Тема: Логистическая регрессия.

Задачи:

1. Написать метод стохастического градиентного спуска.
2. Использовать “Bias Trick” для обучения параметра w0.
3. Написать логистический классификатор.
4. Построить карту классификации.
5. Построить разделяющие прямые во время обучения.

**Написать метод стохастического градиентного спуска.**

Метод градиентого спуска пишется очень быстро:

minimize **=** **function(**f,g,y,x,w,maxiter**=**100,nu**=**0.01,callback**=**callback**)** **{**

l **=** dim**(**x**)[**1**]**

**for** **(**iter **in** 1**:**maxiter**)** **{**

w **=** w **-** nu **\*** g**(**y,x,w,sample**(**1**:**l,1**))**

print**(**sprintf**(**"%d %f", iter, f**(**y,x,w**)))**

callback**(**w**)**

**}**

w

**}**

**Использовать “Bias Trick” для обучения параметра w0.**

Напомним что количество очков, даваемое классу, определяется как:

****

Тогда создадим вектор

****

И обучим с решающим правилом:

****

Тогда в итоге получим:

****

**Написать логистический классификатор.**

Обучив параметры *w*, можно написать решающее правило:

class **=** **function(**w,x**)** **{**

sign**(**sum**(**w**\***x**))**

**}**

a **=** **function(**w, x**)** **{**

x**[**n**+**1**]** **=** 1

c **=** class**(**w,x**)**

**if** **(**c **==** 1**)** **{**

return**(**1**)**

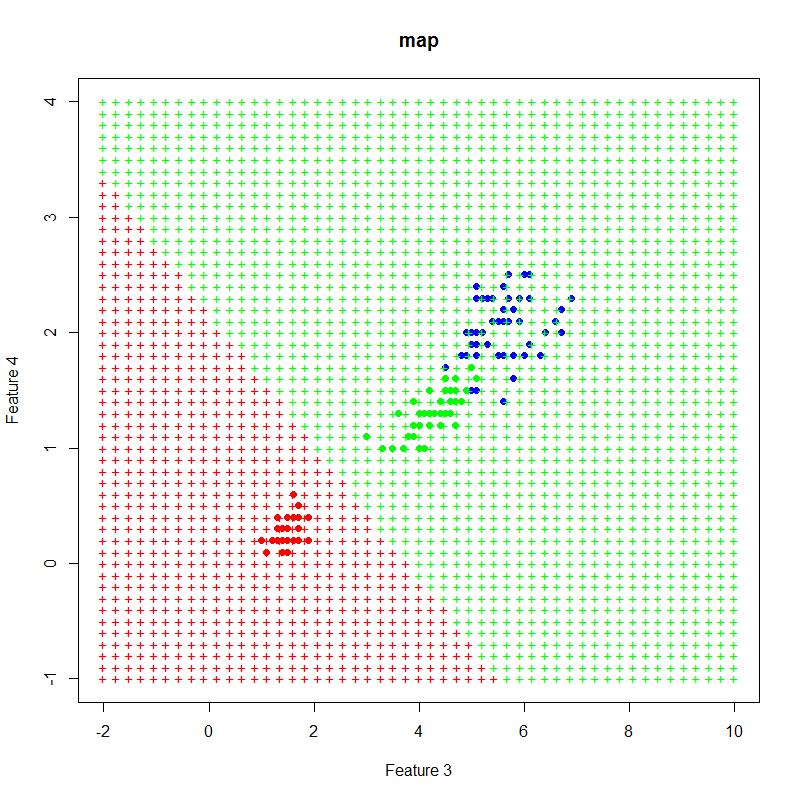
**}** **else** **{**

return**(**2**)**

**}**

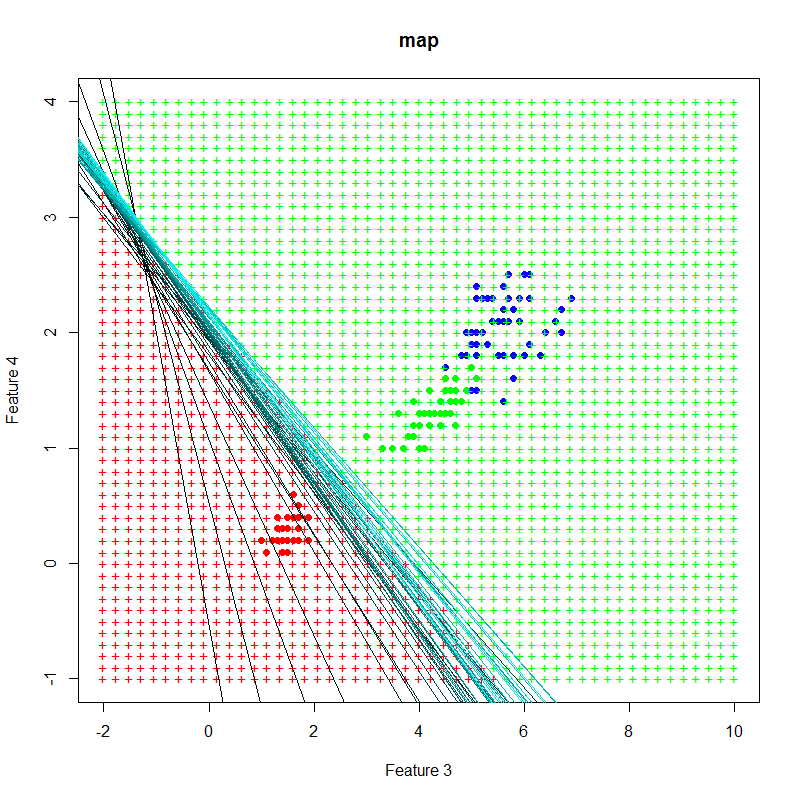
**}**

**Построить карту классификации.**



Это классификационная карта для задачи “setosa-vs-all”

**Построить разделяющие прямые во время обучения**



Циановые прямые – это прямые на более поздних итерациях градиентного спуска.

Выводы:

* Логистическая регрессия – хороший алгоритм.
* Поставленные задачи достигнуты.