Министерство образования и науки РФ
Санкт-Петербургский Политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа искусственного интеллекта

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 3

«Метод опорных векторов»
по дисциплине «Машинное обучение»

Выполнила:
студентка гр. 3540201/20301 _____ Климова О. А.
подпись, дата

Проверил:
д.т.н., проф. _____ Уткин Л. В.
подпись, дата

Санкт-Петербург
2022

Содержание

Постановка задачи

В рамках данной работы необходимо:

1. Построить алгоритм метода опорных векторов типа "C-classification" с параметром C = 1, используя ядро "linear". Визуализировать разбиение пространства признаков на области с помощью полученной модели. Вывести количество полученных опорных векторов, а также ошибки классификации на обучающей и тестовой выборках.

2. Используя алгоритм метода опорных векторов типа "C-classification" с линейным ядром, добиться нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра C. Выберать оптимальное значение данного параметра и объяснить свой выбор. Всегда ли нужно добиваться минимизации ошибки на обучающей выборке?

3. Среди ядер "polynomial", "radial" и "sigmoid" выберать оптимальное в плане количества ошибок на тестовой выборке. Попробовать различные значения параметра degree для полиномиального ядра.

4. Среди ядер "polynomial", "radial" и "sigmoid" выбрать оптимальное в плане количества ошибок на тестовой выборке.

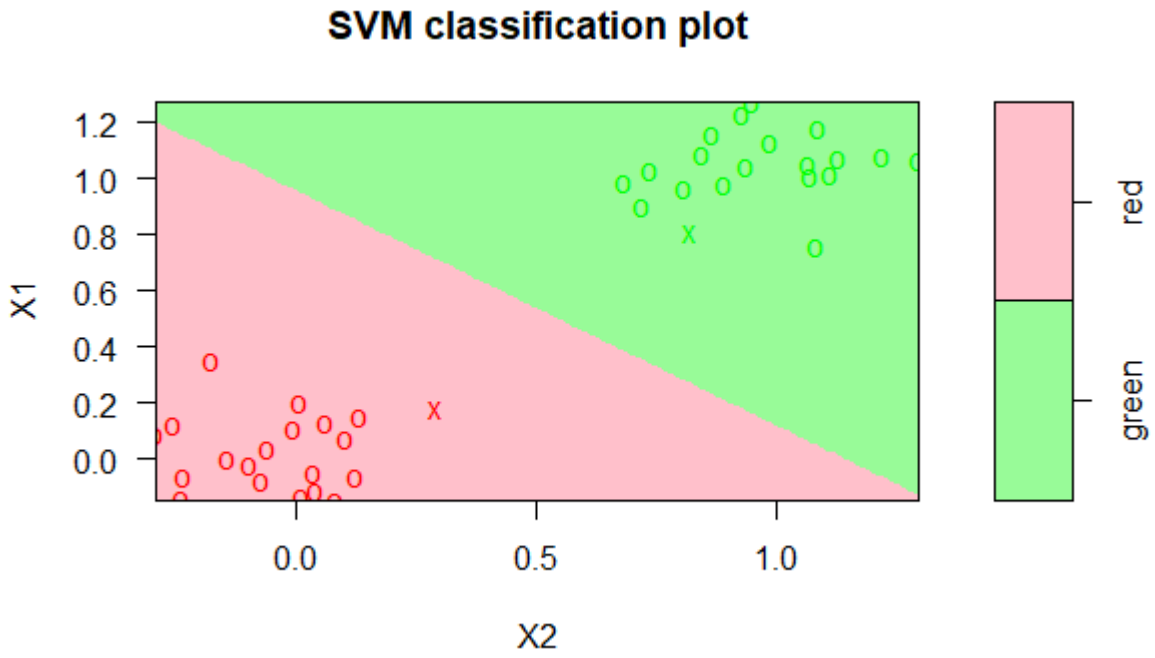
5. Среди ядер "polynomial", "radial" и "sigmoid" выбрать оптимальное в плане количества ошибок на тестовой выборке. Изменяя значение параметра gamma, продемонстрировать эффект переобучения, выполнить при этом визуализацию разбиения пространства признаков на области.

6. Построить алгоритм метода опорных векторов типа "eps-regression" с параметром C = 1, используя ядро "radial". Отобразить на графике зависимость среднеквадратичной ошибки на обучающей выборке от значения параметра ε. Прокомментировать полученный результат.

1 Задание 1

В данном задании был построен алгоритм метода опорных векторов типа "C-classification" с параметром C = 1, используя ядро "linear". Код представлен в Приложении 1.

Далее можно видеть визуализированное разбиение пространства признаков на области с помощью полученной модели:



Количество полученных опорных векторов: 2

```
Number of Support Vectors:  2
```

Ошибка классификации на обучающей выборке: 0%

```
      predictions1Train
       green red
green    20   0
red       0  20
```

Ошибка классификации на тестовой выборке: 0%

```
      predictions1Test
       green red
green    20   0
red       0  20
```

2 Задание 2

В данном задании, используя алгоритм метода опорных векторов типа "C-classification" с линейным ядром, была получена нулевая ошибка сначала на обучающей выборке, а затем на тестовой, путем изменения параметра C. Код представлен в Приложении 2.

Ошибка на обучающей выборке равна 0% при $C \geq 183$.

Ошибка на тестовой выборке равна 0% при $C \leq 71$

В качестве оптимального значения C можно взять 71, так как оно является граничным и при нем значение ошибки на тестовой выборке составляет 0%, а на обучающей 2% (что не очень много).

На обучающей выборке добиваться минимизации ошибки всегда не нужно, так как может возникнуть переобучение – модель будет идеально предсказывать значения из обучающей выборки, но выдавать ошибки на тестовой выборке.

3 Задание 3

В данном задании среди ядер "polynomial", "radial" и "sigmoid" было выбрано оптимальное в плане количества ошибок на тестовой выборке, для этого были построены модели с одинаковыми параметрами C = 1 и type="C-classification", но разными значениями ядер.

Код представлен в Приложении 3.

Ядро "polynomial": ошибка = 45%

```
       predictions3Test
        green red
  green    11   0
  red       9   0
> accuracy2
[1] 0.55
```

Ядро "radial": ошибка = 5%

```
       predictions3Test
        green red
  green    10   1
  red       0   9
> accuracy2
[1] 0.95
```

Ядро "sigmoid": ошибка = 60%

```
       predictions3Test
        green red
  green     8   3
  red       9   0
> accuracy2
[1] 0.4
```

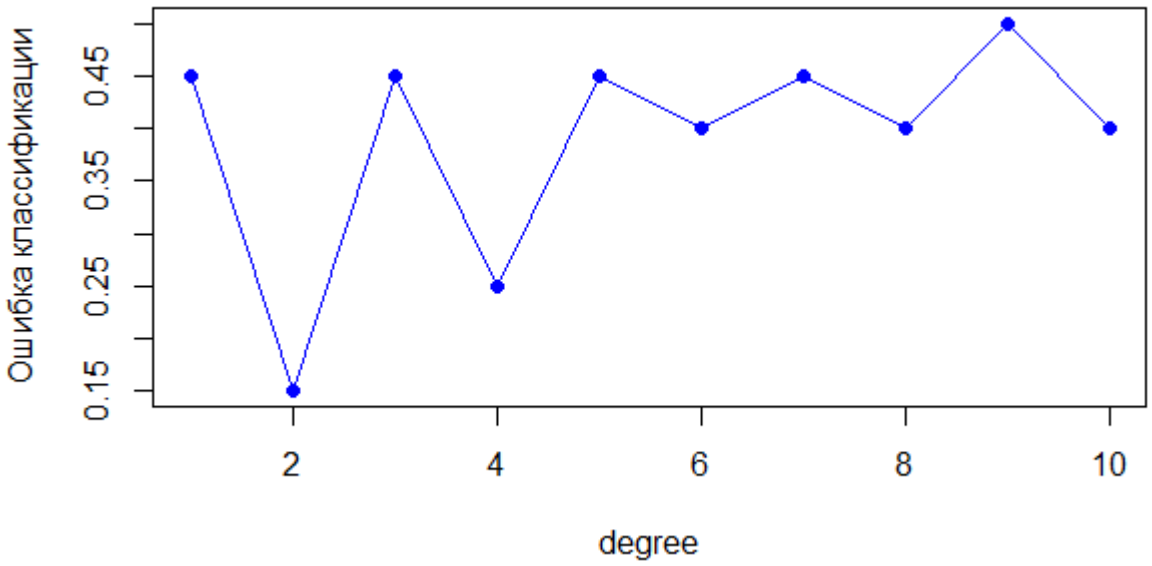
Таким образом, можно сделать вывод, что оптимальнее всего относительно ошибок на тестовой выборке использовать ядро "radial".

Далее представлены значения ошибки на тестовой выборке при постоянных параметрах C = 1, type="C-classification", kernel = "polynomial", но различных значениях параметра degree:

Значение degree	Ошибка классификации
1	0.45
2	0.15
3	0.45

4	0.25
5	0.45
6	0.40
7	0.45
8	0.40
9	0.50
10	0.40

График зависимости имеет следующий вид:



Можно заметить, что четные значения степени дают меньшую ошибка. Ошибка минимальна при degree = 2 (0.15 или 15%).

4 Задание 4

В данном задании для датасета «svmdata4» среди ядер "polynomial", "radial" и "sigmoid" было выбрано оптимальное в плане количества ошибок на тестовой выборке.

Код представлен в Приложении 4.

Ядро "polynomial": ошибка = 13%

```
       predictions4Test
        green red
  green    79  20
  red       6  95
> 1 - accuracy2
[1] 0.13
```

Ядро "radial": ошибка = 11%

```
       predictions4Test
        green red
  green    88  11
  red      11  90
> 1 - accuracy2
[1] 0.11
```

Ядро "sigmoid": ошибка = 19.5%

```
       predictions4Test
        green red
  green    80  19
  red      20  81
> 1 - accuracy2
[1] 0.195
```

Для датасета «svmdata4» оптимальным является ядро "radial", при нем ошибка составляет 11%.

5 Задание 5

В данном задании среди ядер "polynomial", "radial" и "sigmoid" выбирается оптимальное относительно ошибки на тесте, а также, изменяя значение параметра gamma, демонстрируется эффект переобучения. Код представлен в Приложении 5.

Ядро "polynomial" при degree = 2: ошибка = 6%

```
        predictions5Test
         green red
  green    60   0
  red       7  53
> 1 - accuracy2
[1] 0.05833333
```

Ядро "radial": ошибка = 8%

```
        predictions5Test
         green red
  green    57   3
  red       7  53
> 1 - accuracy2
[1] 0.08333333
```

Ядро "sigmoid": ошибка = 53%

```
        predictions5Test
         green red
  green    22  38
  red      26  34
> 1 - accuracy2
[1] 0.5333333
```

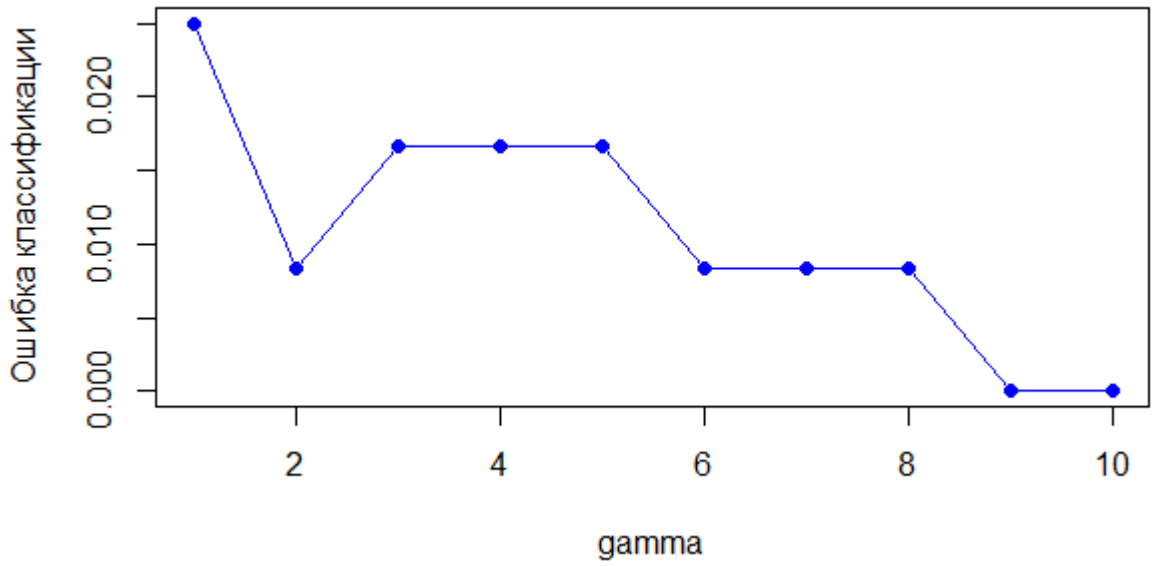
Для датасета «svmdata5» оптимальным является ядро "polynomial", при нем ошибка составляет 6%.

Изменение параметра gamma:

Значение gamma	Ошибка классификации
1	0.025000000
2	0.008333333
3	0.016666667
4	0.016666667
5	0.016666667

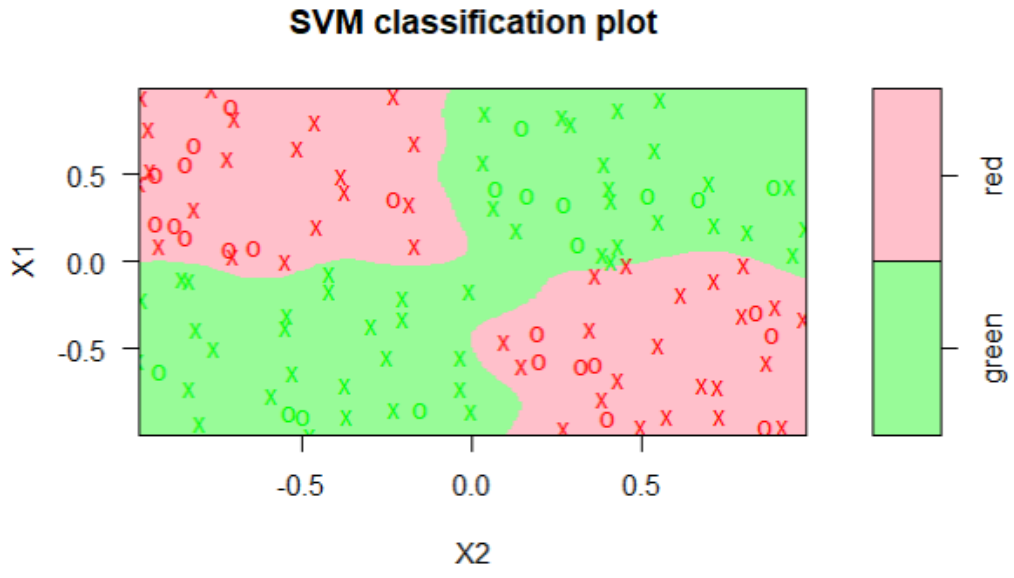
6	0.008333333
7	0.008333333
8	0.008333333
9	0.000000000
10	0.000000000

График данной зависимости имеет вид:



При gamma = 9 происходит переобучение (ошибка на обучающем множестве равна 0).

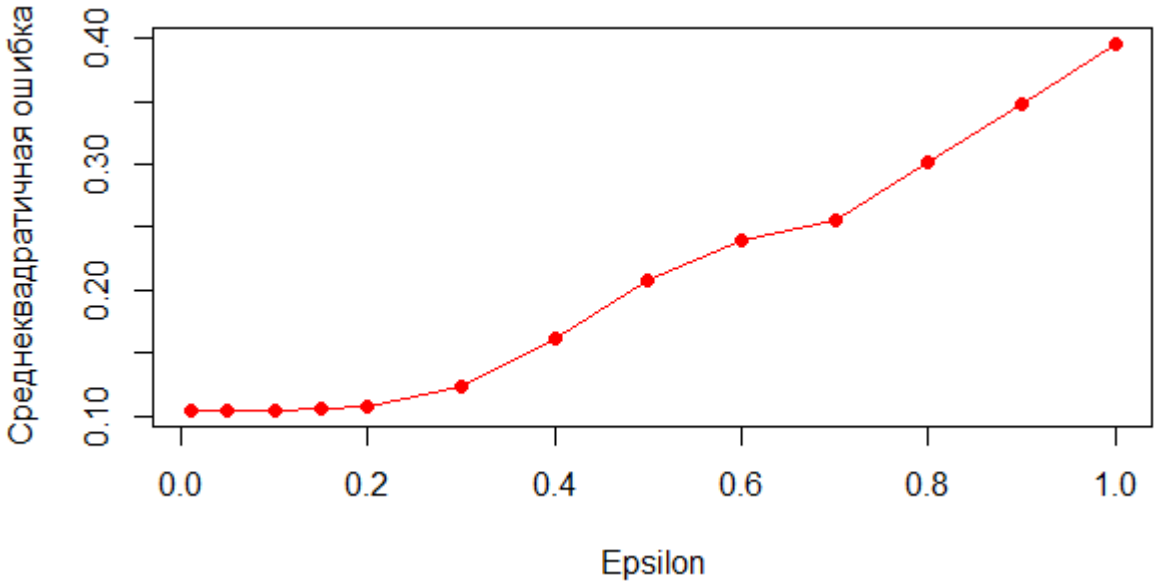
Визуализацию разбиения пространства признаков на области при ядре «radial» и gamma = 9:



6 Задание 6

В данном задании строится алгоритм метода опорных векторов типа "eps-regression" с параметром C = 1, используя ядро "radial". Код представлен в Приложении 6.

Зависимость среднеквадратичной ошибки на обучающей выборке от значения параметра ε имеет вид:



Среднеквадратичная ошибка минимальна при ε = 0.1. При увеличении значения ε среднеквадратичная ошибка увеличивается.

Код, используемый для работы с датасетом svmdata4 представлен в Приложении 3.

Приложение 1. Код для задания 1

```
##ЗАДАНИЕ 1
library(e1071)
#окрашивание фона
area.pallete = function(n = 2)
{
  cols = rainbow(n)
  cols[1:2] = c("PaleGreen", "Pink")
  return(cols)
}
#цвет символов
symbols.pallete = c("Green", "Red")
#загрузка обучающей и тестовой выборки
data1Train <- read.table("svmdata1.txt", sep = "\t", stringsAsFactors = TRUE)
data1Test <- read.table("svmdata1test.txt", sep = "\t", stringsAsFactors = TRUE)
#построение модели типа "C-classification" с параметром C = 1 и ядром "linear"
svmModel1 <- svm(Color ~., data=data1Train, type="C-classification", cost=1,
          kernel="linear")
#построение графика
plot(svmModel1, data1Train, grid=250,  symbolPalette = symbols.pallete, color.palette = area.pallete)
#определение полученных опорных векторов
svmModel1
#определение ошибки классификации на обучающей выборке
predictions1Train <- predict(svmModel1, data1Train)
table(data1Train$Color, predictions1Train)
#определение ошибки классификации на тестовой выборке
predictions1Test <- predict(svmModel1, data1Test)
table(data1Test$Color, predictions1Test)
```

Приложение 2. Код для задания 2

```
##ЗАДАНИЕ 2
library(e1071)
#загрузка обучающей и тестовой выборки
data2Train <- read.table("svmdata2.txt", sep = "\t", stringsAsFactors = TRUE)
data2Test <- read.table("svmdata2test.txt", sep = "\t", stringsAsFactors = TRUE)
#построение первой модели типа "C-classification" с параметром C = 183 и ядром "linear"
svmModel2 <- svm(Colors ~., data=data2Train, type="C-classification", cost=183,
                 kernel="linear")

#определение точности классификации на обучающей выборке
predictions2Train <- predict(svmModel2, data2Train)
tab1 = table(data2Train$Colors, predictions2Train)
tab1
accuracy1 = (tab1[1,1] + tab1[2,2]) / (tab1[1,1] + tab1[2,2] + tab1[1,2] + tab1[2,1])
accuracy1

#построение второй модели типа "C-classification" с параметром C = 71  и ядром "linear"
svmModel2 <- svm(Colors ~., data=data2Train, type="C-classification", cost=71,
                 kernel="linear")
#определение точности классификации на тестовой выборке
predictions2Test <- predict(svmModel2, data2Test)
tab2 = table(data2Test$Colors, predictions2Test)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
accuracy2
```

Приложение 3. Код для задания 3

```
##ЗАДАНИЕ 3
library(e1071)
#считываем данные и разделяем на тестовую и обучающую выборки
data3_raw <- read.table("svmdata3.txt", sep = "\t", stringsAsFactors = TRUE)
n <- dim(data3_raw)[1]
data3_rand <- data3_raw[ order(runif(n)),]
#80% для обучения
nt <- as.integer(n*0.8)
data3Train <- data3_rand[1:nt, ]
data3Test <- data3_rand[(nt+1):n, ]

#построение модели типа "C-classification" с параметром C = 1 и ядром "polynomial"
svmModel3 <- svm(Colors ~., data=data3Train, type="C-classification", cost=1,
        kernel="polynomial")
#определение точности классификации на тестовой выборке
predictions3Test <- predict(svmModel3, data3Test)
tab2 = table(data3Test$Colors, predictions3Test)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
tab2
accuracy2

#построение модели типа "C-classification" с параметром C = 1 и ядром "radial"
svmModel3 <- svm(Colors ~., data=data3Train, type="C-classification", cost=1,
        kernel="radial")
#определение точности классификации на тестовой выборке
predictions3Test <- predict(svmModel3, data3Test)
tab2 = table(data3Test$Colors, predictions3Test)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
tab2
accuracy2

#построение модели типа "C-classification" с параметром C = 1 и ядром "sigmoid"
svmModel3 <- svm(Colors ~., data=data3Train, type="C-classification", cost=1,
        kernel="sigmoid")
#определение точности классификации на тестовой выборке
predictions3Test <- predict(svmModel3, data3Test)
tab2 = table(data3Test$Colors, predictions3Test)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
tab2
accuracy2

#определение ошибки при различных значениях degree
Depdeg = matrix(c(1:10, 1:10),nrow = 10, ncol = 2, byrow = TRUE)
for (i in 1:10){
```

```
#построение модели типа "C-classification" с параметром C = 1 и ядром "polynomial"
svmModel3 <- svm(Colors ~., data=data3Train, type="C-classification", cost=1,
                 kernel="polynomial", degree = i)
#определение точности классификации на тестовой выборке
predictions3Test <- predict(svmModel3, data3Test)
tab2 = table(data3Test$Colors, predictions3Test)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
Depdeg[i, 1] = i
Depdeg[i, 2] = 1 - accuracy2
}
plot(Depdeg[,1],Depdeg[,2], col="blue", ylab="Ошибка классификации", xlab="degree", pch =
19, type="o")
Depdeg
```

Приложение 4. Код для задания 4

```
##Задание 4
data4Train <- read.table("svmdata4.txt", sep = "\t", stringsAsFactors = TRUE)
data4Test <- read.table("svmdata4test.txt", sep = "\t", stringsAsFactors = TRUE)

#построение модели типа "C-classification" с параметром C = 1 и ядром "polynomial"
svmModel4 <- svm(Colors ~., data=data4Train, type="C-classification", cost=1,
        kernel="polynomial")
#определение точности классификации на тестовой выборке
predictions4Test <- predict(svmModel4, data4Test)
tab2 = table(data4Test$Colors, predictions4Test)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
tab2
1 - accuracy2

#построение модели типа "C-classification" с параметром C = 1 и ядром "radial"
svmModel4 <- svm(Colors ~., data=data4Train, type="C-classification", cost=1,
        kernel="radial")
#определение точности классификации на тестовой выборке
predictions4Test <- predict(svmModel4, data4Test)
tab2 = table(data4Test$Colors, predictions4Test)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
tab2
1 - accuracy2

#построение модели типа "C-classification" с параметром C = 1 и ядром "sigmoid"
svmModel4 <- svm(Colors ~., data=data4Train, type="C-classification", cost=1,
        kernel="sigmoid")
#определение точности классификации на тестовой выборке
predictions4Test <- predict(svmModel4, data4Test)
tab2 = table(data4Test$Colors, predictions4Test)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
tab2
1 - accuracy2
```

Приложение 5. Код для задания 5

```
##ЗАДАНИЕ 5
data5Train <- read.table("svmdata5.txt", sep = "\t", stringsAsFactors = TRUE)
data5Test <- read.table("svmdata5test.txt", sep = "\t", stringsAsFactors = TRUE)

#построение модели типа "C-classification" с параметром C = 1 и ядром "polynomial"
svmModel5 <- svm(Colors ~., data=data5Train, type="C-classification", cost=1,
                 kernel="polynomial", degree = 2)
#определение точности классификации на тестовой выборке
predictions5Test <- predict(svmModel5, data5Test)
tab2 = table(data5Test$Colors, predictions5Test)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
tab2
1 - accuracy2

#построение модели типа "C-classification" с параметром C = 1 и ядром "radial"
svmModel5 <- svm(Colors ~., data=data5Train, type="C-classification", cost=1,
                 kernel="radial")
#определение точности классификации на тестовой выборке
predictions5Test <- predict(svmModel5, data5Test)
tab2 = table(data5Test$Colors, predictions5Test)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
tab2
1 - accuracy2

#построение модели типа "C-classification" с параметром C = 1 и ядром "sigmoid"
svmModel5 <- svm(Colors ~., data=data5Train, type="C-classification", cost=1,
                 kernel="sigmoid")
#определение точности классификации на тестовой выборке
predictions5Test <- predict(svmModel5, data5Test)
tab2 = table(data5Test$Colors, predictions5Test)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
tab2
1 - accuracy2

##демонстрация переобучения при изменении gamma
#определение ошибки при различных значениях degree
Depgam = matrix(c(1:10, 1:10),nrow = 10, ncol = 2, byrow = TRUE)
for (i in 1:10){
  #построение модели типа "C-classification" с параметром C = 1 и ядром "radial"
  svmModel5 <- svm(Colors ~., data=data5Train, type="C-classification", cost=1,
                   kernel="radial", gamma = i)
  #определение точности классификации на обучающей выборке
  predictions5Train <- predict(svmModel5, data5Train)
  tab2 = table(data5Train$Colors, predictions5Train)
```

```
  accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
  Depgam[i, 1] = i
  Depgam[i, 2] = 1 - accuracy2
}
plot(Depgam[,1],Depgam[,2], col="blue", ylab="Ошибка классификации", xlab="gamma", pch
= 19, type="o")
Depgam[,2]

area.pallete = function(n = 2)
{
  cols = rainbow(n)
  cols[1:2] = c("PaleGreen", "Pink")
  return(cols)
}
#цвет символов
symbols.pallete = c("Green", "Red")
#построение модели типа "C-classification" с параметром C = 1 и ядром "linear"
svmModel5 <- svm(Colors ~., data=data5Train, type="C-classification", cost=1,
                 kernel="radial", gamma = 9)
#построение графика
plot(svmModel5, data5Train, grid=250,  symbolPalette = symbols.pallete, color.palette =
area.pallete)
```

Приложение 6. Код для задания 6

```
##ЗАДАНИЕ 6
data6 <- read.table("svmdata6.txt", sep = "\t", stringsAsFactors = TRUE)
#построение регрессионной модели
regression_model <- svm(data6$X, data6$Y, type = "eps-regression", cost=1,
                        kernel="radial")
Eps = c(0.01,0.05,0.1,0.15,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)
DepEps = c()
for(i in 1:length(Eps)){
  regression_model <- svm(data6$X, data6$Y, type = "eps-regression", cost=1,
                          epsilon=Eps[i], kernel="radial")
  predictions = predict(regression_model, data6$X)
#подсчет среднеквадратичной ошибки
  DepEps <- append(DepEps, sd(data6$Y-predictions))
}
plot(Eps, DepEps,col="red", pch = 19, xlab="Epsilon", ylab="Среднеквадратичная ошибка",
type = "o")
```