Lab experiment – 8B

Name: Prithviraj Guntha

Reg. No.: 20BRS1188

Subject: Essentials of data analytics

Subject code: CSE3506

Professor: A Sheik Abdullah

Slot: L55+L56

1. **Perform SVM using different kernels:**
    a. **Sigmoid**
    b. **Polynomial**
    c. **Radial**

Code:

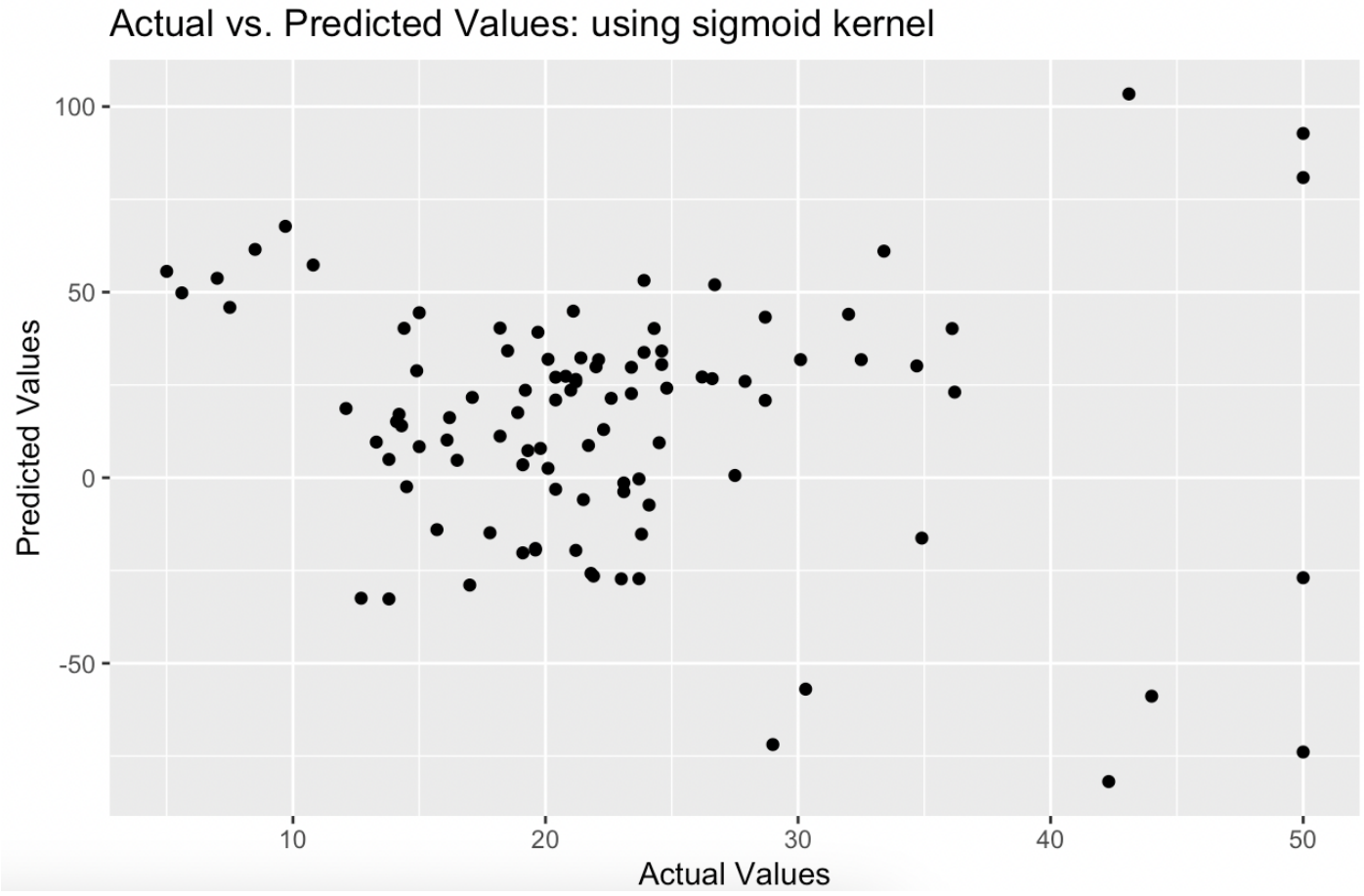
```
library(MASS)
library(caret)
library(e1071)
library(ggplot2)
library(lattice)

data <- read.csv("/Users/prithviraj/Downloads/Boston.csv")
set.seed(123)
trainIndex <- createDataPartition(data$MEDV, p = 0.8, list =
FALSE)
training_set <- data[trainIndex, ]
testing_set <- data[-trainIndex, ]


#for sigmoid kernel
svm_model <- svm(MEDV ~ ., data = training_set, kernel =
"sigmoid")
predictions <- predict(svm_model, testing_set)

# Calculate mean squared error (MSE) and mean absolute error
(MAE)
mse <- mean((testing_set$MEDV - predictions)^2)
mae <- mean(abs(testing_set$MEDV - predictions))
var_y <- var(testing_set$MEDV)
rsq_mse <- 1 - (mse / var_y)

# Create a scatter plot of predicted vs. actual values
ggplot(data = data.frame(actual = testing_set$MEDV, predicted
= predictions)) +
  geom_point(mapping = aes(x = actual, y = predicted)) +
  ggtitle("Actual vs. Predicted Values: using sigmoid kernel")
+
  xlab("Actual Values") +
  ylab("Predicted Values")

# Print the results
cat("MSE using sigmoid kernel: ", mse, "\n")
cat("MAE using sigmoid kernel: ", mae, "\n")
cat("R squared value using sigmoid kernel: ", rsq_mse)
```

```
#for polynomial kernel
svm_model <- svm(MEDV ~ ., data = training_set, kernel =
"polynomial", degree = 2, coef0 = 1)
predictions <- predict(svm_model, testing_set)

# Calculate mean squared error (MSE) and mean absolute error
(MAE)
mse <- mean((testing_set$MEDV - predictions)^2)
mae <- mean(abs(testing_set$MEDV - predictions))
var_y <- var(testing_set$MEDV)
rsq_mse <- 1 - (mse / var_y)

# Create a scatter plot of predicted vs. actual values
ggplot(data = data.frame(actual = testing_set$MEDV, predicted
= predictions)) +
  geom_point(mapping = aes(x = actual, y = predicted)) +
  ggtitle("Actual vs. Predicted Values: using polynomial
kernel") +
  xlab("Actual Values") +
  ylab("Predicted Values")

# Print the results
cat("MSE using polynomial kernel: ", mse, "\n")
cat("MAE using polynomial kernel: ", mae, "\n")
cat("R squared value using polynomial kernel: ", rsq_mse)


# using radial kernel
svm_model <- svm(MEDV ~ ., data = training_set, kernel =
"radial", gamma=0.1,cost=10)
predictions <- predict(svm_model, testing_set)

# Calculate mean squared error (MSE) and mean absolute error
(MAE)
mse <- mean((testing_set$MEDV - predictions)^2)
mae <- mean(abs(testing_set$MEDV - predictions))
var_y <- var(testing_set$MEDV)
rsq_mse <- 1 - (mse / var_y)

# Create a scatter plot of predicted vs. actual values
ggplot(data = data.frame(actual = testing_set$MEDV, predicted
= predictions)) +
  geom_point(mapping = aes(x = actual, y = predicted)) +
  ggtitle("Actual vs. Predicted Values: using radial kernel")
+
  xlab("Actual Values") +
  ylab("Predicted Values")

# Print the results
cat("MSE using radial kernel: ", mse, "\n")
```

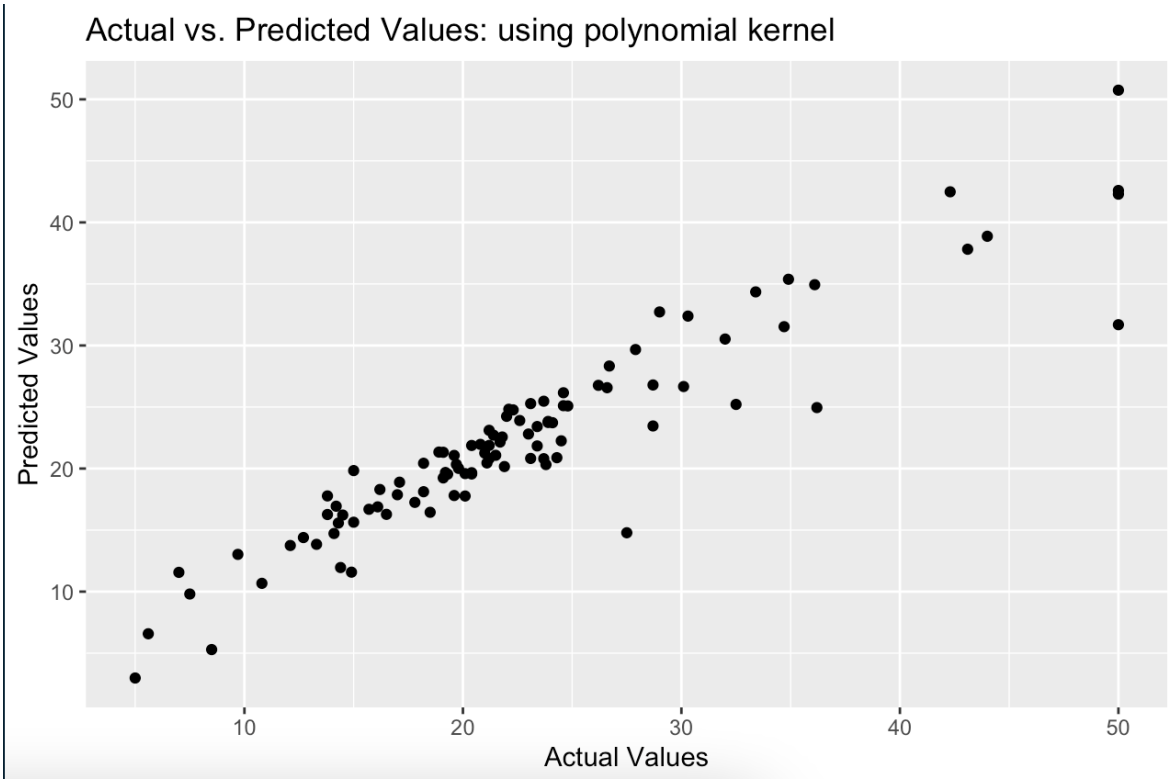
```
cat("MAE using radial kernel: ", mae, "\n")
cat("R squared value using radial kernel: ", rsq_mse)
```

Output:

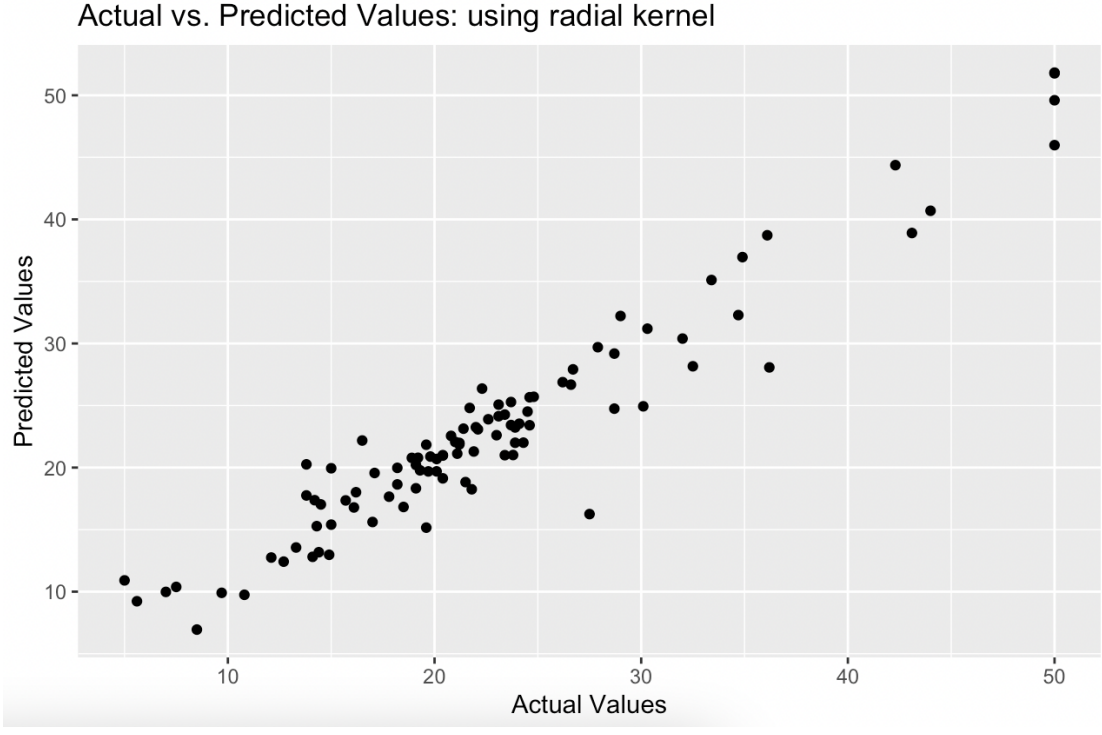
```
> cat("MSE using sigmoid kernel: ", mse, "\n")
MSE using sigmoid kernel:  1320.373
> cat("MAE using sigmoid kernel: ", mae, "\n")
MAE using sigmoid kernel:  25.29826
> cat("R squared value using sigmoid kernel: ", rsq_mse)
R squared value using sigmoid kernel:  -14.06419>
```




```
> cat("MSE using polynomial kernel: ", mse, "\n")
MSE using polynomial kernel:  12.06543
> cat("MAE using polynomial kernel: ", mae, "\n")
MAE using polynomial kernel:  2.189459
> cat("R squared value using polynomial kernel: ", rsq_mse)
R squared value using polynomial kernel:  0.8623451>
```




```
> cat("MSE using radial kernel: ", mse, "\n")
MSE using radial kernel:  7.151801
> cat("MAE using radial kernel: ", mae, "\n")
MAE using radial kernel:  1.964371
> cat("R squared value using radial kernel: ", rsq_mse)
R squared value using radial kernel:  0.9184048
```



Inference:

Hence, we have successfully implemented SVM using various kernels and we have found out that the Radial kernel performed much better than the polynomial and the sigmoid kernel. Sigmoid kernel performed below par compared to the polynomial and radial kernel.

Radial kernel > polynomial kernel > sigmoid kernel