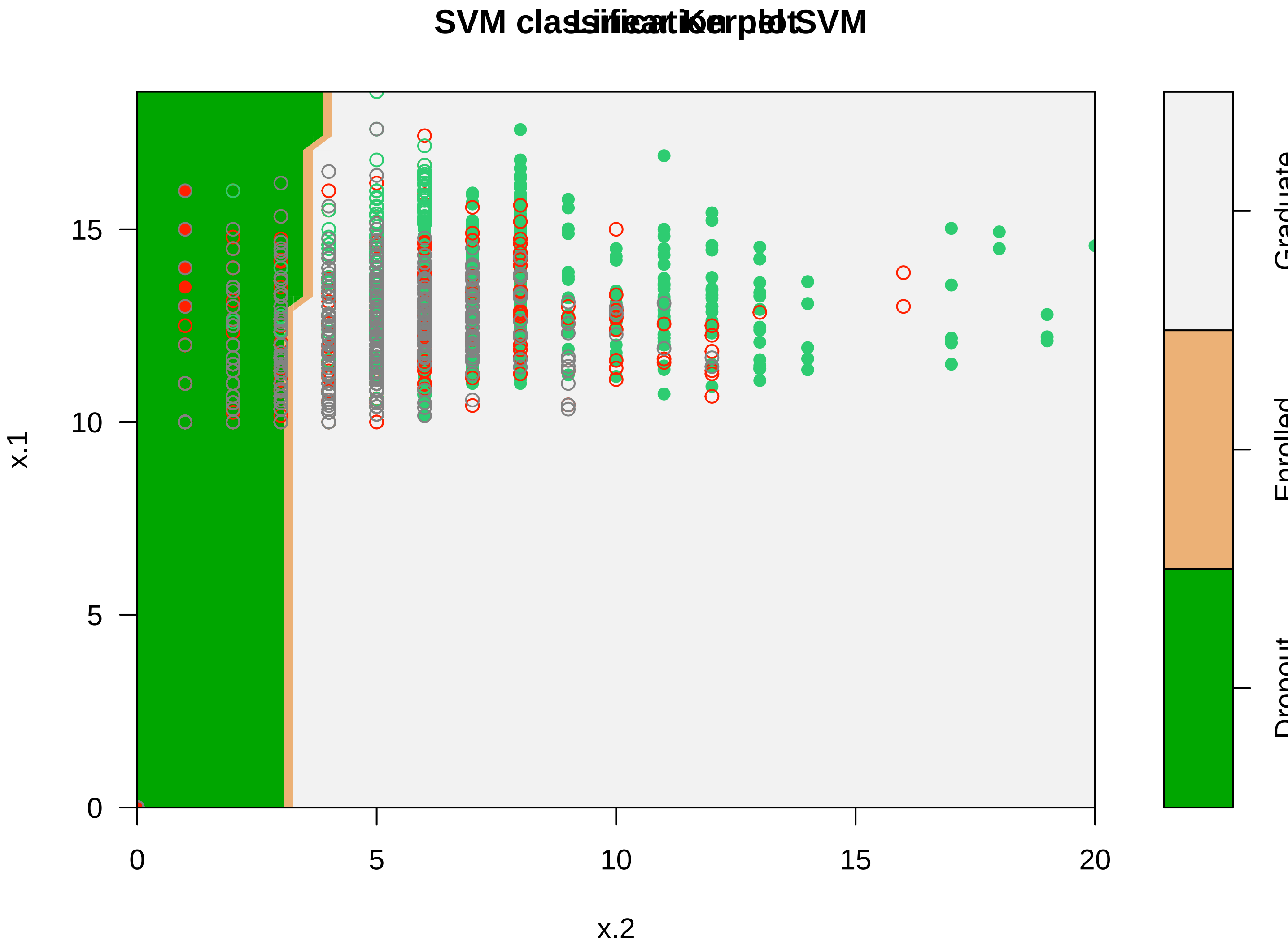
```r
results <- data.frame(
  Model = c("Linear", "Radial", "Polynomial"),
  Accuracy = c(cm_linear$overall['Accuracy'],
               cm_radial$overall['Accuracy'],
               cm_poly$overall['Accuracy'])
)

knitr::kable(results, digits = 3)
```

Model	Accuracy
Linear	0.716
Radial	0.718
Polynomial	0.673

1.8 Decision Boundary Visualizations

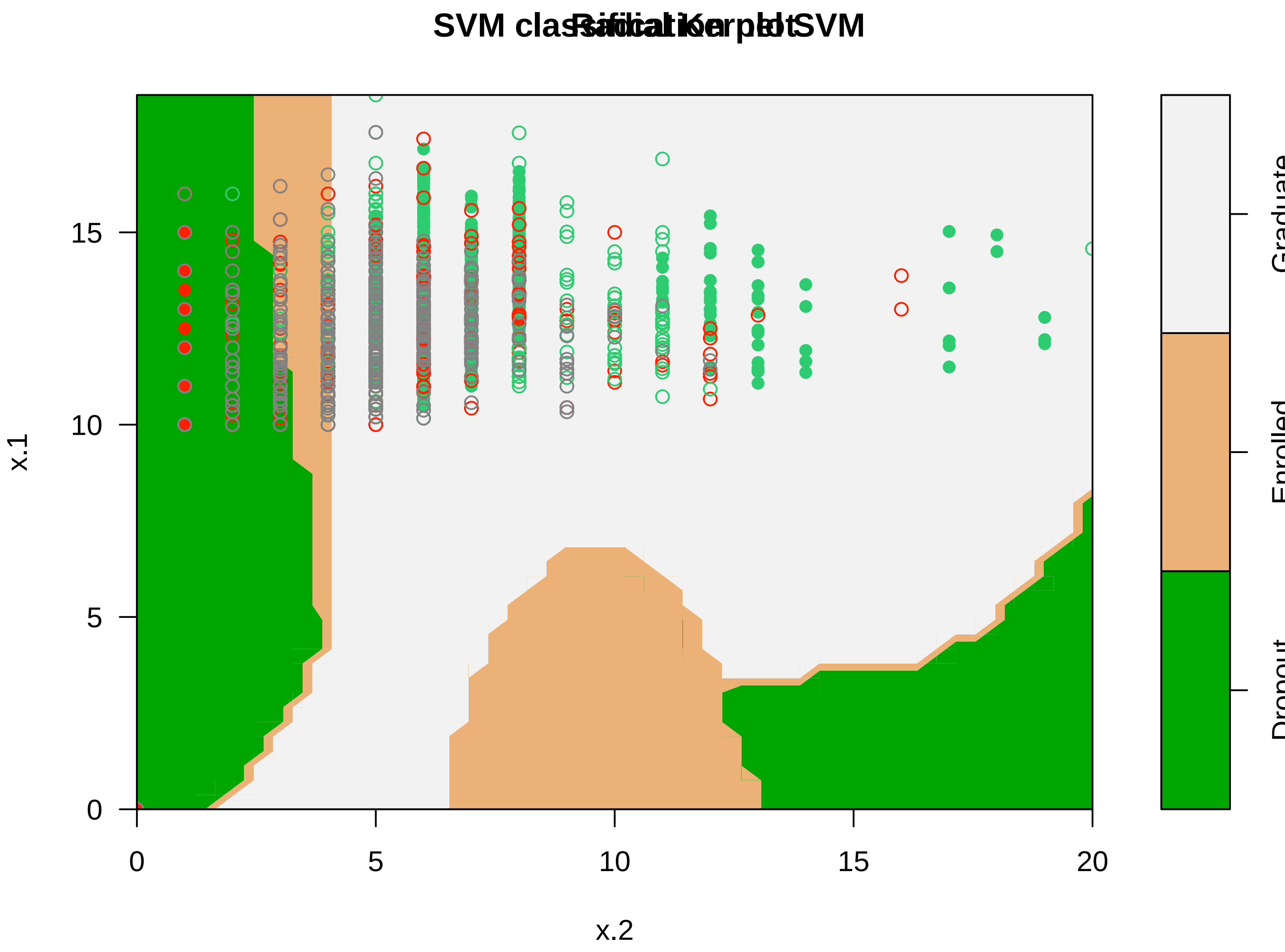
1.8.1 Linear Kernel

```
plot(svm_linear, train_data, x.1 ~ x.2,
     svSymbol = 1, dataSymbol = 16,
     symbolPalette = c("#ff2000", "#828282", "#2ecc71"),
     color.palette = terrain.colors)
title("Linear Kernel SVM")
```



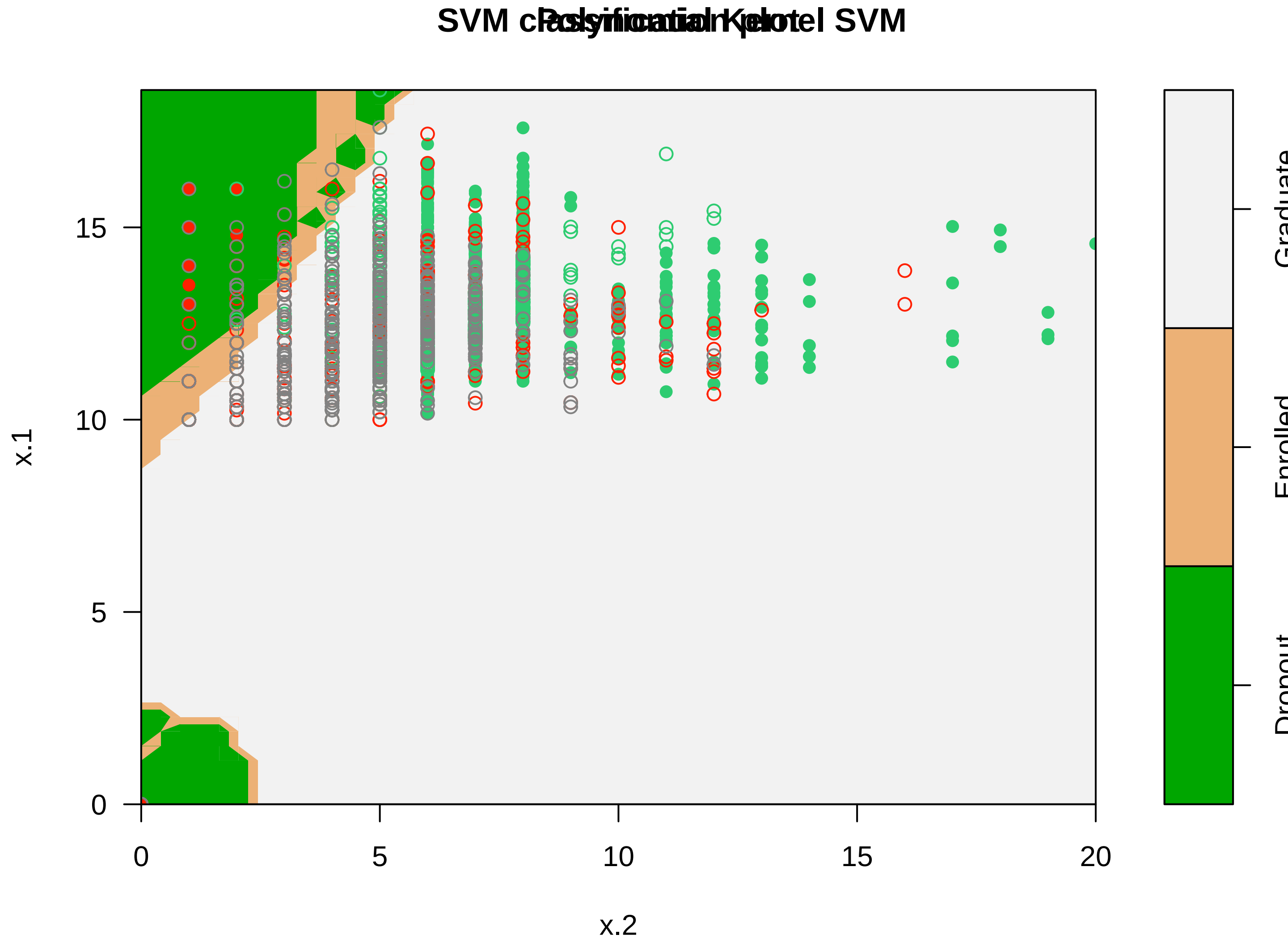
1.8.2 Radial Kernel

```
plot(svm_radial, train_data, x.1 ~ x.2,
     svSymbol = 1, dataSymbol = 16,
     symbolPalette = c("#ff2000", "#828282", "#2ecc71"),
     color.palette = terrain.colors)
title("Radial Kernel SVM")
```



1.8.3 Polynomial Kernel

```
plot(svm_poly, train_data, x.1 ~ x.2,
     svSymbol = 1, dataSymbol = 16,
     symbolPalette = c("#ff2000", "#828282", "#2ecc71"),
     color.palette = terrain.colors)
title("Polynomial Kernel SVM")
```


SVM classification plot
Polynomial Kernel SVM
x.1
x.2
0
5
10
15
20
Graduate
Enrolled
Dropout