Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Машинное обучение (МО)

Отчет о лабораторной работе №2

«Логистическая регрессия. Многоклассовая классификация»

|  |
| --- |
| **Выполнил:** |
| **Студент гр. 858341** |
| **Немкович А. В.** |

|  |
| --- |
| **Проверил:** |
| **Стержанов М. В.** |

**Минск 2019**

Содержание

[Постановка задачи 3](#_Toc25275667)

[Ход выполнения 5](#_Toc25275668)

[Выводы 13](#_Toc25275669)

# Постановка задачи

Набор данных **ex2data1.txt** представляет собой текстовый файл, содержащий информацию об оценке студента по первому экзамену (первое число в строке), оценке по второму экзамену (второе число в строке) и поступлении в университет (0 - не поступил, 1 - поступил).

Набор данных **ex2data2.txt** представляет собой текстовый файл, содержащий информацию о результате первого теста (первое число в строке) и результате второго теста (второе число в строке) изделий и результате прохождения контроля (0 - контроль не пройден, 1 - контроль пройден).

Набор данных **ex2data3.mat** представляет собой файл формата \*.mat (т.е. сохраненного из Matlab). Набор содержит 5000 изображений 20x20 в оттенках серого. Каждый пиксель представляет собой значение яркости (вещественное число). Каждое изображение сохранено в виде вектора из 400 элементов. В результате загрузки набора данных должна быть получена матрица 5000x400. Далее расположены метки классов изображений от 1 до 9 (соответствуют цифрам от 1 до 9), а также 10 (соответствует цифре 0).

**Задание.**

1. Загрузите данные **ex2data1.txt** из текстового файла.
2. Постройте график, где по осям откладываются оценки по предметам, а точки обозначаются двумя разными маркерами в зависимости от того, поступил ли данный студент в университет или нет.
3. Реализуйте функции потерь J(θ) и градиентного спуска для логистической регрессии с использованием векторизации.
4. Реализуйте другие методы (как минимум 2) оптимизации для реализованной функции стоимости (например, Метод Нелдера — Мида, Алгоритм Бройдена — Флетчера — Гольдфарба — Шанно, генетические методы и т.п.). Разрешается использовать библиотечные реализации методов оптимизации (например, из библиотеки scipy).
5. Реализуйте функцию предсказания вероятности поступления студента в зависимости от значений оценок по экзаменам.
6. Постройте разделяющую прямую, полученную в результате обучения модели. Совместите прямую с графиком из пункта 2.
7. Загрузите данные **ex2data2.txt** из текстового файла.
8. Постройте график, где по осям откладываются результаты тестов, а точки обозначаются двумя разными маркерами в зависимости от того, прошло ли изделие контроль или нет.
9. Постройте все возможные комбинации признаков x1 (результат первого теста) и x2 (результат второго теста), в которых степень полинома не превышает 6, т.е. 1, x1, x2, x12, x1x2, x22, …, x1x25, x26 (всего 28 комбинаций).
10. Реализуйте L2-регуляризацию для логистической регрессии и обучите ее на расширенном наборе признаков методом градиентного спуска.
11. Реализуйте другие методы оптимизации.
12. Реализуйте функцию предсказания вероятности прохождения контроля изделием в зависимости от результатов тестов.
13. Постройте разделяющую кривую, полученную в результате обучения модели. Совместите прямую с графиком из пункта 7.
14. Попробуйте различные значения параметра регуляризации λ. Как выбор данного значения влияет на вид разделяющей кривой? Ответ дайте в виде графиков.
15. Загрузите данные **ex2data3.mat** из файла.
16. Визуализируйте несколько случайных изображений из набора данных. Визуализация должна содержать каждую цифру как минимум один раз.
17. Реализуйте бинарный классификатор с помощью логистической регрессии с использованием векторизации (функции потерь и градиентного спуска).
18. Добавьте L2-регуляризацию к модели.
19. Реализуйте многоклассовую классификацию по методу “один против всех”.
20. Реализуйте функцию предсказания класса по изображению с использованием обученных классификаторов.
21. Процент правильных классификаций на обучающей выборке должен составлять около 95%.
22. Ответы на вопросы представьте в виде отчета.

# Ход выполнения

**Задания 1-2**: Загрузка данных ex2data1.txt из текстового файла и построение графика, где по осям откладываются оценки по предметам, а точки обозначаются двумя разными маркерами в зависимости от того, поступил ли данный студент в университет или нет

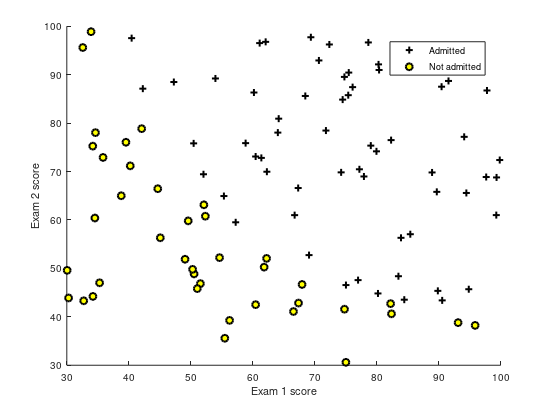


Рис 1. – точечный график тренировочного набора (ex2data1.txt)

**Задание 3**: Реализация функции потерь J(θ) и градиентного спуска для логистической регрессии с использованием векторизации.

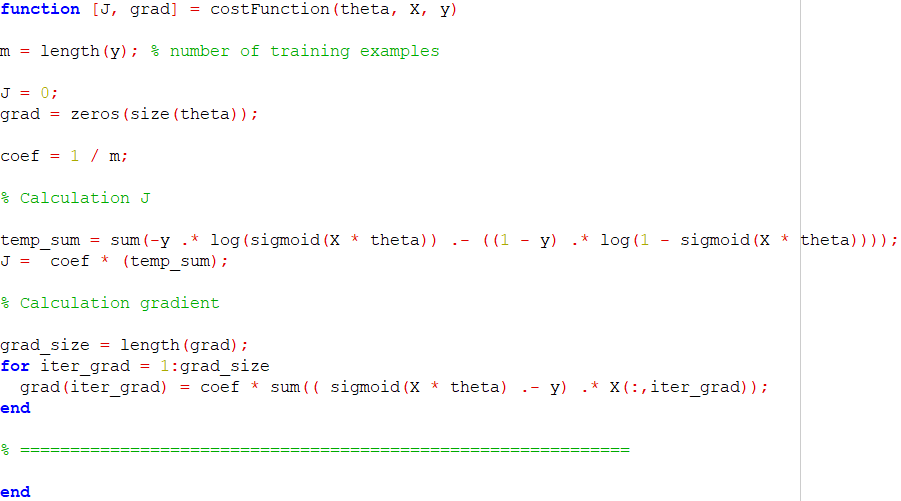


Рис 2. – Код функции потерь и градиентного спуска

**Задание 5**: Реализация функции предсказания вероятности поступления студента в зависимости от значений оценок по экзаменам.

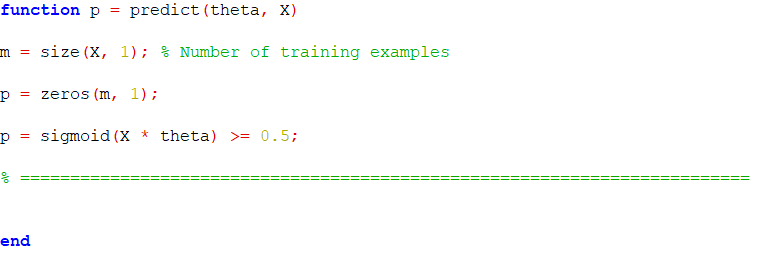


Рис 3. – Код функции предсказания вероятности

**Задание 6**: Постройте разделяющую прямую, полученную в результате обучения модели. Совместите прямую с графиком из пункта 2.

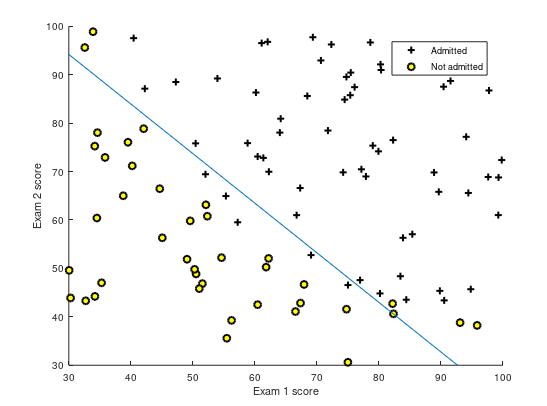


Рис 4. – тренировочный набор совместно с разделяющей прямой (ex2data1.txt)

**Задания 7-8**: загрузка данных и построение графика, где по осям откладываются результаты тестов, а точки обозначаются двумя разными маркерами в зависимости от того, прошло ли изделие контроль или нет.

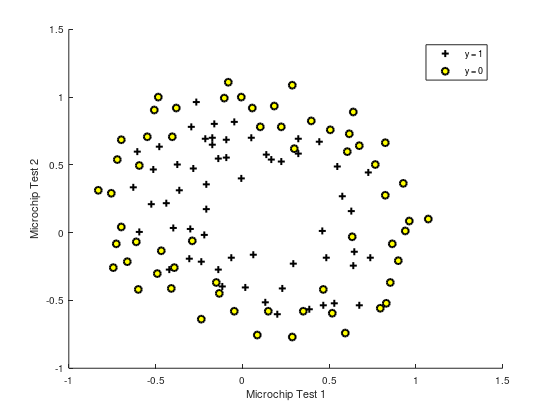


Рис 5. – точечный график тренировочного набора (ex2data2.txt)

**Задание 9**: Построение всех возможные комбинации признаков x1 (результат первого теста) и x2 (результат второго теста), в которых степень полинома не превышает 6.

Для построения всех возможных комбинаций использовалась функция mapFeature:

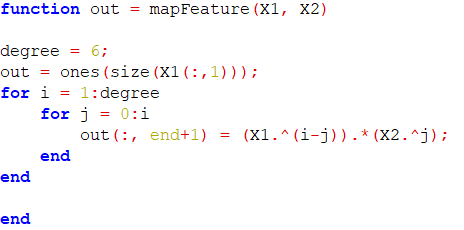
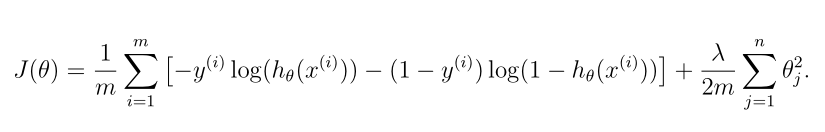


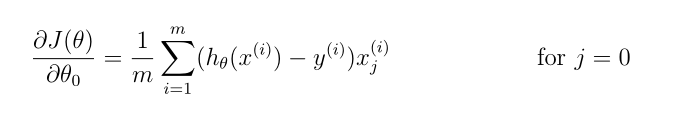
Рис 6. – Код функции mapFeature

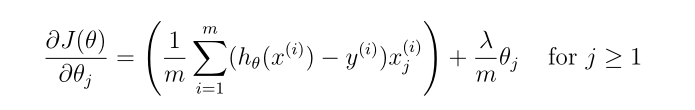
**Задание 10**: Реализация L2-регуляризации для логистической регрессии и обучение ее на расширенном наборе признаков методом градиентного спуска.

Формула L2-регуляризации для функции потерь:



Формула L2-регуляризации для частных производных в градиентом спуске:





Для реализации L2-регуляризации была написана функция costFunctionReg:

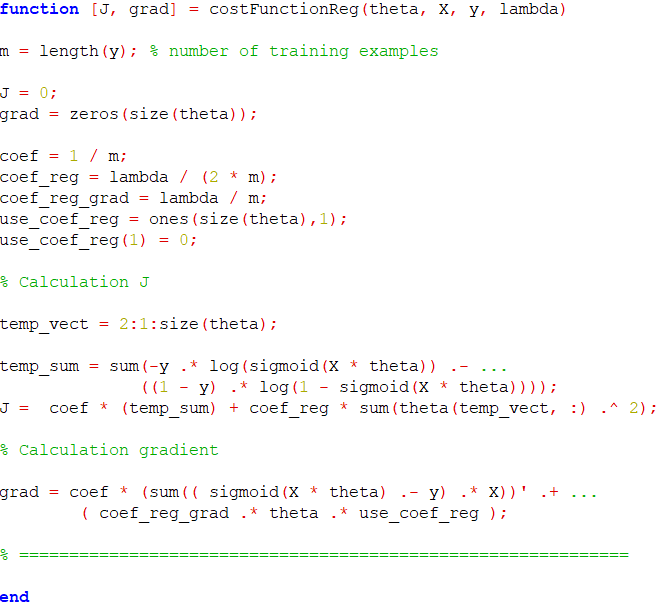


Рис 7. – Код функции costFunctionReg

**Задание 11**: Реализация функции предсказания вероятности прохождения контроля изделием в зависимости от результатов тестов.

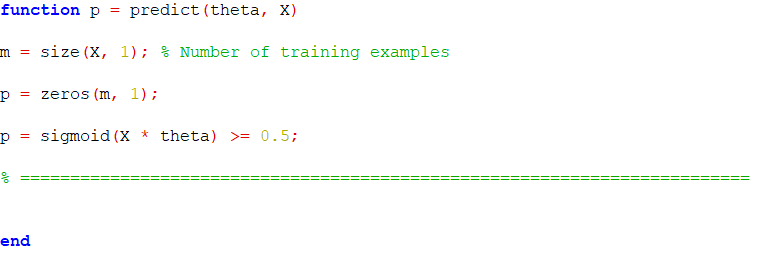


Рис 8. – Код функции предсказания вероятности

**Задание 13**: Построение разделяющей кривой, полученной в результате обучения модели. Совмещение прямой с графиком из пункта 7.

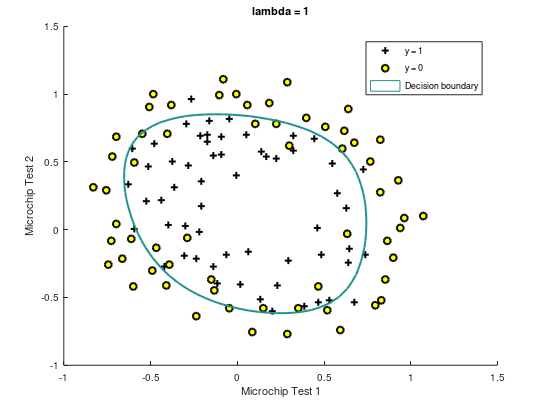


Рис 9. – тренировочный набор совместно с разделяющей прямой (ex2data2.txt)

**Задание 14**: Использование различных значений параметра регуляризации λ.

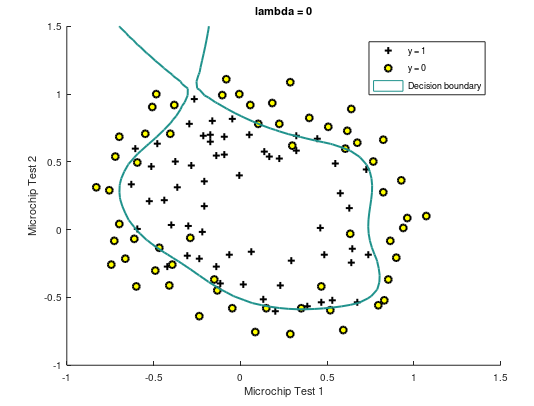


Рис 10. – модель с параметром λ = 0 (переобучение)

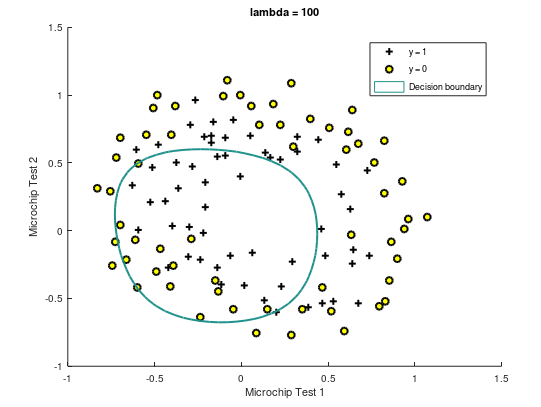


Рис 11. – модель с параметром λ = 100 (недообучение)

**Задания 15-16**: Загрузка данных и визуализация несколько случайных изображений из набора данных.

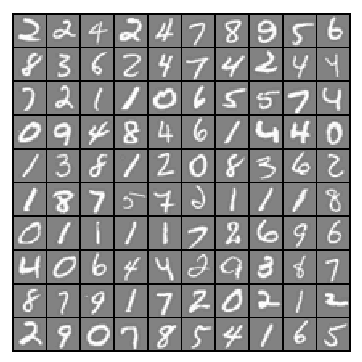


Рис 12. – визуализация данных из тренировочного набора (ex2data3.mat)

**Задания 17-18**: Реализация бинарного классификатора с помощью логистической регрессии с использованием векторизации (функции потерь и градиентного спуска). Добавление L2-регуляризацию к модели.

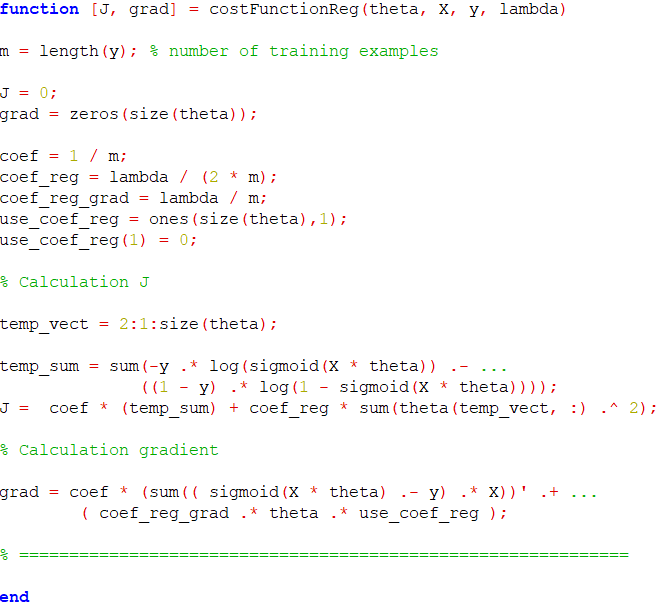


Рис 13. – Код функции costFunctionReg

**Задание 19**: Реализуйте многоклассовую классификацию по методу “один против всех”.

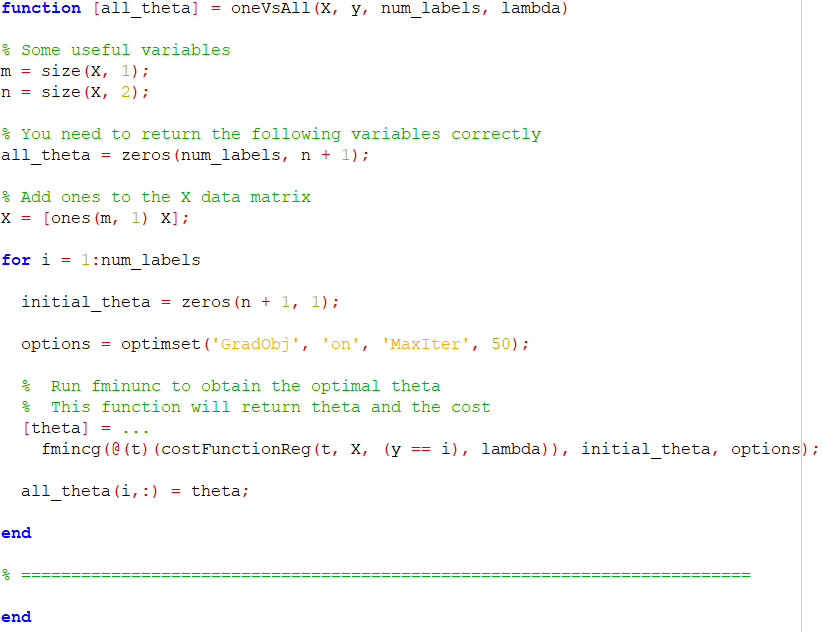


Рис 14. – Код функции oneVsAll

**Задание 20:** Реализация функции предсказания класса по изображению с использованием обученных классификаторов.

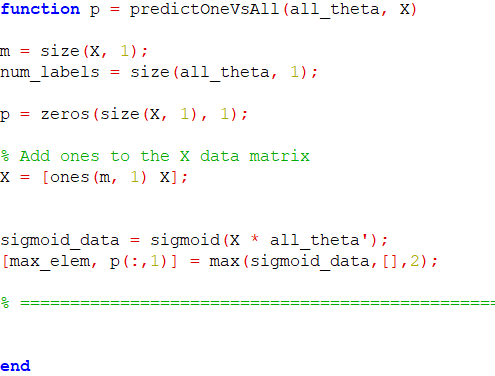


Рис 15. – Код функции предсказания класса по изображению

**Задание 21**: Процент правильных классификаций на обучающей выборке должен составлять около 95%.

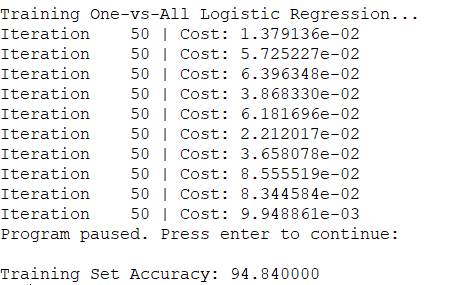


Рис 16. – Результат классификации на обучающей выборке

# Выводы

Мы научились решать задачу логической регрессии с помощью градиентного спуска. А также выполнять многоклассовую классификацию. В процессе мы изучили построение разделяющей границы. Использовали метод «один против всех» для многоклассовой классификации. Выяснили что для решения проблемы переобучения модели используются следующие подходы:

1. Уменьшение количества признаков

* ручной выбор какие признаки оставлять
* используя алгоритм выбора модели

2. Регуляризация

* сохраняет все признаки, но уменьшает величину параметров θ
* хорошо работает, когда имеем большое число полезных признаков