Homework 8

Al Pakrosnis
Prof. Dale Embers
Sp25 STAT 385

1. Margin explanation

In a separable dataset with two features, the margin is the distance from the separating hyperplane to the closest data point of either class. A support vector machine seeks to maximize this margin to ensure better generalization. A larger margin typically implies a simpler model with lower variance, assuming data remains separable.

2. Coding

```r
# 2a. Load and subset data

library(ISLR2)
library(e1071)
library(ggplot2)
library(caret)
```

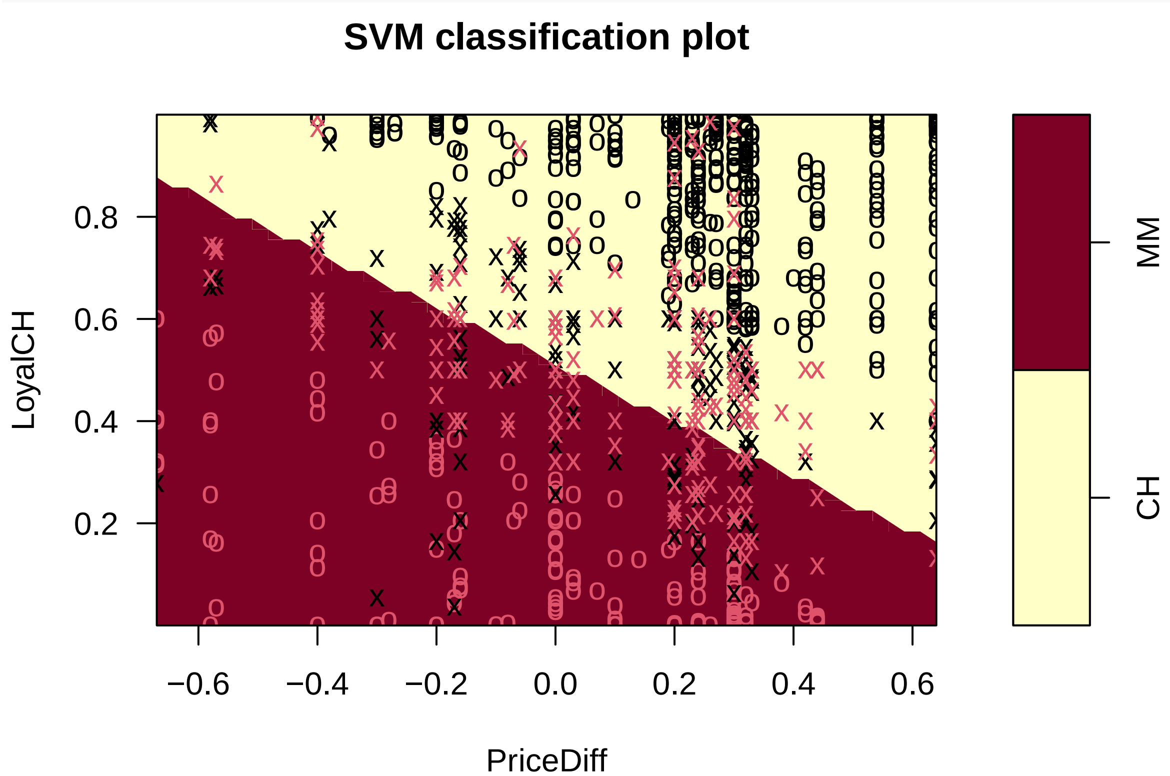
```
## Loading required package: lattice
```

```r
set.seed(20240410)
data(OJ)
oj_df <- OJ[, c("Purchase", "LoyalCH", "PriceDiff")]

train_index <- createDataPartition(oj_df$Purchase, p = 0.8, list = FALSE)
train_data <- oj_df[train_index, ]
test_data <- oj_df[-train_index, ]
```

```r
# 2b. Plot the data
plot(train_data$LoyalCH, train_data$PriceDiff, col = ifelse(train_data$Purchase == "CH", 1, 2),
     xlab = "LoyalCH", ylab = "PriceDiff", main = "OJ Purchase Data")
legend("topright", legend = c("CH", "MM"), col = c(1,2), pch = 1)
```


```r
# 2c. Linear SVM with default cost = 1
svm_linear <- svm(Purchase ~ ., data = train_data, kernel = "linear", cost = 1)
plot(svm_linear, train_data)
```




```r
# 2d. Evaluate on test set
pred_linear <- predict(svm_linear, test_data)
```

```r
conf_matrix_linear <- table(Predicted = pred_linear, Actual = test_data$Purchase)
error_linear <- mean(pred_linear != test_data$Purchase)
conf_matrix_linear
```

```
##          Actual
## Predicted  CH  MM
##        CH 114  16
##        MM  16  67
```

```r
error_linear
```

```
## [1] 0.1502347
```

```r
# 2e. Tune linear SVM (cost 1 to 10 by 0.2)
tune_linear <- tune.svm(Purchase ~ ., data = train_data, kernel = "linear", cost = seq(1, 10, by = 0.2))
tune_linear$best.parameters
```

```
##   cost
## 4  1.6
```

```r
best_linear <- tune_linear$best.model
pred_best_linear <- predict(best_linear, test_data)
conf_matrix_best_linear <- table(Predicted = pred_best_linear, Actual = test_data$Purchase)
error_best_linear <- mean(pred_best_linear != test_data$Purchase)
conf_matrix_best_linear
```

```
##          Actual
## Predicted  CH  MM
##        CH 114  16
##        MM  16  67
```

```r
error_best_linear
```

```
## [1] 0.1502347
```

```r
# 2f. Radial SVM tuning
cost_seq <- seq(10, 100, by = 10)
gamma_seq <- 3 / (1:10)
tune_radial <- tune.svm(Purchase ~ ., data = train_data, kernel = "radial",
                        cost = cost_seq, gamma = gamma_seq)
tune_radial$best.parameters
```

```
##       gamma cost
## 7 0.4285714   10
```

```r
best_radial <- tune_radial$best.model
pred_radial <- predict(best_radial, test_data)
conf_matrix_radial <- table(Predicted = pred_radial, Actual = test_data$Purchase)
error_radial <- mean(pred_radial != test_data$Purchase)
conf_matrix_radial
```

```
##          Actual
## Predicted  CH  MM
##        CH 114  16
##        MM  16  67
```

```r
error_radial
```

```
## [1] 0.1502347
```

```
# 2g. Polynomial SVM tuning
tune_poly <- tune.svm(Purchase ~ ., data = train_data, kernel = "polynomial",
                      cost = 1:10, degree = 2:6)
tune_poly$best.parameters
```

```
##    degree cost
## 12      3    3
```

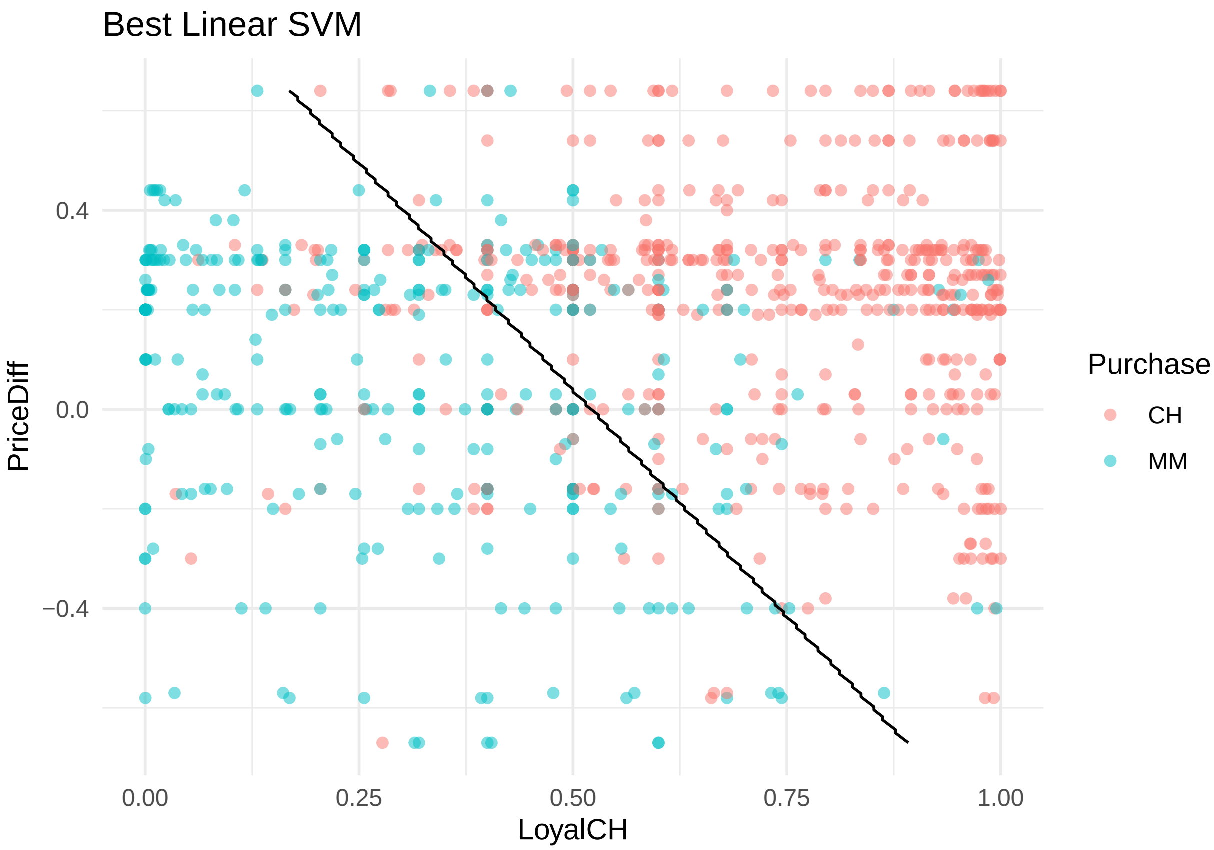
```
best_poly <- tune_poly$best.model
pred_poly <- predict(best_poly, test_data)
conf_matrix_poly <- table(Predicted = pred_poly, Actual = test_data$Purchase)
error_poly <- mean(pred_poly != test_data$Purchase)
conf_matrix_poly
```

```
##          Actual
## Predicted  CH  MM
##        CH 124  37
##        MM   6  46
```

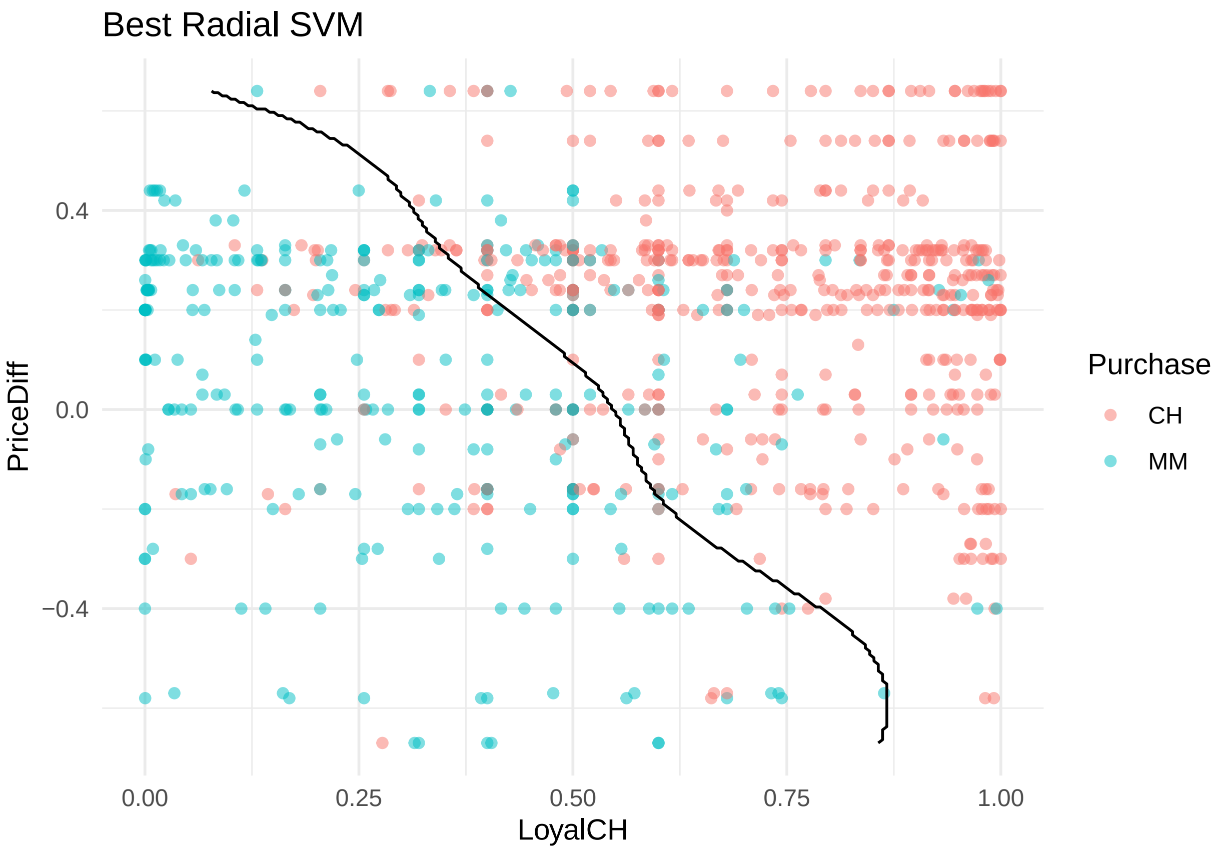
```
error_poly
```

```
## [1] 0.2018779
```

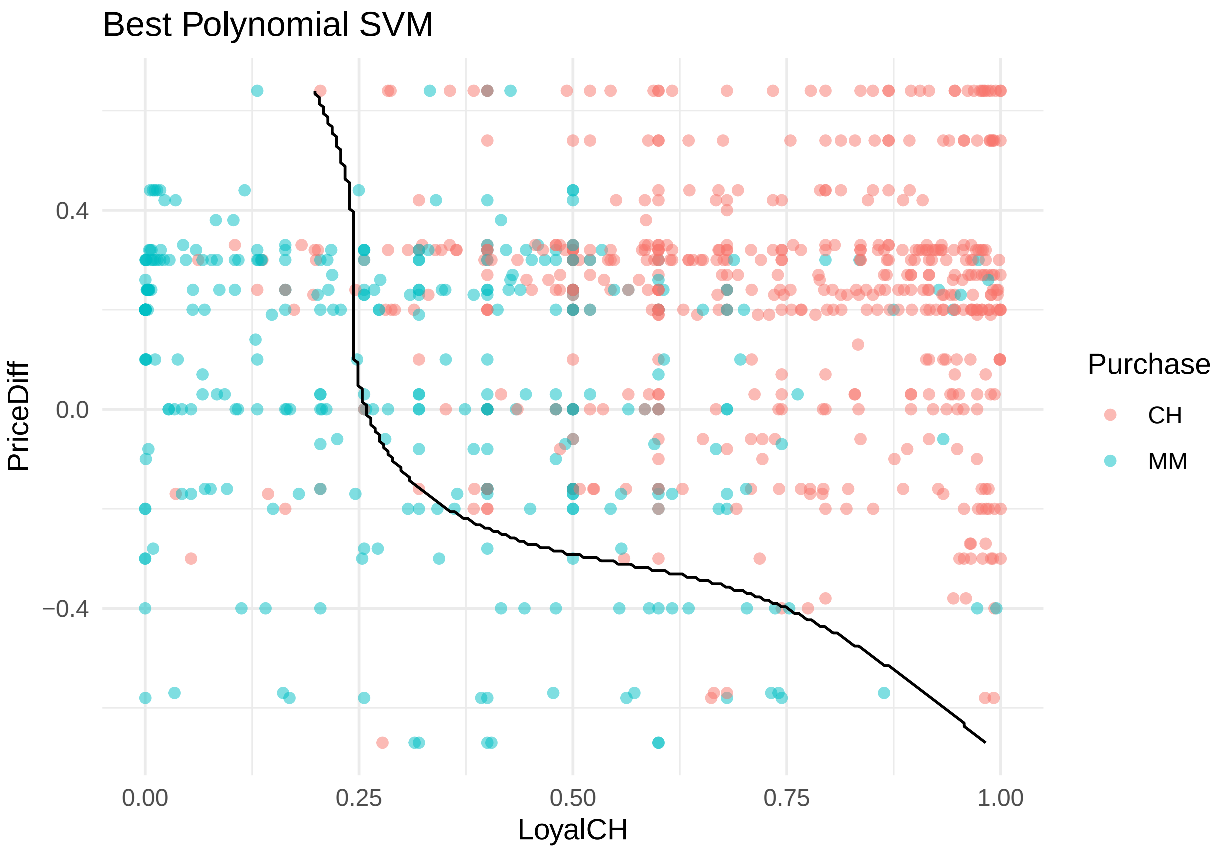
```
#  2h. Plot separation boundaries for all models
par(mfrow = c(3,3))
# Linear
grid_linear <- expand.grid(
  LoyalCH = seq(min(train_data$LoyalCH), max(train_data$LoyalCH), length.out = 200),
  PriceDiff = seq(min(train_data$PriceDiff), max(train_data$PriceDiff), length.out = 200)
)
grid_linear$Prediction <- predict(best_linear, grid_linear)
ggplot(train_data, aes(x = LoyalCH, y = PriceDiff, color = Purchase)) +
  geom_point(alpha = 0.5) +
  geom_contour(data = cbind(grid_linear, z = as.numeric(grid_linear$Prediction == "CH")),
               aes(z = z), breaks = 0.5, color = "black") +
  ggtitle("Best Linear SVM") + theme_minimal()
```




```r
# Radial
grid_radial <- expand.grid(
  LoyalCH = seq(min(train_data$LoyalCH), max(train_data$LoyalCH), length.out = 200),
  PriceDiff = seq(min(train_data$PriceDiff), max(train_data$PriceDiff), length.out = 200)
)
grid_radial$Prediction <- predict(best_radial, grid_radial)
ggplot(train_data, aes(x = LoyalCH, y = PriceDiff, color = Purchase)) +
  geom_point(alpha = 0.5) +
  geom_contour(data = cbind(grid_radial, z = as.numeric(grid_radial$Prediction == "CH")),
               aes(z = z), breaks = 0.5, color = "black") +
  ggtitle("Best Radial SVM") + theme_minimal()
```




```r
# Polynomial
grid_poly <- expand.grid(
  LoyalCH = seq(min(train_data$LoyalCH), max(train_data$LoyalCH), length.out = 200),
  PriceDiff = seq(min(train_data$PriceDiff), max(train_data$PriceDiff), length.out = 200)
)
grid_poly$Prediction <- predict(best_poly, grid_poly)
ggplot(train_data, aes(x = LoyalCH, y = PriceDiff, color = Purchase)) +
  geom_point(alpha = 0.5) +
  geom_contour(data = cbind(grid_poly, z = as.numeric(grid_poly$Prediction == "CH")),
               aes(z = z), breaks = 0.5, color = "black") +
  ggtitle("Best Polynomial SVM") + theme_minimal()
```


Best Polynomial SVM
PriceDiff
0.4
0.0
−0.4
0.00
0.25
0.50
0.75
1.00
LoyalCH
Purchase
CH
MM