Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра информатики

Отчет по лабораторной работе:

**Лабораторная работа №1 “Линейная регрессия”**

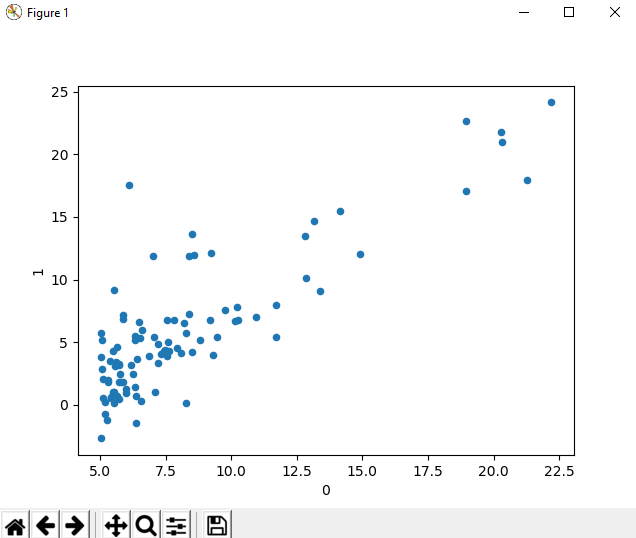
Выполнил: Карп Александр Игоревич

магистрант кафедры информатики

группа №858641

Минск 2019

1. Загрузите набор данных **ex1data1.txt** из текстового файла.  
   <https://github.com/Karpengold/machine_learning/blob/master/lab1/index.py#L125>
2. Постройте график зависимости прибыли ресторана от населения города, в котором он расположен:  
   <https://github.com/Karpengold/machine_learning/blob/master/lab1/index.py#L134>

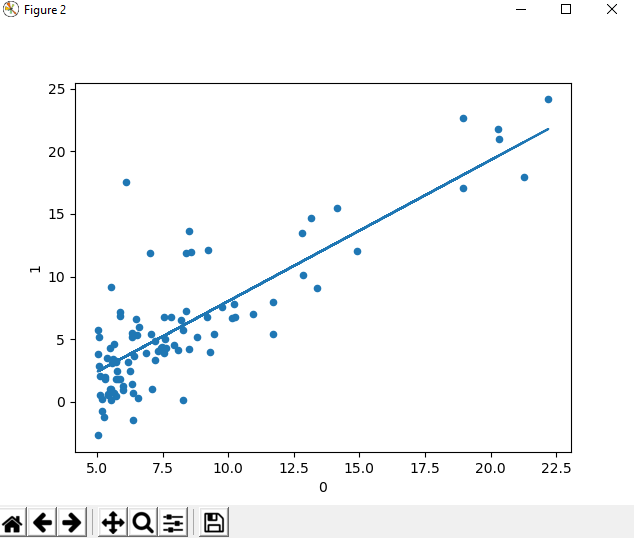


1. Реализуйте функцию потерь J(θ) для набора данных **ex1data1.txt**.

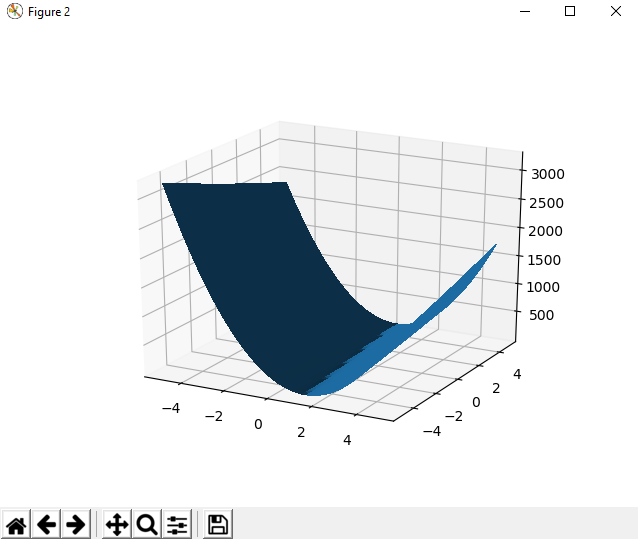
<https://github.com/Karpengold/machine_learning/blob/master/lab1/index.py#L138>

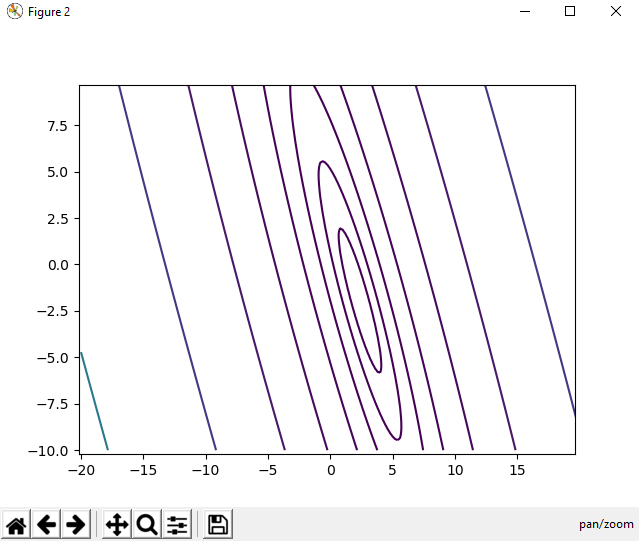
1. Реализуйте функцию градиентного спуска для выбора параметров модели. Постройте полученную модель (функцию) совместно с графиком из пункта 2.

<https://github.com/Karpengold/machine_learning/blob/master/lab1/index.py#L141>



1. Постройте трехмерный график зависимости функции потерь от параметров модели (θ0 и θ1) как в виде поверхности, так и в виде изолиний (contour plot).  
   <https://github.com/Karpengold/machine_learning/blob/master/lab1/index.py#L147>

****



1. Загрузите набор данных **ex1data2.txt** из текстового файла.  
   <https://github.com/Karpengold/machine_learning/blob/master/lab1/index.py#L152>
2. Произведите нормализацию признаков. Повлияло ли это на скорость сходимости градиентного спуска? Ответ дайте в виде графика.

<https://github.com/Karpengold/machine_learning/blob/master/lab1/index.py#L158>

Без нормализации алгоритм расходился, так что нормализация определенно повлияла на сходимость.

1. Реализуйте функции потерь J(θ) и градиентного спуска для случая многомерной линейной регрессии с использованием векторизации.

<https://github.com/Karpengold/machine_learning/blob/master/lab1/index.py#L166>

1. Покажите, что векторизация дает прирост производительности.  
   <https://github.com/Karpengold/machine_learning/blob/master/lab1/index.py#L174>

*Время сходимости алгоритма без использования векторизации:*

*С использованием:*

*Скорость сходимости с использованием векторизации увеличилась в 74 раза*

1. Постройте модель, используя аналитическое решение, которое может быть получено методом наименьших квадратов. Сравните результаты данной модели с моделью, полученной с помощью градиентного спуска.  
   <https://github.com/Karpengold/machine_learning/blob/master/lab1/index.py#L180>

*Результаты метода наименьших квадратов и метода градиентного спуска идентичны.*