

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

ПРАКТИЧНЕ МАШИННЕ НАВЧАННЯ.

МЕТОД ОПОРНИХ ВЕКТОРІВ

Звіт

до лабораторної роботи №5

з курсу «Методи і засоби комп'ютерного навчання»

Виконав:

студент гр. СПКм-12

Сергієнко В.Р.

Прийняв

Пукач А.І.

Львів 2014

Мета: оволодіти методом опорних векторів в задачах класифікації та відновлення регресії в системі R.

Короткі теоретичні відомості

Функція SVM пакета *e1071*

Метод опорних векторів реалізований в пакеті *e1071*.

Основна функція, що реалізує метод – *svm*.

```
svm(formula, data = NULL, ..., subset, na.action = na.omit, scale = TRUE)
svm(x, y = NULL, scale = TRUE, type = NULL, kernel =
"radial", degree = 3, gamma = if (is.vector (x)) 1 else 1 / ncol (x),
coef0 = 0, cost = 1, nu = 0,5, class.weights = NULL, cachesize = 40,
tolerance= 0,001, epsilon= 0,1, shrinking = TRUE, cross = 0,
probability = FALSE, fitted = TRUE, ..., subset, na.action = na.omit)
```

Для задач класифікації з числом класів $k > 2$, *libsvm* використовує послідовний підхід, в якому відбувається навчання бінарних класифікаторів. Відповідний клас знаходиться з використанням схеми голосування.

libsvm використовує розріджене представлення даних, яке також підтримується пакетом *SparseM*.

Результатом роботи функції є об'єкт класу 'svm', що містить навчену модель, яка включає в себе:

SV – опорні вектори (можливо масштабовані);

index – індексний вектор, що визначає опорні вектори в матриці вихідних даних. Дані індекси належать до вже оброблених вихідними даними (після використання *na.omit* та *subset*);

coefs – коефіцієнти відповідають поточним ярликам;

rho – негативні перехоплення;

sigma – у разі ймовірнісної регресійної моделі, масштабуючий параметр гіпотетичного $(1 - 0)$ розподілу Лапласа, оцінений за допомогою максимальної правдоподібності;

probA, *probB* – числовий вектор довжини $k(k - 1)/2$ (k – число класів), що містить параметри логістичних розподілів,

Індивідуальне завдання

5. Серед ядер типів *polynomial*, *radial*, *sigmoid* вибрати оптимальне в сенсі кількості помилок на тестовій вибірці. Змінюючи значення параметра *gamma*, продемонструвати ефект перенавчання моделі

Виконання завдання

```
cat("\014")
data=read.table("C:\Program
Files\LAB_YEAR_5\MZKN\Lab_05\svmdata3.txt",header=TRUE)
x=subset(data,select=-Colors)
library(e1071)
model1=svm(data$Colors~.,data,kernel="polynomial",degree=5,gamma=0.35)
model1
call:
```

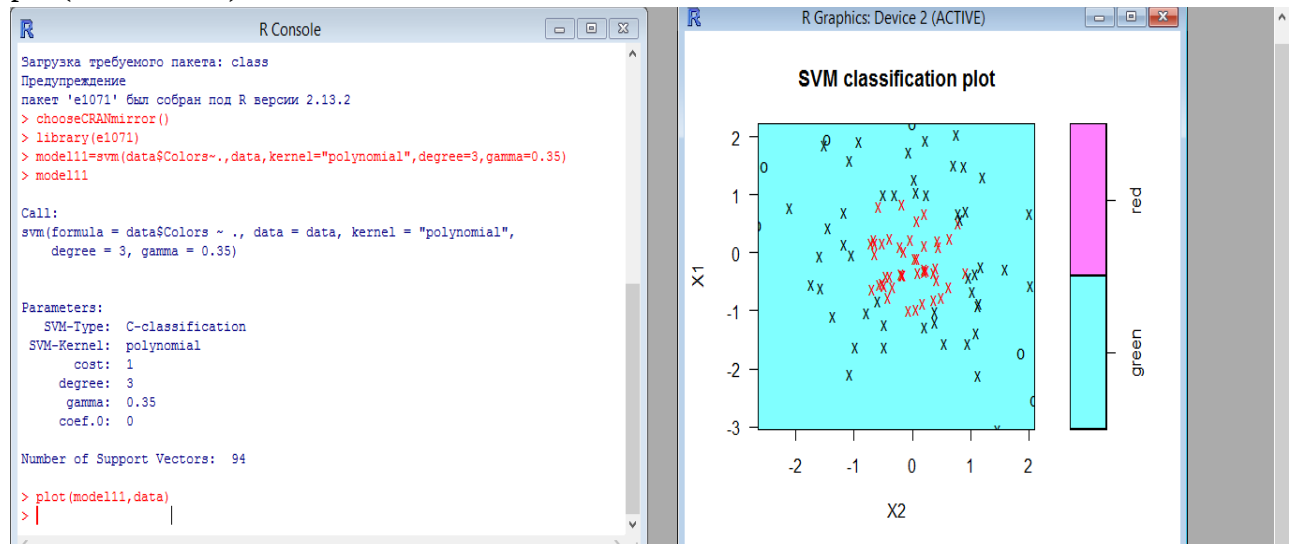
```
svm(formula = data$Colors ~ ., data = data, kernel = "polynomial", degree = 5, gamma = 0.35)
```

Parameters:

```
SVM-Type: C-classification
SVM-Kernel: polynomial
cost: 1
degree: 5
gamma: 0.35
coef.0: 0
```

Number of Support Vectors: 116

```
plot(model1,data)
```

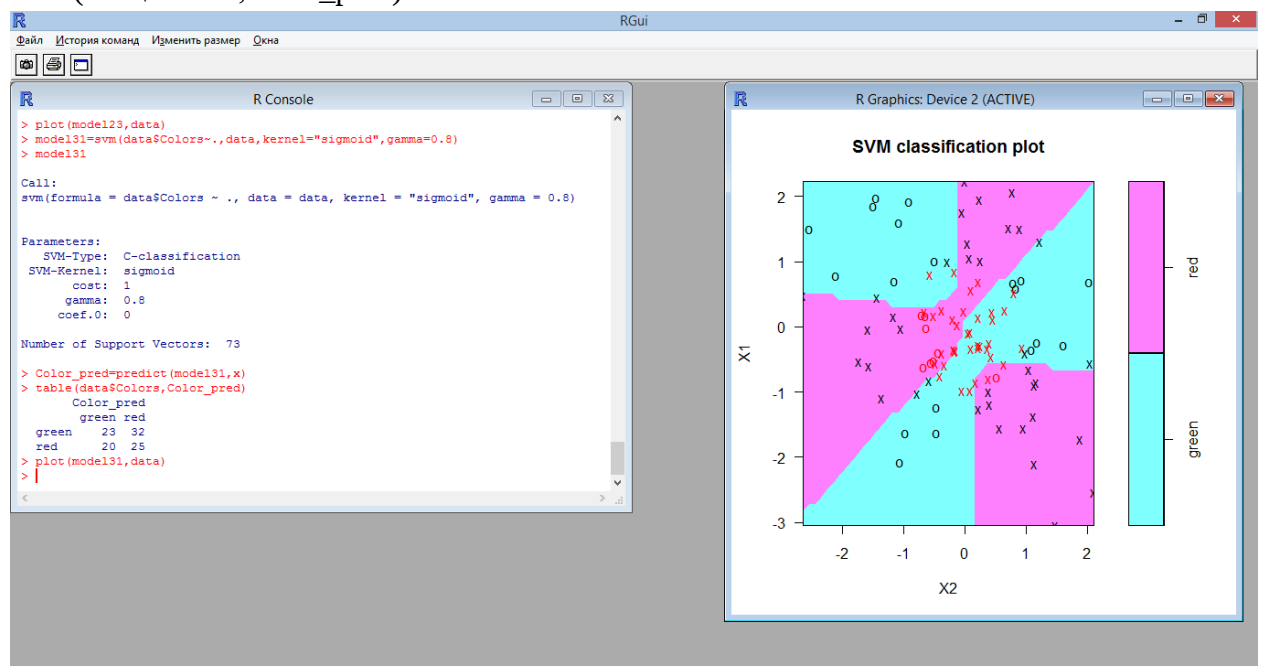


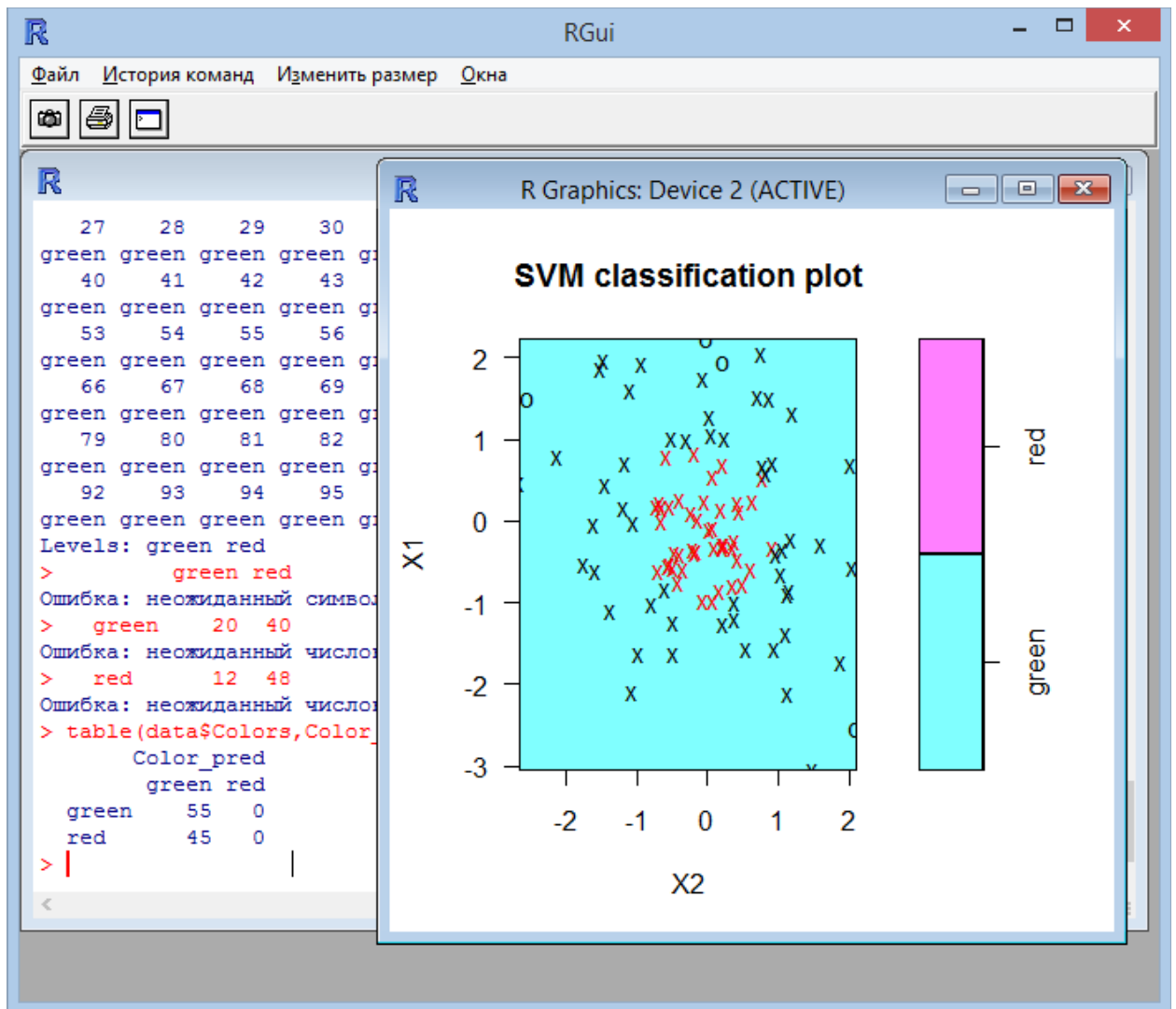
```
Color_pred=predict(model1,x)
```

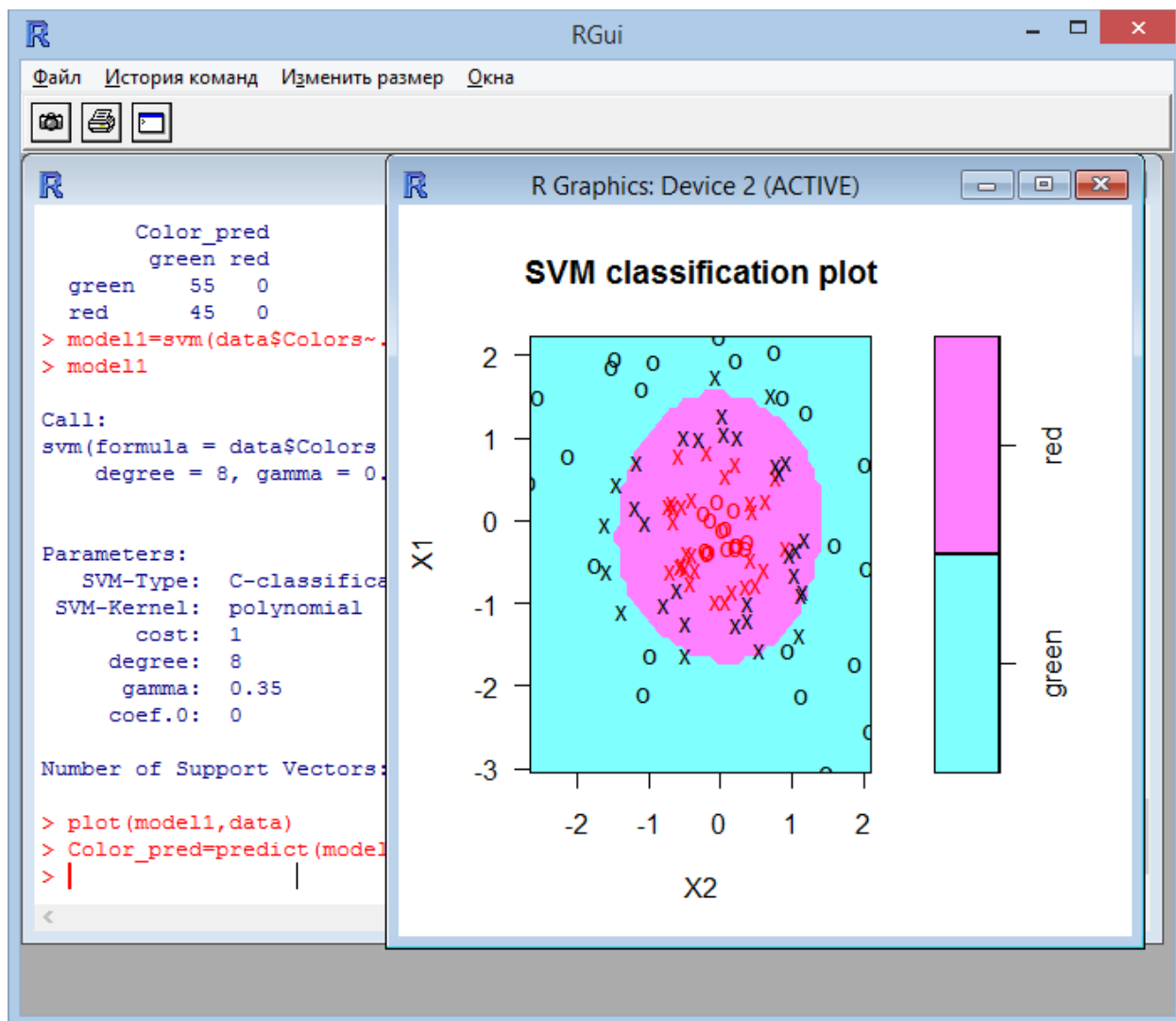
```
Color_pred
```

```
green red
green 20 40
red 12 48
```

```
table(data$Colors,Color_pred)
```



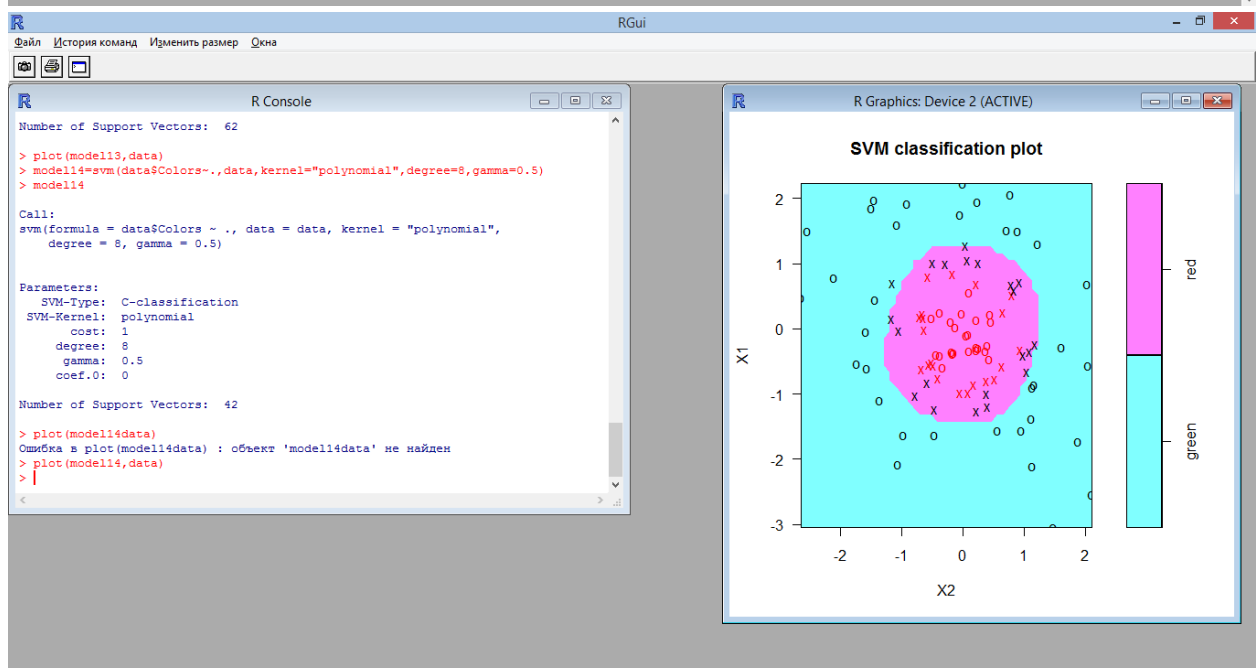
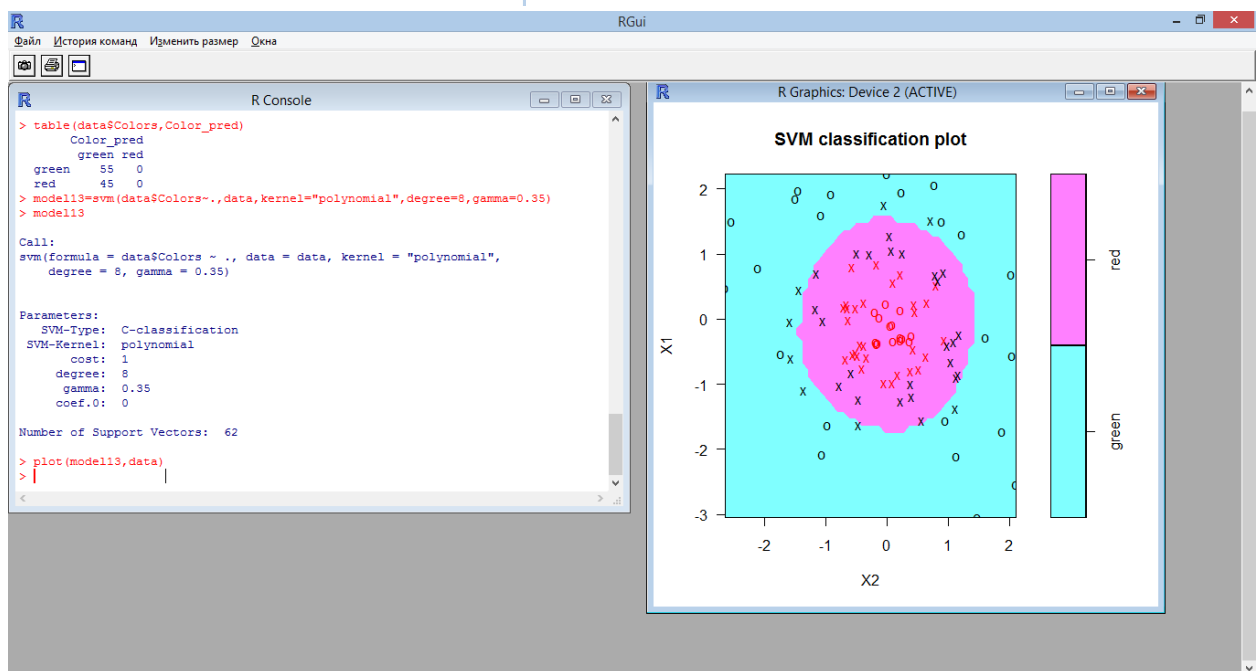
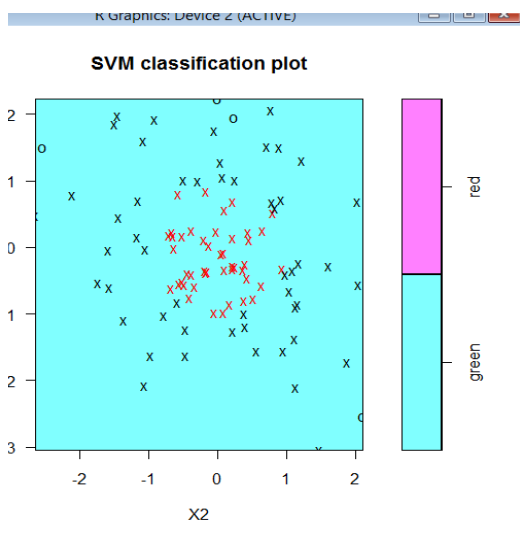




```

Color_pred
 1  2  3  4  5  6  7  8  9 10 11 12 13
green green green green green green green green green green green green green
14 15 16 17 18 19 20 21 22 23 24 25 26
green green green green green green green green green green green green green
27 28 29 30 31 32 33 34 35 36 37 38 39
green green green green green green green green green green green green green
40 41 42 43 44 45 46 47 48 49 50 51 52
green green green green green green green green green green green green green
53 54 55 56 57 58 59 60 61 62 63 64 65
green green green green green green green green green green green green green
66 67 68 69 70 71 72 73 74 75 76 77 78
green green green green green green green green green green green green green
79 80 81 82 83 84 85 86 87 88 89 90 91
green green green green green green green green green green green green green
92 93 94 95 96 97 98 99 100
green green green green green green green green green green
Levels: green red
table(data$Colors,Color_pred)
      green 55 0
      red 45 0

```



model2=svm(data\$Colors~.,data,kernel="radial",gamma=0.6)

model2

Call:

```
svm(formula = data$Colors ~ ., data = data, kernel = "radial", gamma = 0.6)
```

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 1

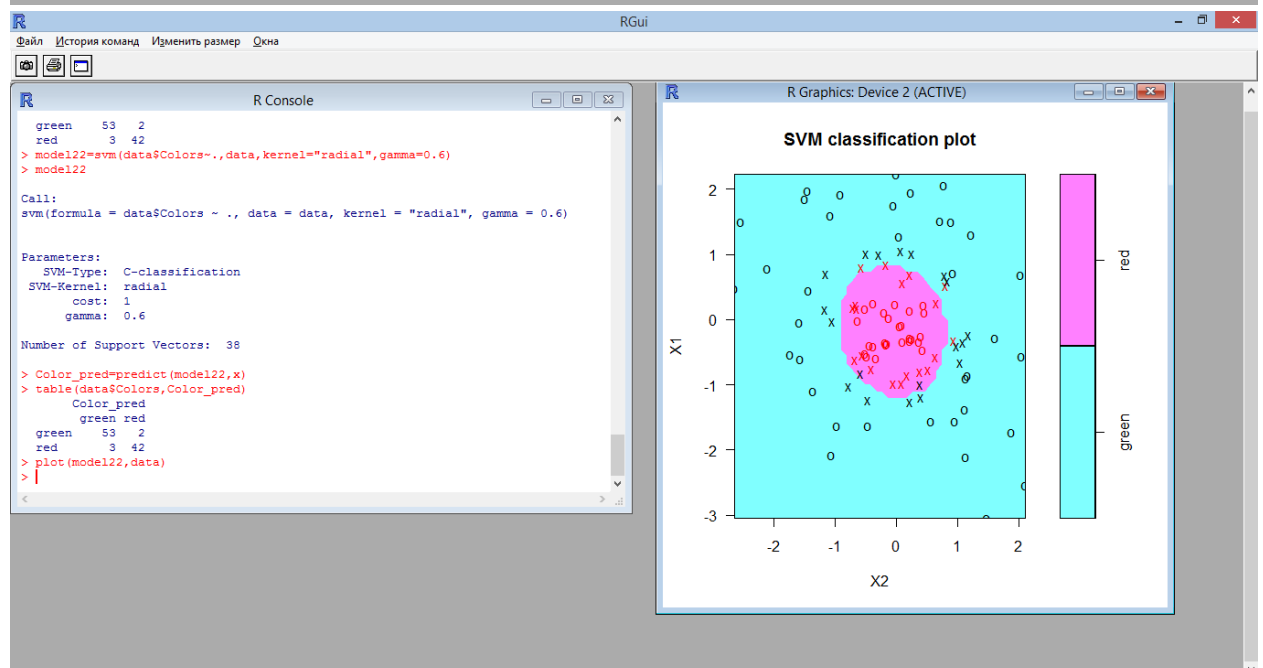
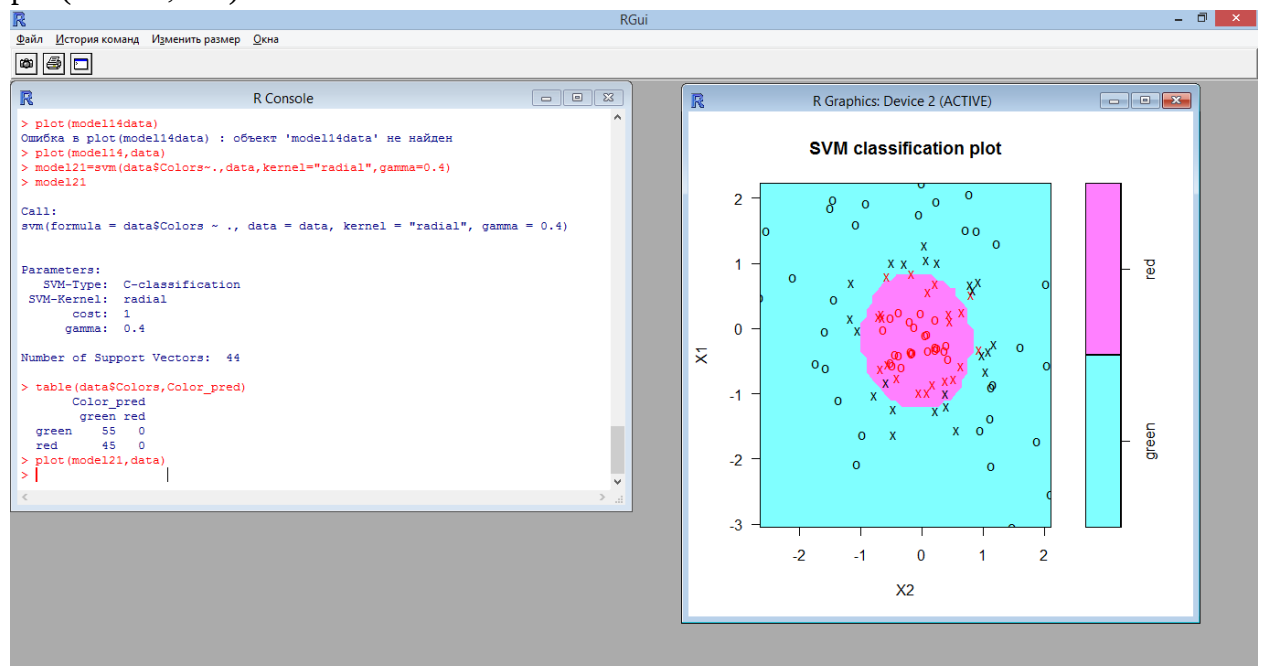
gamma: 0.6

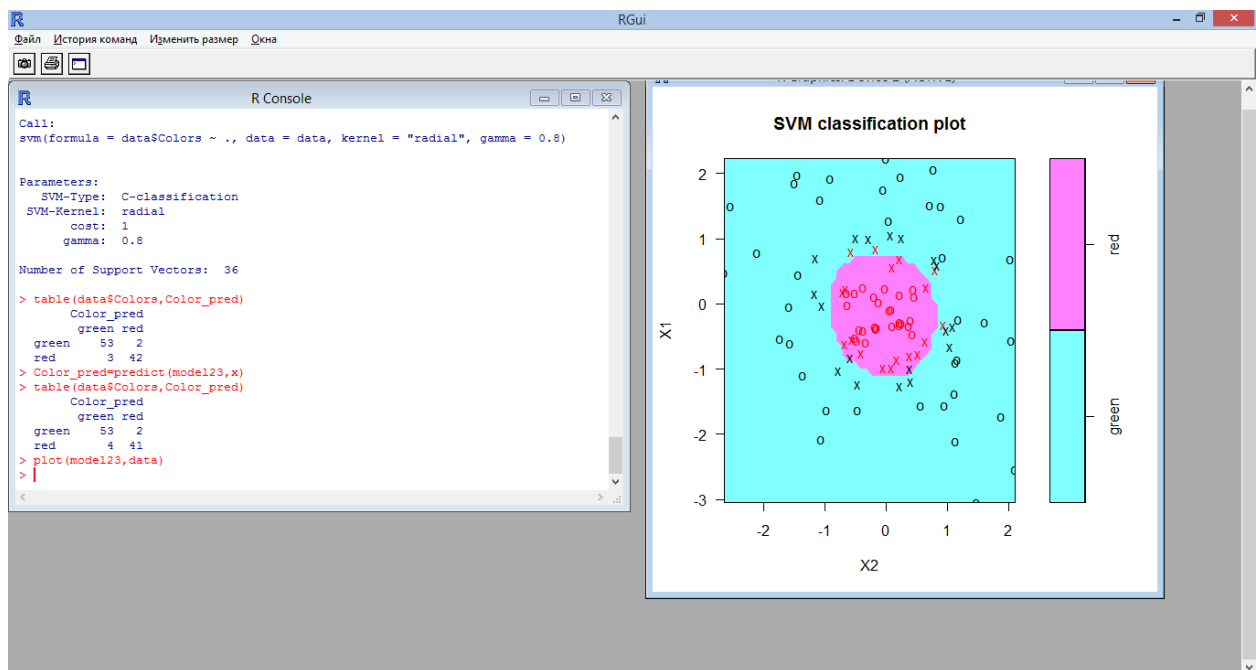
Number of Support Vectors: 51

```
table(data$Colors,Color_pred)
```

	Color_pred	
	green	red
green	60	0
red	4	56

```
plot(model2,data)
```





```
Color_pred=predict(model2,x)
```

```
table(data$Colors,Color_pred)
```

```
model3=svm(data$Colors~.,data,kernel="sigmoid",gamma=0.8)
```

```
model3
```

```
Call:
```

```
svm(formula = data$Colors ~ ., data = data, kernel = "sigmoid", gamma = 0.8)
```

```
Parameters:
```

```
  SVM-Type:  C-classification
```

```
  SVM-Kernel: sigmoid
```

```
    cost:    1
```

```
   gamma:   0.8
```

```
  coef.0:    0
```

```
Number of Support Vectors: 80
```

```
Color_pred=predict(model3,x)
```

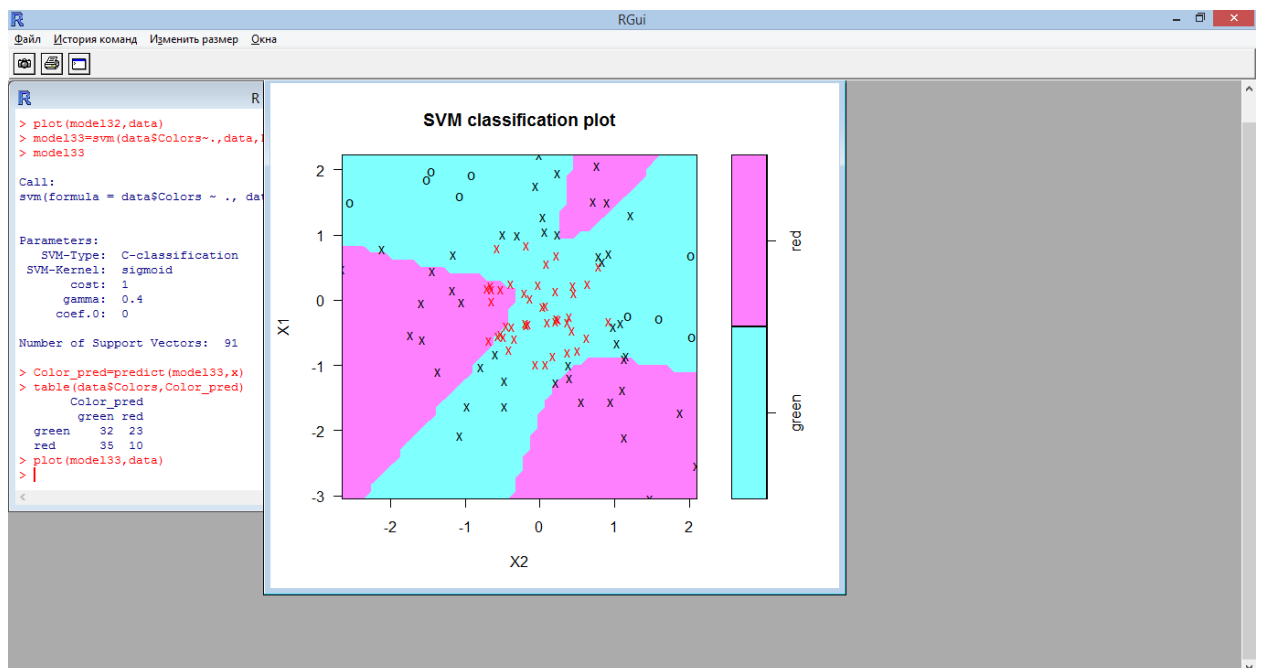
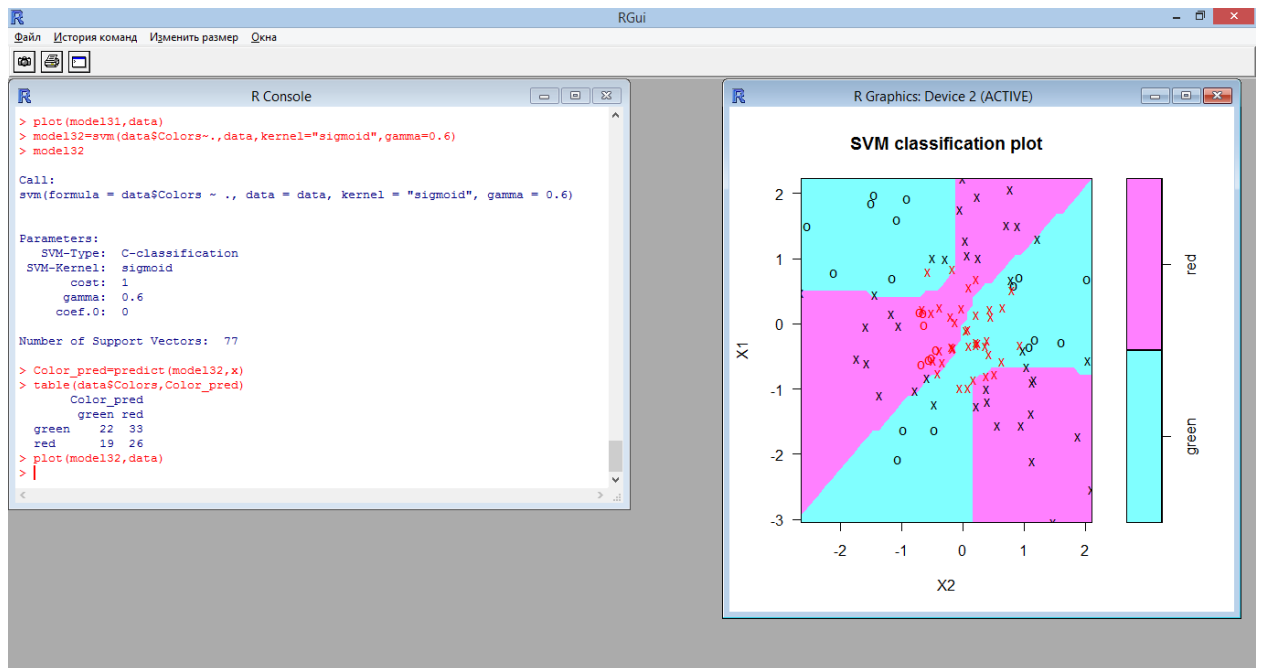
```
table(data$Colors,Color_pred)
```

```
  Color_pred
```

```
  green red
```

```
green    29  31
```

```
red      35  25
```

```
data=read.table("/home/taras/Документы/LAB_YEAR_5/MZKN_Pukach/Lab_05/svmdata3.txt",
,header=TRUE)
x=subset(data,select=-Colors)
library(e1071)
chooseCRANmirror()
library(e1071)
model11=svm(data$Colors~.,data,kernel="polynomial",degree=3,gamma=0.35)
model11
```

```
Call:
svm(formula = data$Colors ~ ., data = data, kernel = "polynomial",
    degree = 3, gamma = 0.35)
```

```
Parameters:
  SVM-Type: C-classification
  SVM-Kernel: polynomial
    cost: 1
    degree: 3
    gamma: 0.35
    coef.0: 0
```

```
Number of Support Vectors: 94
```

```
plot(model11,data)
Color_pred=predict(model11,x)
```

```
Color_pred
  1    2    3    4    5    6    7    8    9   10   11   12   13
green green green green green green green green green green green green
 14   15   16   17   18   19   20   21   22   23   24   25   26
green green green green green green green green green green green green
 27   28   29   30   31   32   33   34   35   36   37   38   39
green green green green green green green green green green green green
 40   41   42   43   44   45   46   47   48   49   50   51   52
green green green green green green green green green green green green
 53   54   55   56   57   58   59   60   61   62   63   64   65
green green green green green green green green green green green green
 66   67   68   69   70   71   72   73   74   75   76   77   78
green green green green green green green green green green green green
 79   80   81   82   83   84   85   86   87   88   89   90   91
green green green green green green green green green green green green
 92   93   94   95   96   97   98   99  100
green green green green green green green green green
Levels: green red
green red
```

```
green
```

```
table(data$Colors,Color_pred)
  Color_pred
    green red
green    55  0
red     45  0
Color_pred
  1    2    3    4    5    6    7    8    9   10   11   12   13
green green green green green green green green green green green green
 14   15   16   17   18   19   20   21   22   23   24   25   26
green green green green green green green green green green green green
 27   28   29   30   31   32   33   34   35   36   37   38   39
green green green green green green green green green green green green
 40   41   42   43   44   45   46   47   48   49   50   51   52
green green green green green green green green green green green green
 53   54   55   56   57   58   59   60   61   62   63   64   65
green green green green green green green green green green green green
 66   67   68   69   70   71   72   73   74   75   76   77   78
green green green green green green green green green green green green
 79   80   81   82   83   84   85   86   87   88   89   90   91
green green green green green green green green green green green green
 92   93   94   95   96   97   98   99  100
green green green green green green green green green
Levels: green red
```

green red

Ошибка: неожиданный символ в " green red"

```
model12=svm(data$Colors~.,data,kernel="polynomial",degree=5,gamma=0.35)
```

model12

Call:

```
svm(formula = data$Colors ~ ., data = data, kernel = "polynomial",
     degree = 5, gamma = 0.35)
```

Parameters:

```
SVM-Type: C-classification
SVM-Kernel: polynomial
cost: 1
degree: 5
gamma: 0.35
coef.0: 0
```

Number of Support Vectors: 96

```
plot(model12,data)
```

```
plot(model12,data)
```

```
Color_pred=predict(model12,x)
```

```
table(data$Colors,Color_pred)
```

	Color_pred	
	green	red
green	55	0
red	45	0

```
model13=svm(data$Colors~.,data,kernel="polynomial",degree=8,gamma=0.35)
```

model13

Call:

```
svm(formula = data$Colors ~ ., data = data, kernel = "polynomial",
     degree = 8, gamma = 0.35)
```

Parameters:

```
SVM-Type: C-classification
SVM-Kernel: polynomial
cost: 1
degree: 8
gamma: 0.35
coef.0: 0
```

Number of Support Vectors: 62

```
plot(model13,data)
```

```
Color_pred=predict(model12,x)
```

```
table(data$Colors,Color_pred)
```

	Color_pred	
	green	red
green	55	0
red	45	0

```
Color_pred=predict(model13,x)
```

```
table(data$Colors,Color_pred)
```

	Color_pred	
	green	red
green	30	25
red	0	45

```
model14=svm(data$Colors~.,data,kernel="polynomial",degree=8,gamma=0.5)
```

```
model14
```

Call:

```
svm(formula = data$Colors ~ ., data = data, kernel = "polynomial",  
     degree = 8, gamma = 0.5)
```

Parameters:

```
SVM-Type: C-classification  
SVM-Kernel: polynomial  
cost: 1  
degree: 8  
gamma: 0.5  
coef.0: 0
```

Number of Support Vectors: 42

```
Color_pred=predict(model14,x)
```

```
table(data$Colors,Color_pred)
```

	Color_pred	
	green	red
green	35	20
red	0	45

```
plot(model14,data)
```

```
model21=svm(data$Colors~.,data,kernel="radial",gamma=0.4)
```

```
model21
```

Call:

```
svm(formula = data$Colors ~ ., data = data, kernel = "radial", gamma = 0.4)
```

Parameters:

```
SVM-Type: C-classification  
SVM-Kernel: radial  
cost: 1  
gamma: 0.4
```

Number of Support Vectors: 44

```
plot(model21,data)
```

```
Color_pred=predict(model21,x)
```

```
table(data$Colors,Color_pred)
```

```

      Color_pred
      green red
green      53  2
red        3 42
model22=svm(data$Colors~.,data,kernel="radial",gamma=0.6)
model22

```

Call:

```
svm(formula = data$Colors ~ ., data = data, kernel = "radial", gamma = 0.6)
```

Parameters:

```

SVM-Type:  C-classification
SVM-Kernel: radial
cost:      1
gamma:     0.6

```

Number of Support Vectors: 38

```

Color_pred=predict(model22,x)
table(data$Colors,Color_pred)
      Color_pred
      green red
green      53  2
red        3 42
plot(model22,data)
model23=svm(data$Colors~.,data,kernel="radial",gamma=0.8)
model23

```

Call:

```
svm(formula = data$Colors ~ ., data = data, kernel = "radial", gamma = 0.8)
```

Parameters:

```

SVM-Type:  C-classification
SVM-Kernel: radial
cost:      1
gamma:     0.8

```

Number of Support Vectors: 36

```

table(data$Colors,Color_pred)
      Color_pred
      green red
green      53  2
red        3 42
Color_pred=predict(model23,x)
table(data$Colors,Color_pred)
      Color_pred
      green red
green      53  2

```

```

      red      4  41
plot(model23,data)
model31=svm(data$Colors~.,data,kernel="sigmoid",gamma=0.8)
model31

```

Call:

```
svm(formula = data$Colors ~ ., data = data, kernel = "sigmoid", gamma = 0.8)
```

Parameters:

```

SVM-Type:  C-classification
SVM-Kernel:  sigmoid
cost:      1
gamma:     0.8
coef.0:    0

```

Number of Support Vectors: 73

```

Color_pred=predict(model31,x)
table(data$Colors,Color_pred)

```

	Color_pred	
	green	red
green	23	32
red	20	25

```

plot(model31,data)
model32=svm(data$Colors~.,data,kernel="sigmoid",gamma=0.6)
model32

```

Call:

```
svm(formula = data$Colors ~ ., data = data, kernel = "sigmoid", gamma = 0.6)
```

Parameters:

```

SVM-Type:  C-classification
SVM-Kernel:  sigmoid
cost:      1
gamma:     0.6
coef.0:    0

```

Number of Support Vectors: 77

```

Color_pred=predict(model32,x)
table(data$Colors,Color_pred)

```

	Color_pred	
	green	red
green	22	33
red	19	26

```

plot(model32,data)
model33=svm(data$Colors~.,data,kernel="sigmoid",gamma=0.4)
model33

```

Call:

```
svm(formula = data$Colors ~ ., data = data, kernel = "sigmoid", gamma = 0.4)
```

Parameters:

```
SVM-Type: C-classification
SVM-Kernel: sigmoid
cost: 1
gamma: 0.4
coef.0: 0
```

Number of Support Vectors: 91

```
Color_pred=predict(model33,x)
```

```
table(data$Colors,Color_pred)
```

	Color_pred	
	green	red
green	32	23
red	35	10

```
plot(model33,data)
```

Висновок

виконуючи лабораторну роботу, оволодів методом опорних векторів в задачах класифікації та відновлення регресії в системі R. Визначив, що збільшення значення числа gamma призводить до зменшення числа векторів, проте число помилок збільшується. Тобто необхідно змінювати значення gamma, коли потрібно досягти найкращого результату по кількості векторів або по числі помилок.