

# Math practice 1

Уденеев Александр

October 2023

## 1 Цель

Нашей целью будет бинарная классификация какого-то объема данных

## 2 Обработка данных

В обработке будем использовать данные с сайта <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

В частности, файлы под названием "a1a" и "mushrooms".

В качестве функции потерь возьмем логистическую функцию потерь:

$$Q(x) = \sum_{i=1}^n \ln(1 + e^{-y_i \langle w, x_i \rangle}) \quad (1)$$

Так как при нахождении лучшей "угадывающей функции"  $w$  мы будем использовать линейный каплинг, то нам также нужен будет градиент функции потерь:

$$\frac{d}{dw} Q(x) = \sum_{i=1}^n \frac{-y_i x_i}{1 + e^{y_i \langle w, x_i \rangle}} \quad (2)$$

"Частные" данные, которые необходимы для обучения, мы будем брать из общих, выделяя какую-то их часть. Тестовые данные мы будем брать из частных, предварительно вырезав их.

Таким образом можно будет судить о верности проведенного обучения модели.

### 2.1 Обработка a1a

Для обработки возьмем количество итераций  $I = 1000$ , шаги  $\alpha_1 = 0.0001, \alpha_2 = 0.0001$ .

Будем менять момент от 0 до 1 с шагом 0.1 (Момент равный 0 соответствует случаю, когда мы опираемся только на общие данные).

В итоге получим такую таблицу.

<b>ala</b>				
momentum	accuracy	precision	recall	F1
0	0.891304	0.771429	0.613636	0.683544
0.1	0.873913	0.741935	0.522727	0.613333
0.2	0.873913	0.741935	0.522727	0.613333
0.3	0.873913	0.741935	0.522727	0.613333
0.4	0.878261	0.735294	0.568182	0.641026
0.5	0.878261	0.735294	0.568182	0.641026
0.6	0.882609	0.742857	0.590909	0.658228
0.7	0.886957	0.75	0.613636	0.675
0.8	0.886957	0.75	0.613636	0.675
0.9	0.886957	0.75	0.613636	0.675
1	0.886957	0.75	0.613636	0.675

Полученные функции потерь будут иметь вид.

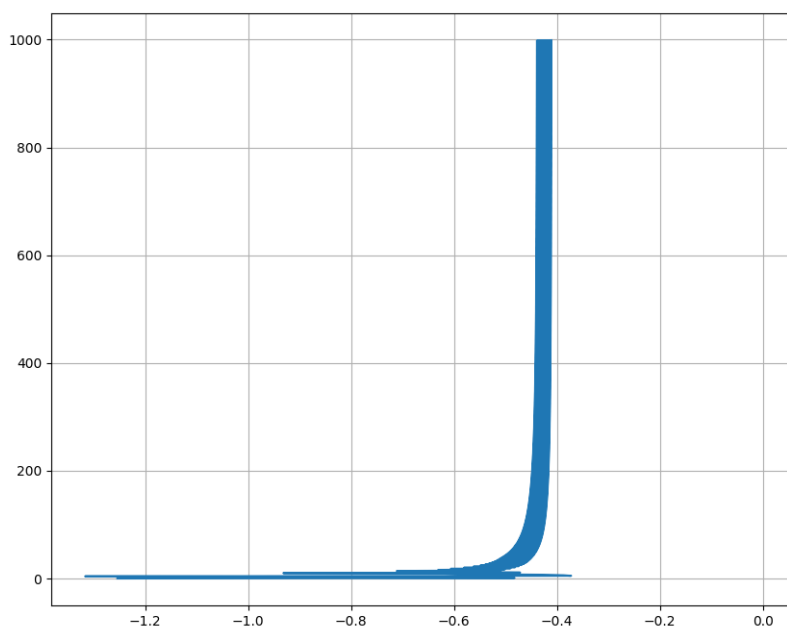


Рис. 1: momentum = 0

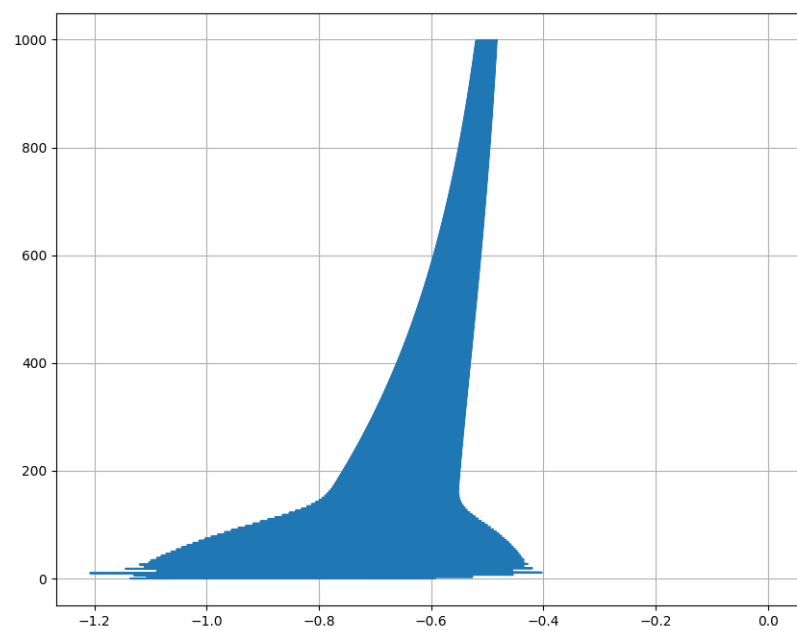


Рис. 2: momentum = 0.1

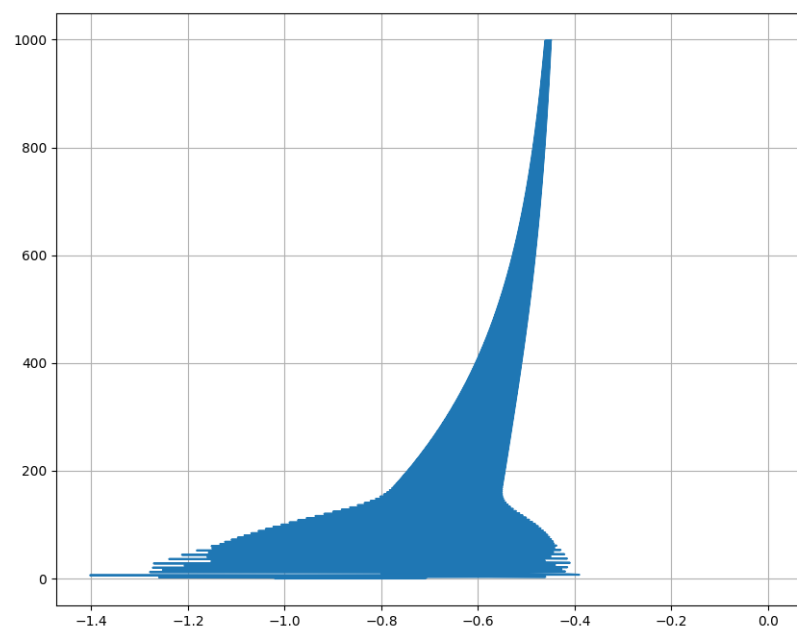


Рис. 3: momentum = 0.2

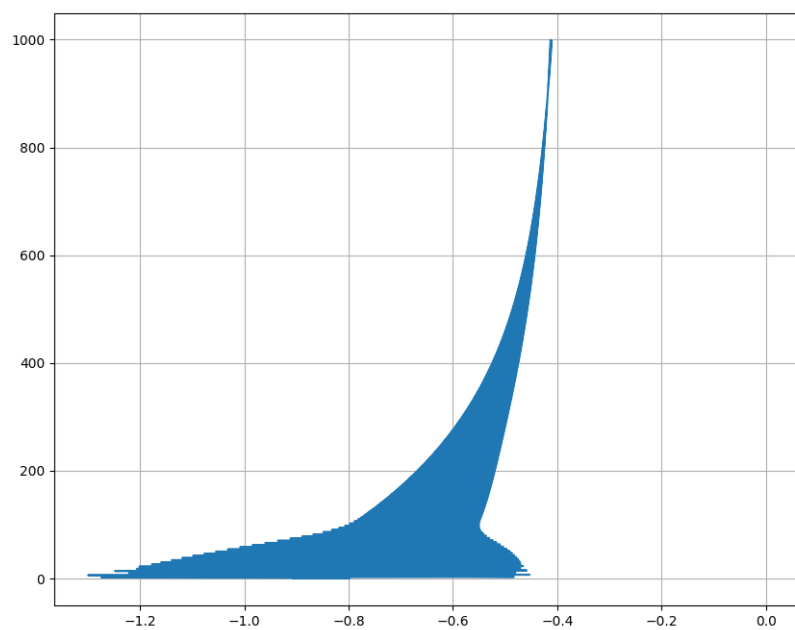


Рис. 4: momentum = 0.3

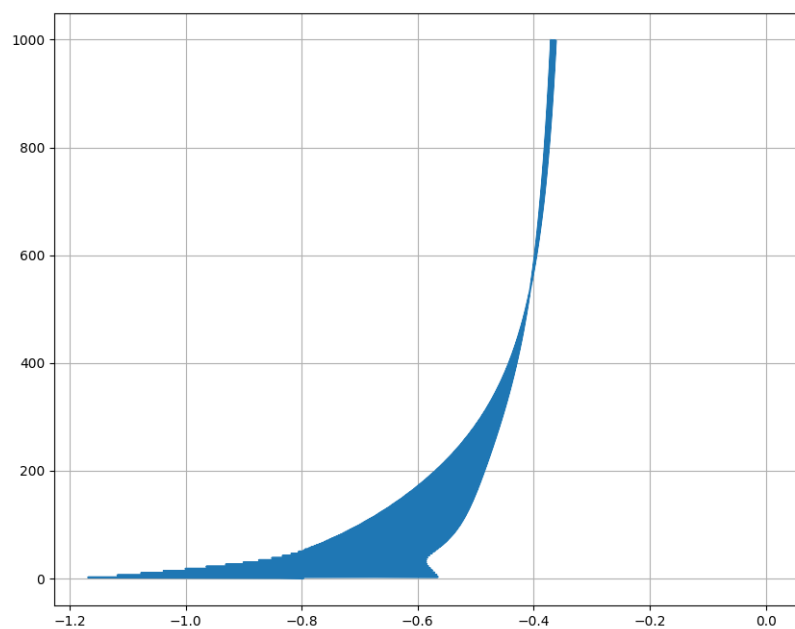


Рис. 5: momentum = 0.4

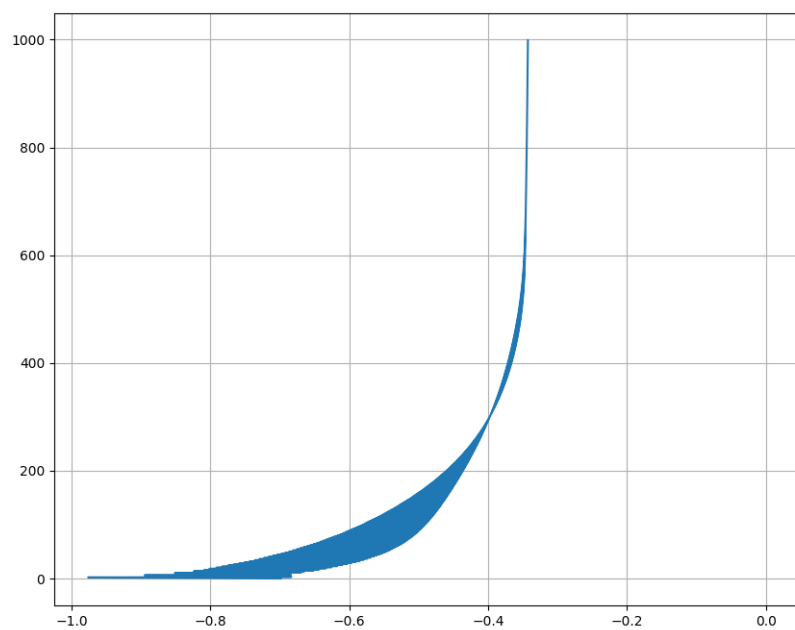


Рис. 6: momentum = 0.5

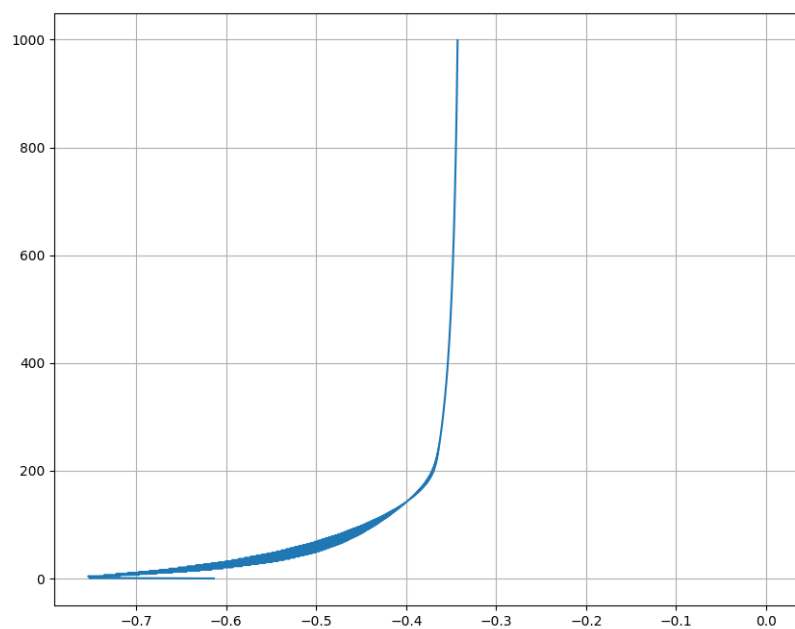


Рис. 7: momentum = 0.6

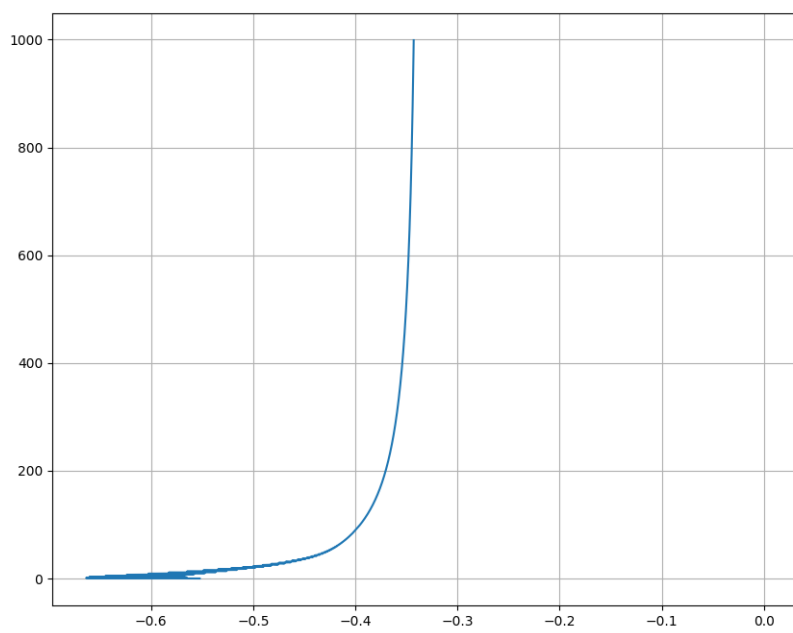


Рис. 8: momentum = 0.7

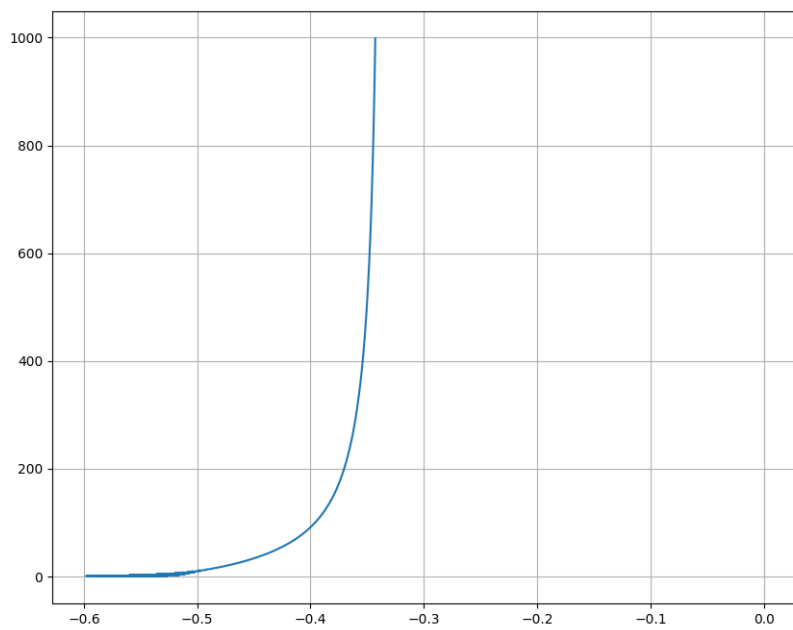


Рис. 9: momentum = 0.8

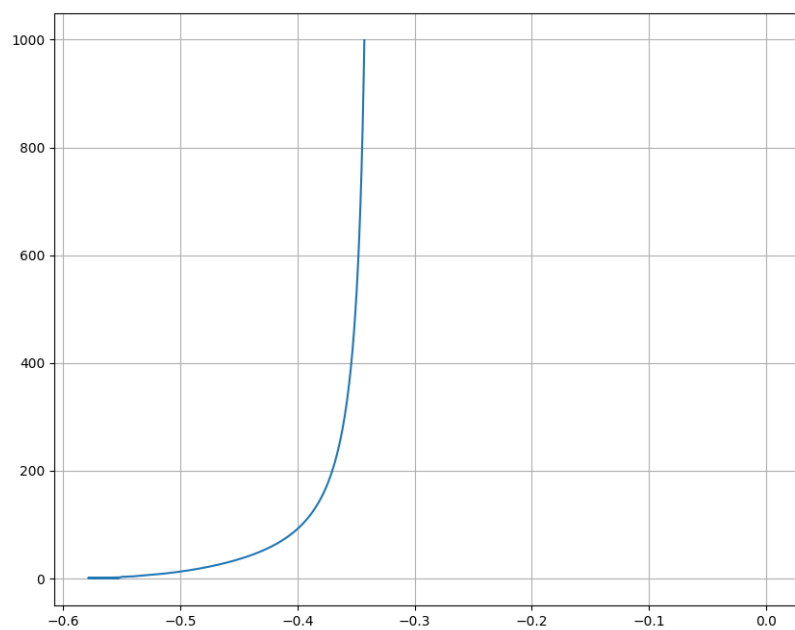


Рис. 10: momentum = 0.9

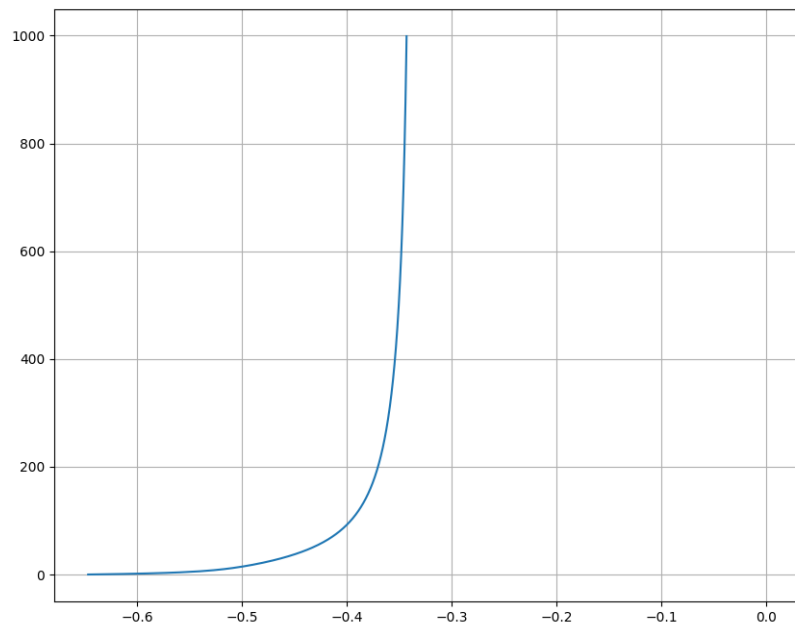


Рис. 11: momentum = 1

## 2.2 Обработка mushrooms

mushrooms				
momentum	accuracy	precision	recall	F1
0	1	1	1	1
0.1	1	1	1	1
0.2	1	1	1	1
0.3	1	1	1	1
0.4	1	1	1	1
0.5	1	1	1	1
0.6	1	1	1	1
0.7	1	1	1	1
0.8	1	1	1	1
0.9	1	1	1	1
1	1	1	1	1

Парадоксально, но мы можем точно угадывать класс объекта по вектору его характеристик (причем это происходит уже после первой же итерации).

Полученные графики функции потерь:

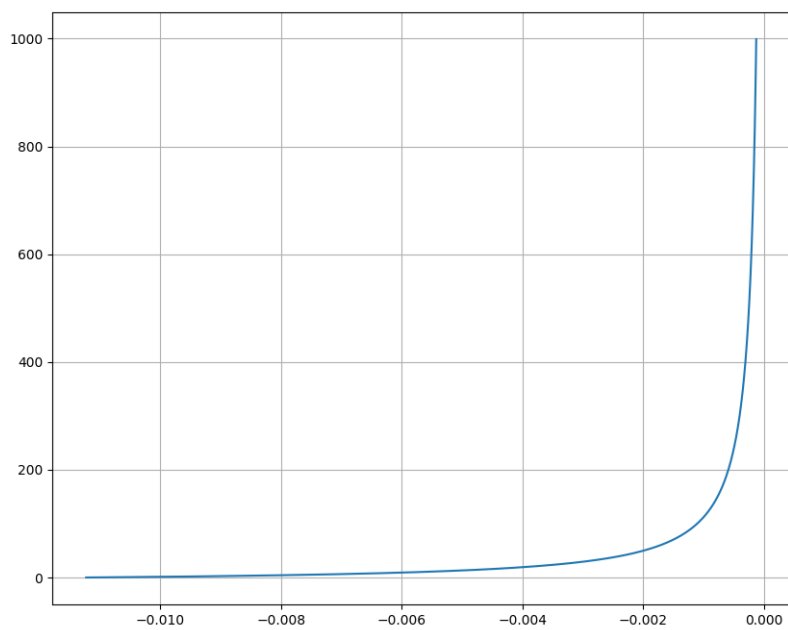


Рис. 12: momentum = 0



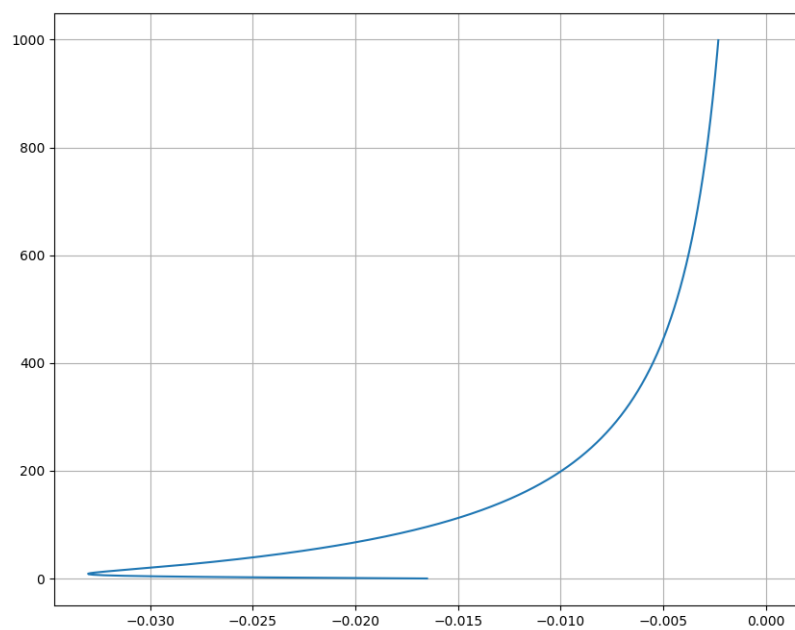


Рис. 13: momentum = 0.1

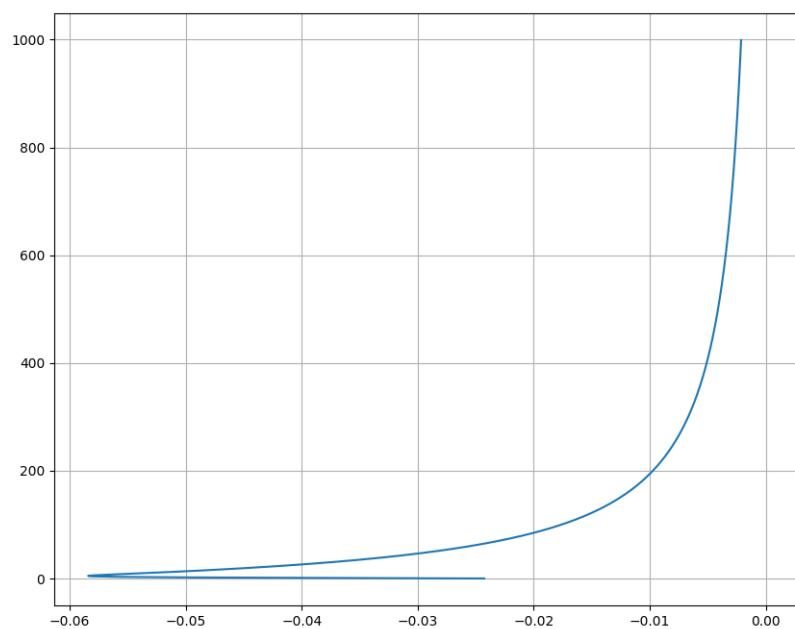


Рис. 14: momentum = 0.2

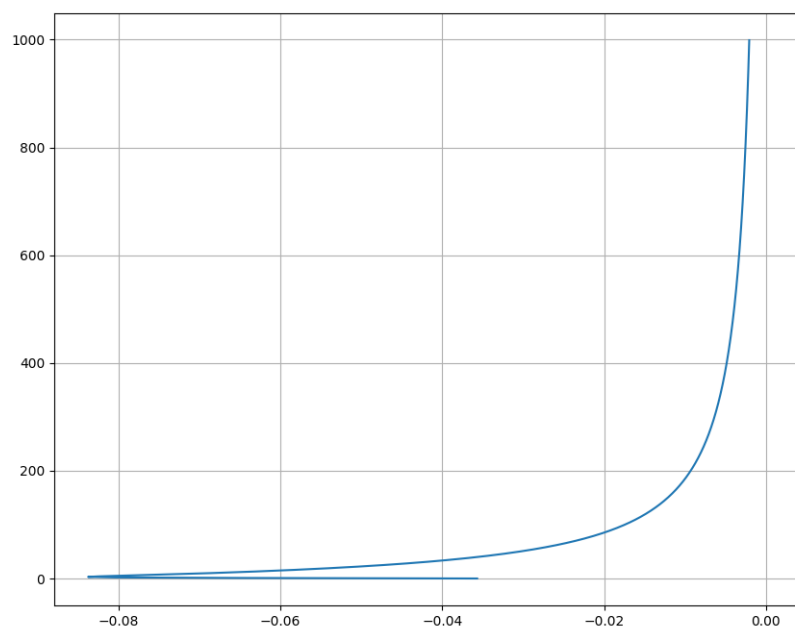


Рис. 15: momentum = 0.3

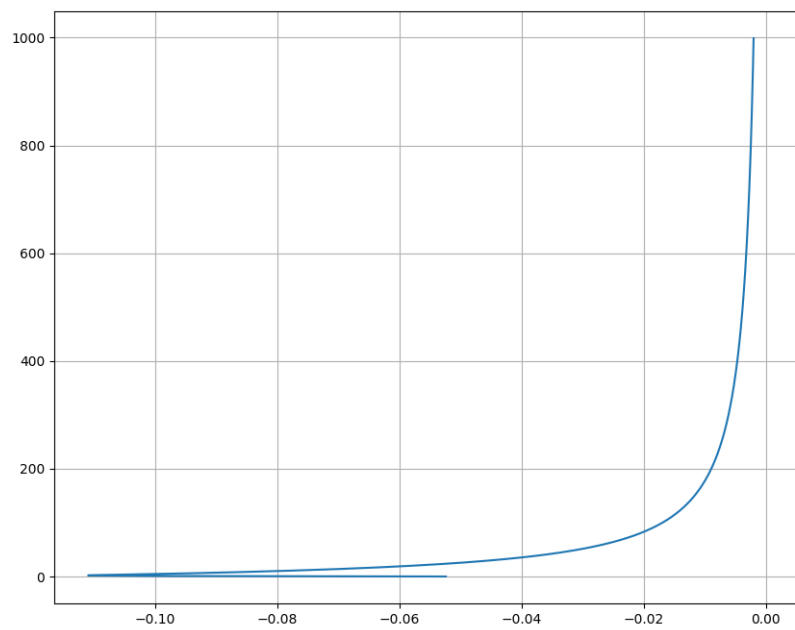


Рис. 16: momentum = 0.4

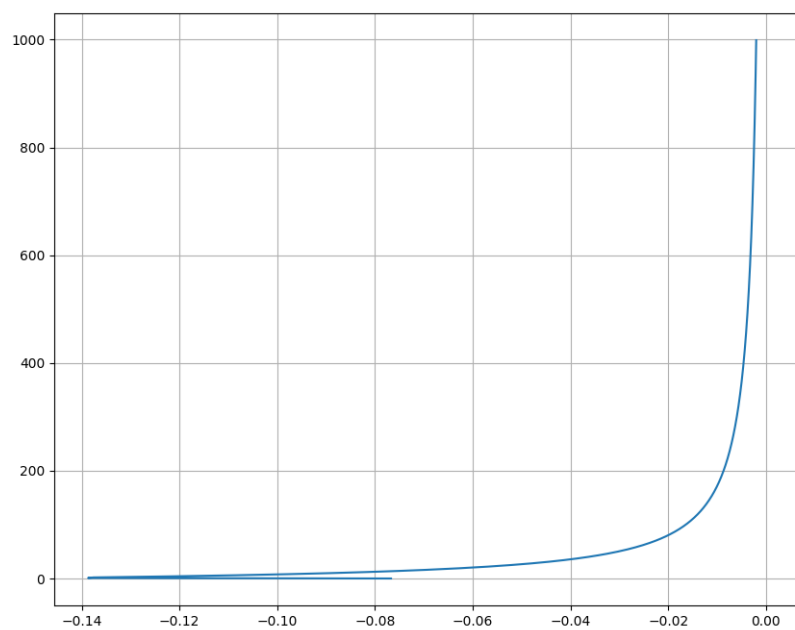


Рис. 17: momentum = 0.5

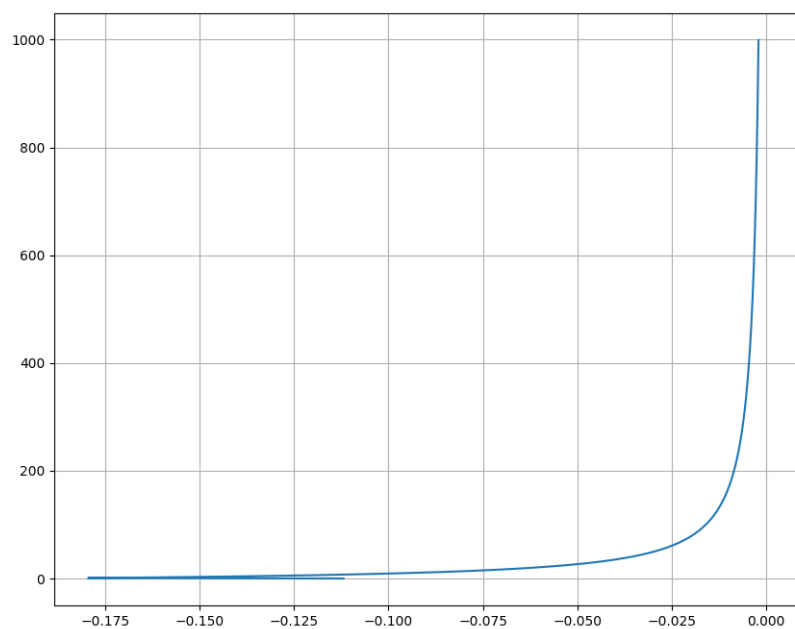


Рис. 18: momentum = 0.6

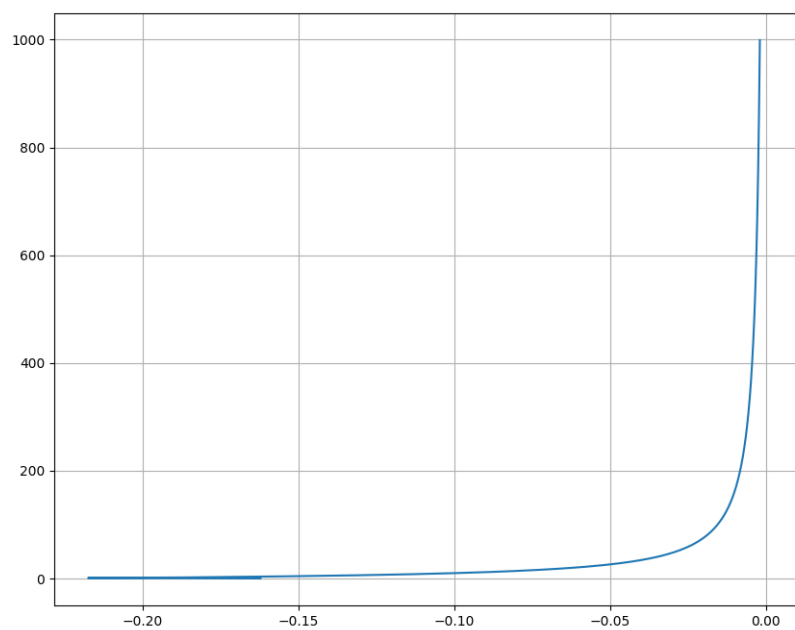


Рис. 19: momentum = 0.7

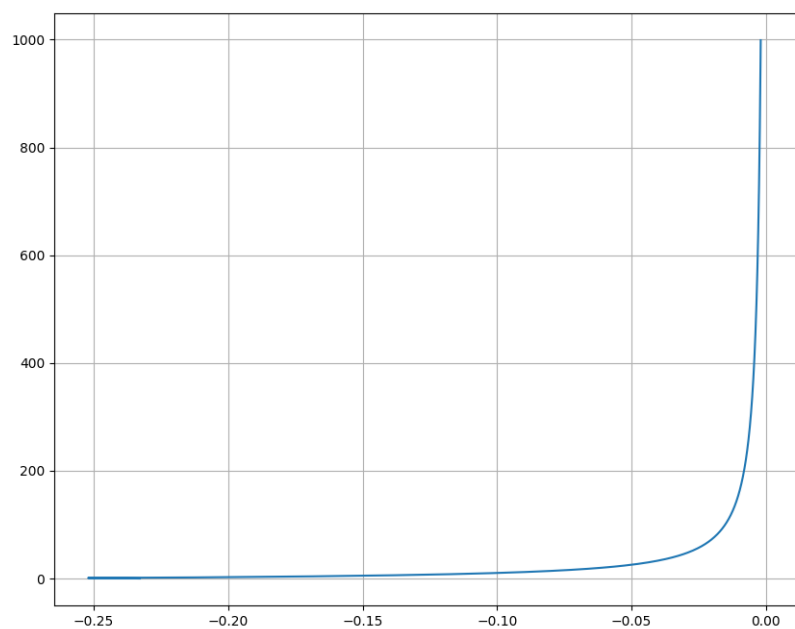


Рис. 20: momentum = 0.8

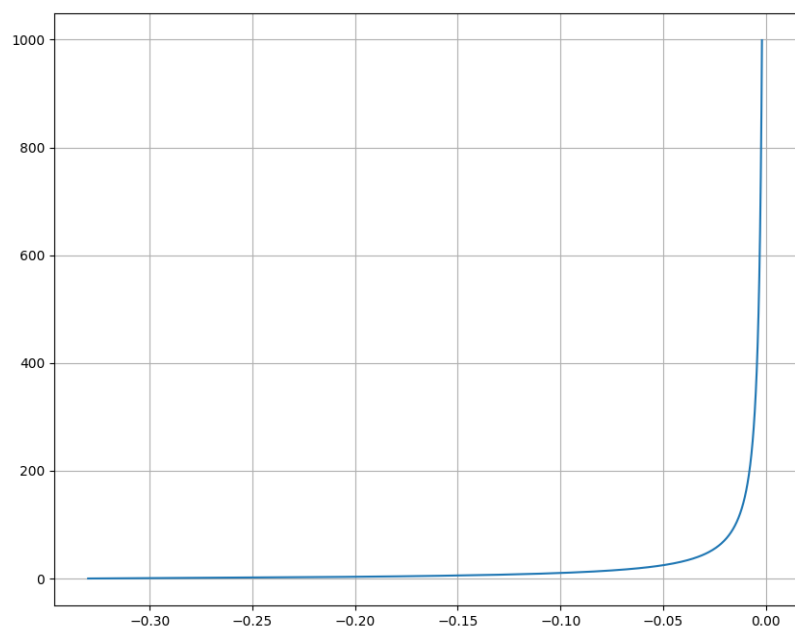


Рис. 21: momentum = 0.9

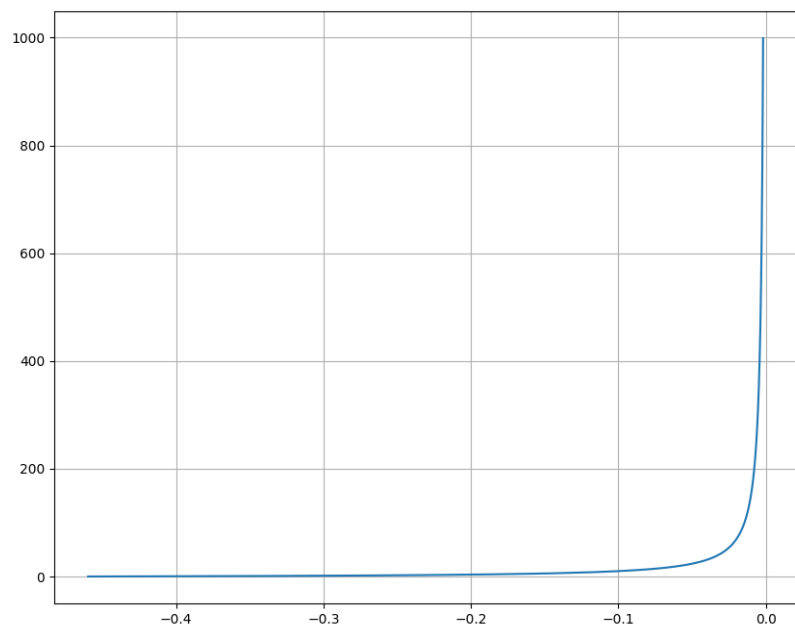


Рис. 22: momentum = 1

### 3 Выводы

По данным из файла a1a удалось получить  $F1\text{-score} = 0.675$ , что хоть и является неплохим результатом, но явно говорит о том, что могли быть упущены какие-то важные моменты, или же сам датасет не является полностью линейно классифицируемым.

В противовес a1a, датасет mushrooms дает на удивление хорошие результаты, позволяющие классифицировать данные абсолютно точно.