Отчет о практическом задании Градиентные методы обучения линейных моделей. Применение линейных моделей для определения токсичности комментария

Практикум 317 группы, ММП ВМК МГУ

Загатин Даниил Ильич

ноябрь 2024

Содержание

1 Введение

Данное практическое задание посвящено изучению и реализации градиентных методов обучения линейных моделей. Поставлена задача бинарной классификации текстовых комментариев на два класса: положительный (токсичные комментарии) и отрицательный (нетоксичные комментарии). Для этого применяется метод логистической регрессии. Обучение модели выполняется с использованием двух подходов: градиентного спуска и стохастического градиентного спуска.

В рамках выполнения задания ставятся следующие задачи:

- Разработка собственной реализации градиентных методов обучения на языке Python.
- Реализация модели бинарной логистической регрессии с использованием собственной реализации.
- Подбор и оптимизация гиперпараметров для методов обучения.
- Сравнение поведения методов градиентного спуска и стохастического градиентного спуска на различных наборах данных.
- Изучение влияния предобработки текстовых данных на качество работы модели, включая применение:
  - Лемматизации;
  - Представлений текста: BagOfWords и TfIdf.

Цель работы — не только познакомиться с алгоритмами градиентного спуска и методами обработки текстовых данных, но и получить представление о взаимосвязи между качеством, скоростью работы алгоритмов и характеристиками данных.

2 Теоретическая часть

2.1 Вывод формулы градиента функции потерь для задачи бинарной классификации

Введём основные понятия:
Линейная модель классификации для двух классов $Y = \{-1, +1\}$:

$$a(x) = \text{sign}\langle w, x\rangle, \quad x, w \in \mathbb{R}^n$$

$$\text{Отступ} \quad M = \langle w, x \rangle y$$

Модель условной вероятности:

$$P(y|x, w) = \sigma(M)$$

Логарифмическая функция потерь:

$$\mathcal{L}(M) = \log\left(1 + e^{-M}\right)$$

Следовательно, для обучения модели требуется минимизировать следующий функционал:

$$Q(X, w) = \frac{1}{l}\sum_{i=1}^{\ell} \mathcal{L}(M_i(w)) = \frac{1}{l}\sum_{i=1}^{\ell} \log\left(1 + e^{-\langle w, x_i \rangle y_i}\right) \to \min_w,$$

где $M_i(w)$ - отступ объекта x_i

Рассмотрим его градиент:

$$\nabla_w Q(X, w) = \nabla_w(\tfrac{1}{l}\textstyle\sum_{i=1}^{l} \log\left(1 + e^{-\langle w, x_i \rangle y_i}\right))$$

Посчитаем дифференциал для одного объекта из выборки:

$$d(\log\left(1 + e^{-\langle w, x_i \rangle y_i}\right)) = \tfrac{1}{1+e^{-\langle w, x_i \rangle y_i}} d(e^{-\langle w, x_i \rangle y_i}) = -y_i x_i \tfrac{e^{-\langle w, x_i \rangle y_i}}{1+e^{-\langle w, x_i \rangle y_i}} dw =$$

$$= -y_i x_i \tfrac{1}{1+e^{\langle w, x_i \rangle y_i}} dw = \{\text{функция сигмоида } \sigma(x) = \tfrac{1}{1+e^{-x}}\} = -y_i x_i \cdot \sigma(-\langle w, x_i \rangle y_i) dw$$

Таким образом, градиент для всей выборки можно представить в следующей форме:

$$\nabla_w Q(X, w) = -\frac{1}{l}\sum_{i=1}^{l} (y_i x_i \cdot \sigma(-\langle w, x_i \rangle y_i)) \tag{1}$$

2.2 Вывод формулы градиента функции потерь для задачи многоклассовой (мультиномиальной) бинарной классификации

Линейный классификатор при произвольном числе классов $|Y|$:

$$a(x) = \arg\max_{y \in Y} \langle w_y, x \rangle, \quad x, w_y \in \mathbb{R}^n.$$

Вероятность того, что объект x относится к классу y:

$$P(y|x,w) = \frac{e^{\langle w_y, x\rangle}}{\sum\limits_{z\in Y} e^{\langle w_z, x\rangle}} = \mathrm{SoftMax}\langle w_y, x\rangle$$

Функция потерь:

$$\mathcal{L}(w) = -\frac{1}{\ell}\sum_{i=1}^{\ell} \ln(P(y_i|x_i,w))$$

Вычислим частную производную по w_k:

$$\frac{\partial}{\partial w_k}\mathcal{L}(w) = -\frac{1}{\ell}\sum_{i=1}^{\ell}\frac{\partial}{\partial w_k}(\ln(P(y_i|x_i,w))) = -\frac{1}{\ell}\sum_{i=1}^{\ell}\frac{\partial}{\partial w_k}[\ln(e^{\langle w_{y_i}, x_i\rangle}) - \ln\sum_{z\in Y} e^{\langle w_z, x_i\rangle}]$$

Промежуточные преобразования:

$$\frac{\partial}{\partial w_k}\ln(e^{\langle w_{y_i}, x_i\rangle}) = \begin{cases} x_i, & \text{если } y_i = k, \\ 0, & \text{если } y_i \neq k. \end{cases}$$

$$\frac{\partial}{\partial w_k}\ln\sum_{z\in Y} e^{\langle w_z, x_i\rangle} = \frac{1}{\sum\limits_{z\in Y} e^{\langle w_z, x_i\rangle}}\cdot\frac{\partial}{\partial w_k}\sum_{z\in Y} e^{\langle w_z, x_i\rangle} =$$

$$= \{\frac{\partial}{\partial w_k}\sum_{z\in Y} e^{\langle w_z, x_i\rangle} = \begin{cases} x_i\cdot e^{\langle w_k, x_i\rangle}, & \text{если } z = k, \\ 0, & \text{если } z \neq k. \end{cases}\} = \frac{x_i e^{\langle w_k, x_i\rangle}}{\sum\limits_{z\in Y} e^{\langle w_z, x_i\rangle}} = x_i P(k|x_i,w)$$

$$-\frac{1}{\ell}\sum_{i=1}^{\ell}\frac{\partial}{\partial w_k}[\ln(e^{\langle w_{y_i}, x_i\rangle}) - \ln\sum_{z\in Y} e^{\langle w_z, x_i\rangle}] = \{\delta_{ij} - \text{символ Кронекера}\} =$$

$$= \frac{1}{\ell}\sum_{i=1}^{\ell}[x_i P(k|x_i,w) - x_i\cdot\delta_{ky_i}] = \frac{1}{\ell}\sum_{i=1}^{\ell}[P(k|x_i,w) - \delta_{ky_i}]x_i$$

Градиент всей функции потерь будет составлен из частных производных вычисленных приведённым способом.

2.3 Сведение мультиномиальной логистической регрессии к бинарной логистической регрессии при количестве классов равном 2

Функция потерь мультиномиальной логистической регрессии при $|Y| = 2$:

$$\mathcal{L}(w) = -\frac{1}{\ell}\sum_{i=1}^{\ell}\ln(P(y_i|x_i, w))$$

Распишем вероятность отношения объекта к одному из двух классов:

$$P(y_i|x_i, w) = \begin{cases} \frac{e^{\langle w_1, x_i\rangle}}{e^{\langle w_1, x_i\rangle}+e^{\langle w_2, x_i\rangle}} = \frac{1}{1+e^{\langle w_2 - w_1, x_i\rangle}} = \sigma(\langle w_1 - w_2, x_i\rangle \cdot (+1)), & \text{если } y_i = +1, \\ \frac{e^{\langle w_2, x_i\rangle}}{e^{\langle w_1, x_i\rangle}+e^{\langle w_2, x_i\rangle}} = \frac{1}{1+e^{-\langle w_2 - w_1, x_i\rangle}} = \sigma(\langle w_1 - w_2, x_i\rangle \cdot (-1)), & \text{если } y_i = -1. \end{cases}$$

Таким образом получаем модель условной вероятности, использующуюся для бинарной логистической регрессии:

$$P(y|x, w) = \sigma(\langle w, x_i\rangle \cdot y_i), \quad w = w_1 - w_2, y_i = \{-1, +1\}$$

Также покажем эквивалентность вычисления градиента функции потерь для мультиномиальной логистической регрессии и для бинарной логистической регрессии:

$$\frac{\partial}{\partial w_k}\mathcal{L}(w) = \frac{1}{\ell}\sum_{i=1}^{\ell}[P(k|x_i, w) - \delta_{ky_i}]x_i = \frac{1}{\ell}\sum_{i=1}^{\ell}[\sigma(\langle w, x_i\rangle \cdot y_i) - \delta_{ky_i}]x_i =$$

$$\{\text{Пусть k} = +1, \text{для -1 аналогично}\} = \frac{1}{\ell}\sum_{i=1}^{\ell}[\sigma(\langle w, x_i\rangle \cdot y_i) - \delta_{+1y_i}]x_i = \{$$

$$[\sigma(\langle w, x_i\rangle \cdot y_i) - \delta_{+1y_i}]x_i = \begin{cases} [\sigma(\langle w, x_i\rangle) - 1]x_i = -\sigma(-\langle w, x_i\rangle)x_i, & \text{если } y_i = +1, \\ \sigma(-\langle w, x_i\rangle)x_i = \sigma(-\langle w, x_i\rangle)x_i, & \text{если } y_i = -1. \end{cases}$$

$$\} = \{\text{объединим оба случая }\} = \frac{1}{\ell}\sum_{i=1}^{\ell}\sigma(-\langle w, x_i\rangle y_i) \cdot (-y_i x_i)$$

Следовательно, градиент функции потерь мультиномиальной логистической регрессии при количестве классов, равном 2, эквивалентен градиенту функции потерь бинарной логистической регрессии.

3 Эксперименты

3.1 Предварительная обработка данных

Данные в нашей задаче представляют собой датасет, содержащий комментарии из раздела обсуждений английской Википедии. В обучающей выборке содержится 52061 запись, в тестовой - 20676. Для классификации на токсичность ключевую роль играют слова, поэтому такие текстовые данные следует предварительно обработать:

- Привести все символы к нижнему регистру, чтобы слова, написанные с разным регистром, определялись как одинаковые;
- Заменить символы, не являющиеся буквами или цифрами, на пробелы (считаем, что они не влияют на классификацию);
- Заменить пустоты в данных на пробелы (для исключения ошибок).

3.2 Преобразование выборки

Признаковым пространством в нашей задаче является множество слов, содержащихся в комментариях. Это пространство имеет большую размерность, а данные являются разряженными. Поэтому выборку целесообразно преобразовать в разряженную матрицу с помощью конструктора `sklearn.feature_extraction.text.CountVectorizer`. Используем параметр `min_df` $= 0.001$, означающий, что слово включается в словарь, если встречается хотя бы в 0.1% документов. Полученное признаковое пространство имеет размерность 3736 слов. Комментарии могут иметь разную длину и, соответственно, абсолютные значения частот могут значительно варьироваться. Чтобы сгладить такие различия в данных, выполним их нормализацию.

3.3 Сравнение численного подсчёта градиента функции потерь с его вычислением по аналитической формуле

3.3.1 Исследование

Сравним время выполнения и разницу результатов численного и аналитического подсчёта градиента функции потерь бинарной логистической регрессии. Для этого создадим случайные выборки из 50 000 объектов с признаковым пространством разной размерности: 10, 50, 100, 250, 400, 500, 750, 1000.

Аналитический метод вычисления будет осуществляться по формуле (1). Численный подсчёт градиента будет выполняться по следующей формуле:

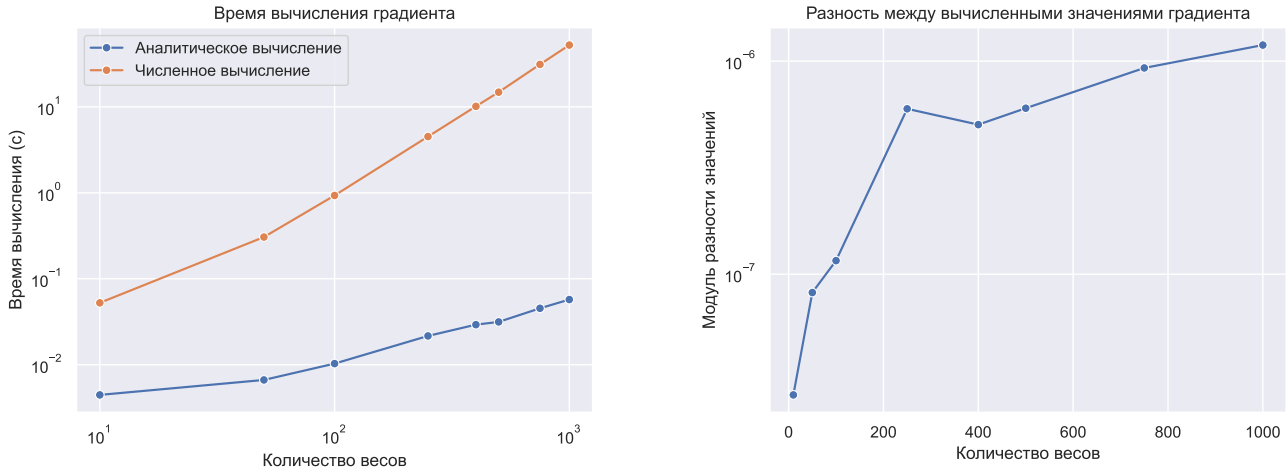
Значение i-ой координаты градиента

$$\text{result}_i := \frac{f(w + \epsilon \cdot e_i) - f(w)}{\epsilon},$$

где e_i — это вектор, в котором все компоненты равны нулю, за исключением i-й компоненты, которая равна единице:

$$e_i = (0, 0, \ldots, 0, 1, 0, \ldots, 0),$$

где единица стоит на i-й позиции.



(a) Сравнение времени выполнения

(b) Модуль разности результатов

Рис. 1: Сравнение численного и аналитического подсчёта градиента

По графику 1a заметно, что зависимость времени выполнения вычислений от количества весов имеет линейный характер в логарифмической шкале. Это означает, что в стандартных координатах зависимость имеет экспоненциальный характер. У аналитического метода подсчёта время вычисления значительно меньше, и с увеличением размерности признакового пространства оно изменяется гораздо медленнее по сравнению с численным методом.

Такой характер зависимости обусловлен тем, что численный метод требует вычисления разностных производных для каждого веса, что приводит к увеличению времени выполнения с ростом размерности признакового пространства. В отличие от этого, аналитический метод опирается на выполнение оптимизированных математических операций, таких как матричное умножение, и его сложность увеличивается гораздо медленнее с ростом размерности.

На графике 1b представлена зависимость модуля разности результатов двух алгоритмов от размерности признакового пространства. С увеличением количества признаков разность результатов растёт, но остаётся сравнительно небольшой (максимальное значение модуля разности находится в порядке 10^{-6}).

3.3.2 Вывод

Таким образом, можно сделать вывод о высокой эффективности вычисления градиента функции потерь с использованием аналитической формулы, а также о схожести результатов, получаемых двумя методами.

3.4 Исследование поведения градиентного спуска для задачи логистической регрессии

Для последующих экспериментов зафиксируем параметры модели по умолчанию: `step_alpha`=1, `step_beta`=0, `tolerance`=1e-5, `max_iter`=1000, `l2_coeff`=0, начальное приближение - нулевой вектор. В каждом из описанных далее экспериментов будет изменяться только один исследуемый параметр.

3.4.1 Подбор параметра шаг α

Проведём исследование поведения градиентного спуска при различных значениях шага α. Рассматриваемые значения: [0, 0.4, 0.8, 2, 5, 10, 20, 50, 70, 90, 100]. Значение $\alpha = 0$ будет использовано только для того, чтобы оценить, насколько далеко продвинется спуск за два шага.

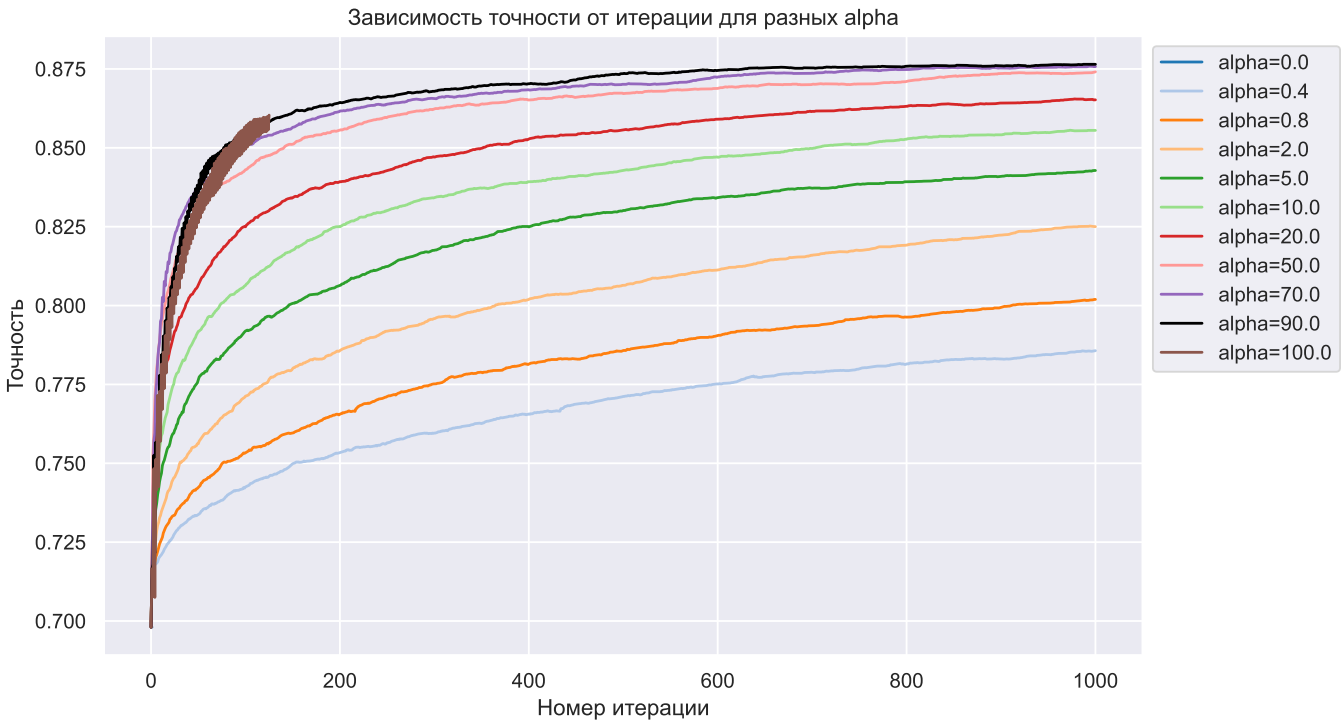


Рис. 2: Точность от итерации для различных α

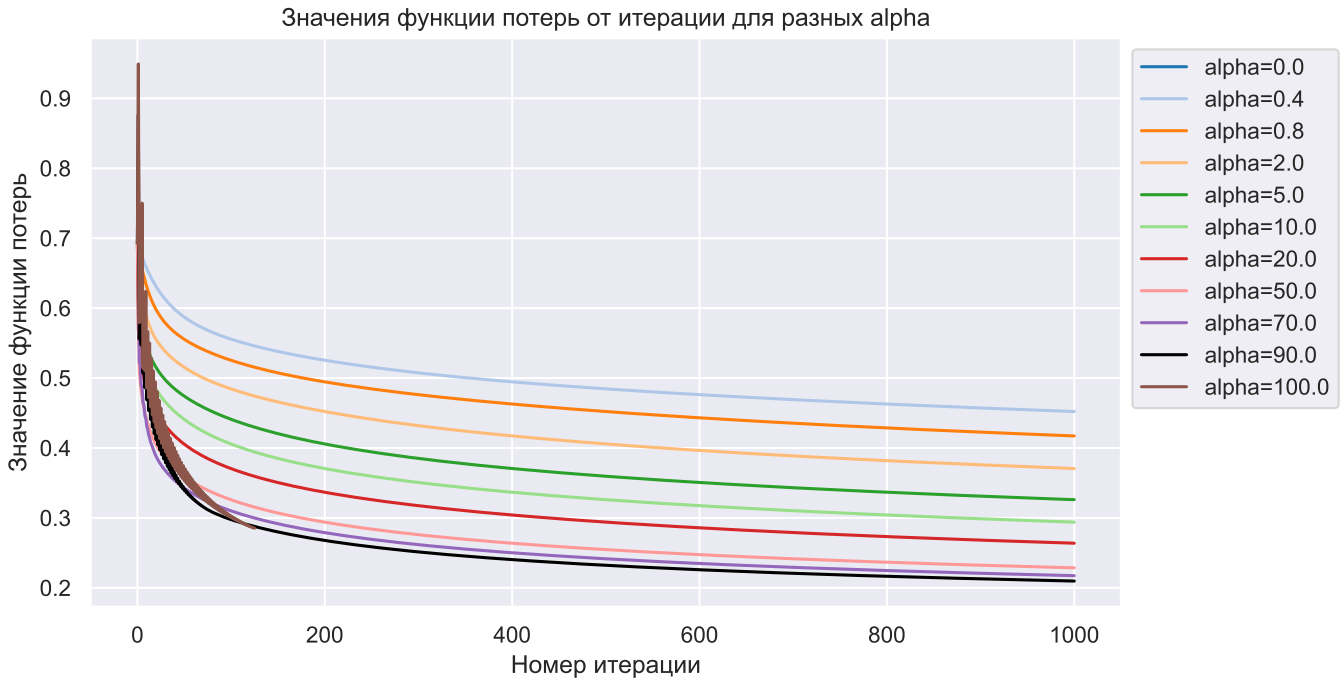


Рис. 3: Значение функции потерь от итерации для различных α

По графику зависимости точности от количества итераций (2) видно, что наилучшая точность достигается при значении шага $\alpha = 90$. При дальнейшем увеличении α градиентный спуск начинает сходиться нестабильно и в конечном итоге останавливается, из-за выполнения условия $|f(x_{k+1}) - f(x_k)| < \text{tolerance}$.

Аналогично, на графике зависимости функции потерь от количества итераций (3) также наблюдается наилучшая сходимость при $\alpha = 90$, в то время как при $\alpha = 100$ сходимость становится нестабильной.

При $\alpha = 100$ градиентный спуск останавливается не в оптимальной точке, что видно из низкой точности и нелучшем значения функции потерь на последней итерации. Это свидетельствует о том, что шаг слишком велик, и мы не можем достичь оптимума.

Таким образом, $\alpha = 90$ является оптимальным значением для данной модели с остальными параметрами по умолчанию. Стоит отметить, что это значение относительно велико, что связано с нормализацией выборки.

3.4.2 Подбор параметра шаг β

Проведём исследование поведения градиентного спуска при различных значениях шага β. Рассматриваемые значения: [0, 0.1, 0.2, 0.3,. . . , 1.4, 3, 5, 10]

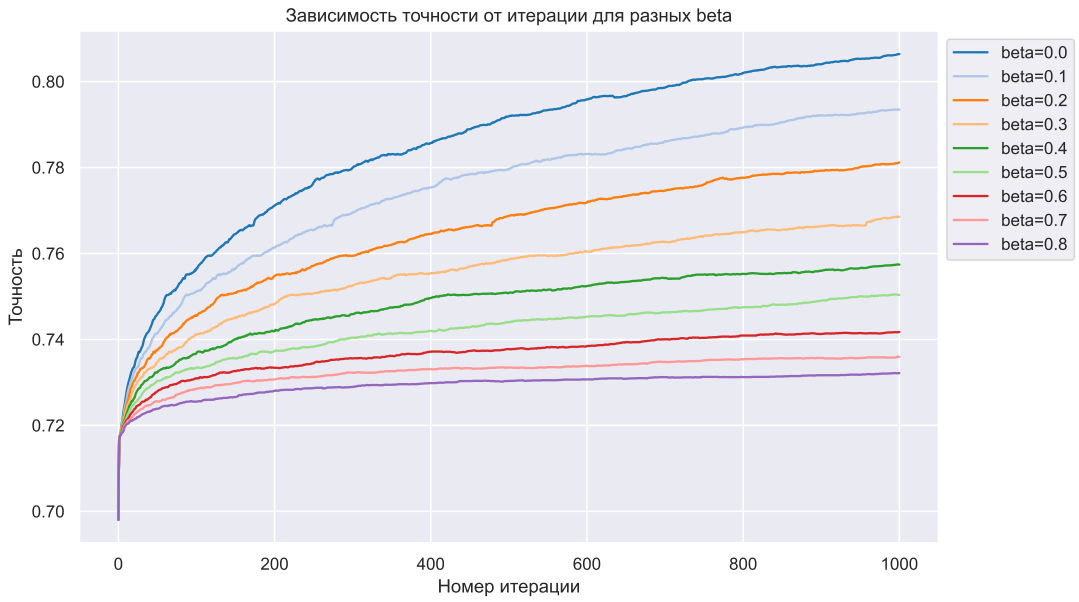


Рис. 4: Точность от итерации для различных β 1

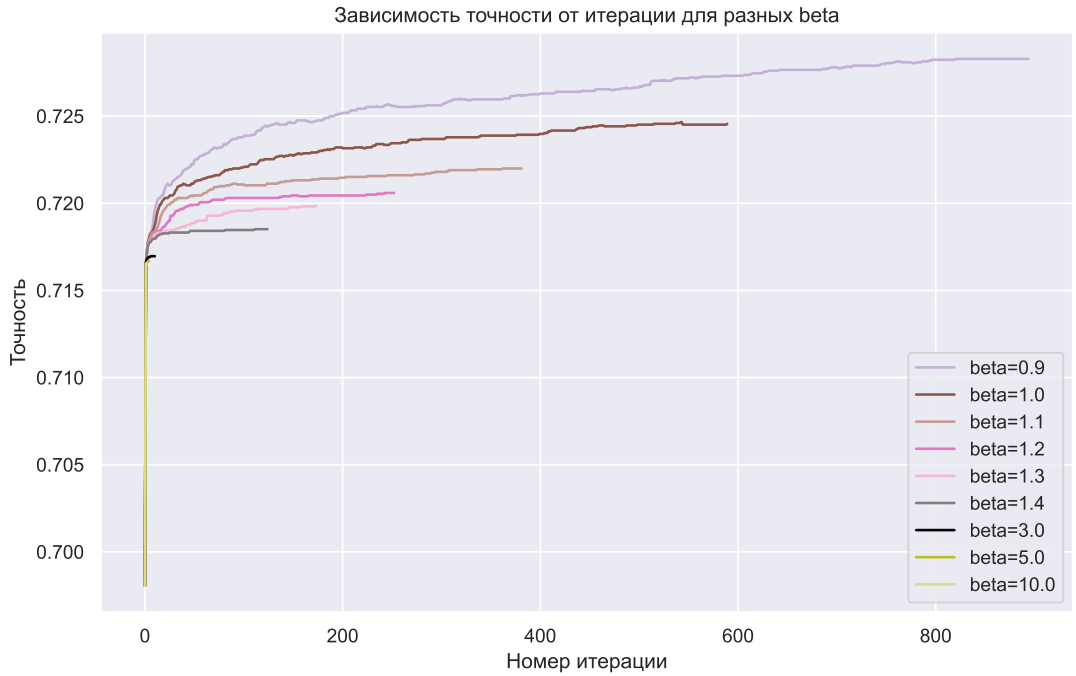


Рис. 5: Точность от итерации для различных β 2

На графиках зависимости точность от номера итерации 4 и 5 можно заметить, что наилучшая точность достигается при $\beta = 0$. При увеличении данного параметра точность уменьшается. При $\beta >= 0.9$ градиентный спуск сходится по условию tolerance за меньшее число итераций, но с худшей точностью.

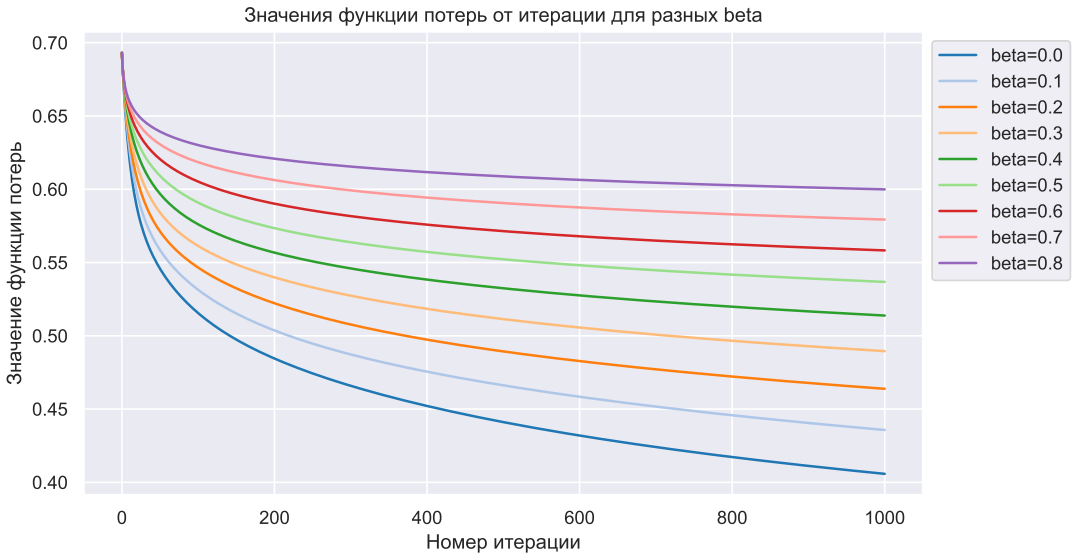


Рис. 6: Значение функции потерь от итерации для различных β 1

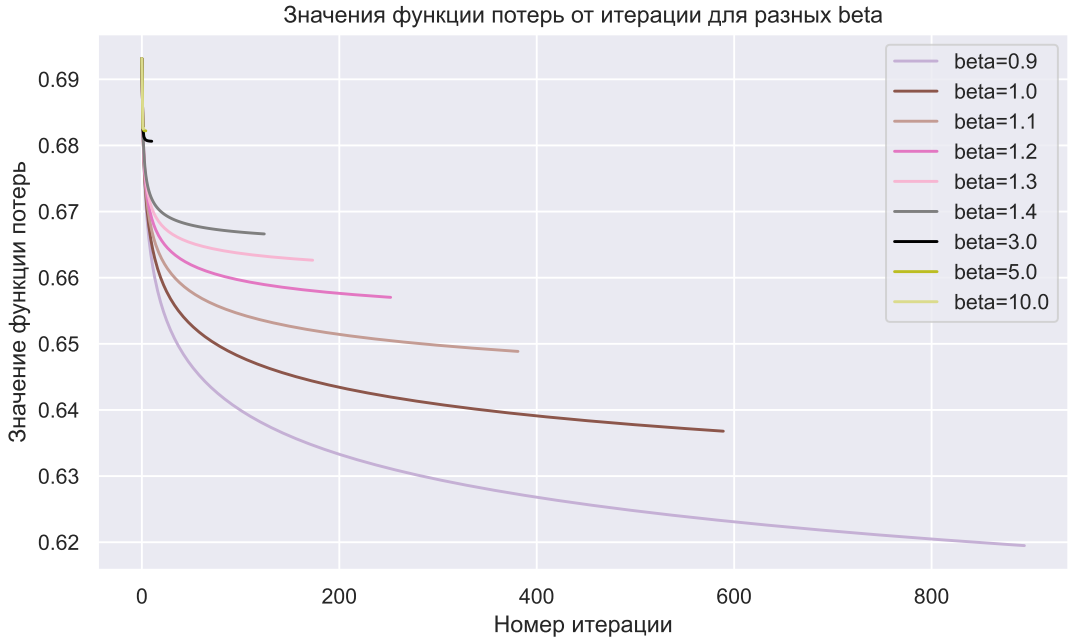


Рис. 7: Значение функции потерь от итерации для различных β 2

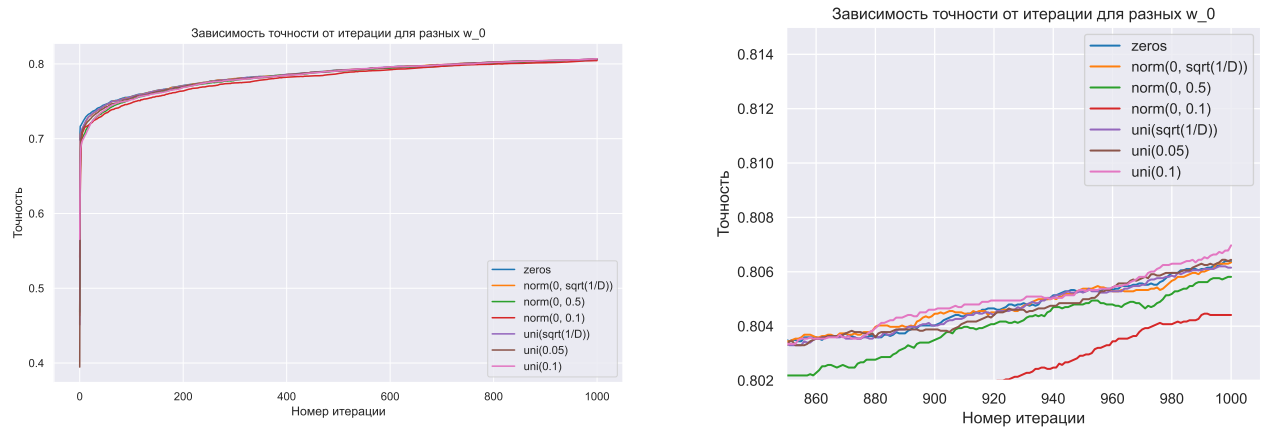
Графики зависимости значения функции потерь от номера итерации (6 и 7) подтверждают предыдущие выводы. При $\beta = 0$ наблюдается наилучшая сходимость по скорости к минимальному значению функции потерь.

При увеличении значения β шаг градиентного спуска быстро уменьшается, что приводит к сходимости за меньшее количество итераций, однако результат оказывается далёким от точки оптимума.

3.4.3 Подбор начального приближения

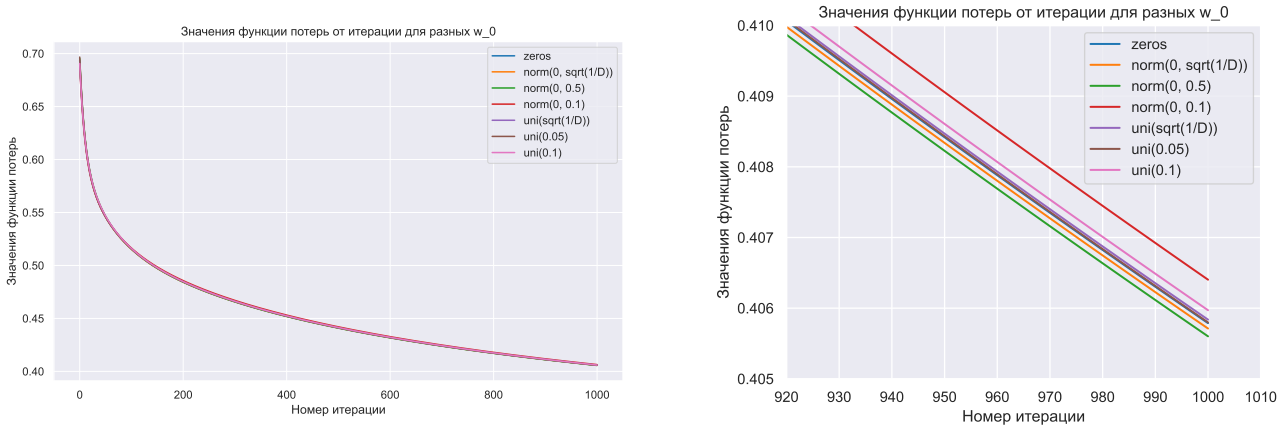
Для начального приближения в различных моделях обычно используют нулевой вектор или векторы, распределённые согласно нормальному или равномерному законам распределения.

Рассмотрим следующие варианты начального приближения:
[zeros, $\mathcal{N}(0, \sqrt{1/D})$, $\mathcal{N}(0, 0.5)$, $\mathcal{N}(0, 0.1)$, $\mathrm{Uni}(\sqrt{1/D})$, $\mathrm{Uni}(0.05)$, $\mathrm{Uni}(0.1)$],
D - количество признаков.



(a) Основной график (b) Приближенный график

Рис. 8: Зависимость точности от итерации при различных начальных приближениях



(a) Основной график (b) Приближенный график

Рис. 9: Зависимость значения функции потерь от итерации при различных начальных приближениях

Данные графики 8 и 9 показывают, что выбор начального приближения практически не влияет на качество и скорость сходимости градиентного спуска. Для

того чтобы разница была более заметна, масштаб графиков был увеличен. Разница в точности между моделями при различных начальных приближениях не превышает 0.003, а разница в значениях функции потерь — 0.002. Относительно выделяется начальное приближение с равномерным распределением (0, 0.1), но и в этом случае разница с другими методами составляет лишь тысячные доли.

Таким образом, можно заключить, что начальное приближение не оказывает существенного влияния на качество и скорость обучения модели в рамках нашего эксперимента. В дальнейшем для экспериментов будет использован нулевой вектор в качестве начальных весов.

3.5 Исследование поведения стохастического градиентного спуска для задачи логистической регрессии

Для последующих экспериментов зафиксируем параметры модели по умолчанию: `batch_size`=250, `random_seed` = 153, `step_alpha`=1, `step_beta`=0, `tolerance`=1e-5, `max_iter`=1000, `l2_coeff`=0, начальное приближение - нулевой вектор. В каждом из описанных далее экспериментов будет изменяться только один исследуемый параметр.

3.5.1 Подбор параметра шаг α

Проведём исследование поведения градиентного спуска при различных значениях шага α. Рассматриваемые значения: [0, 0.2, 0.4, ... 1.4, 2, 5, 10, 20, 50, 70, 90]. Значение $\alpha = 0$ будет использовано только для того, чтобы оценить, насколько далеко продвинется спуск за два шага.

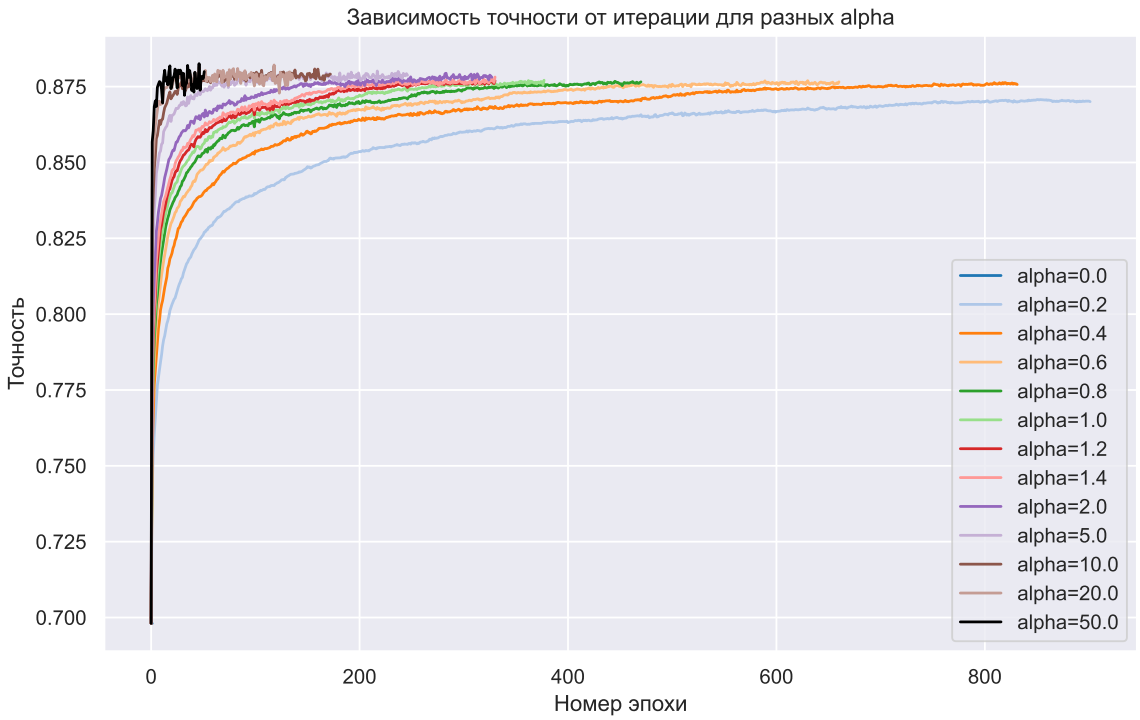


Рис. 10: Точность от эпохи для различных α 1

Рис. 11: Точность от эпохи для различных α 2

По графикам зависимости точности от количества итераций 10 и 11 видно, что наилучшая точность достигается при значениях шага $\alpha = 5$, 10 и 20. При уменьшении α стохастический градиентный спуск сходится стабильнее, но за большее

количество эпох, при этом точность не увеличивается. С увеличением α сходимость становится более нестабильной, и скорость сходимости возрастает, но не бесконечно. При дальнейшем увеличении $\alpha \geq 90$ количество эпох увеличивается, что связано с тем, что шаг становится слишком большим. В этом случае стохастический градиентный спуск то приближается к оптимуму, то удаляется от него из-за колебаний по различным наборам весов, не достигая его.

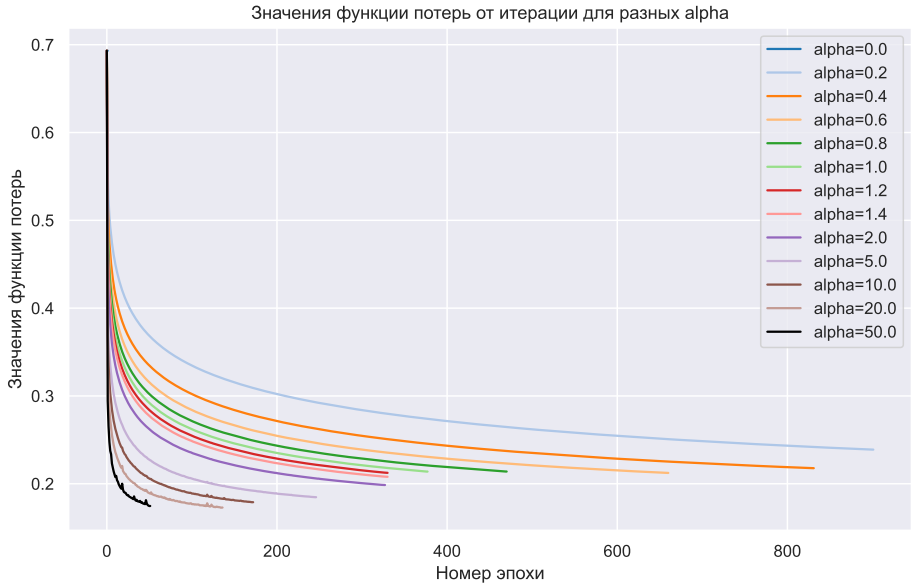


Рис. 12: Значение функции потерь от эпохи для различных α 1

Рис. 13: Значение функции потерь от эпохи для различных α 2

Рассмотрим также графики зависимости значения функции потерь от номера эпохи 12 и 13. Они подтверждают предыдущие суждения о поведении стохастического градиентного спуска.

Таким образом, исходя из совокупности следующих характеристик: высокой точности, быстрой скорости и стабильного характера сходимости, оптимальным значением α можно считать 10.

3.5.2 Подбор параметра шаг β

Проведём исследование поведения градиентного спуска при различных значениях шага β. Будем рассматривать значения от 0 до 1.4 с шагом 0.1. Более большие значения не будем рассматривать исходя из результатов, полученных в аналогичном эксперименте с градиентным спуском.

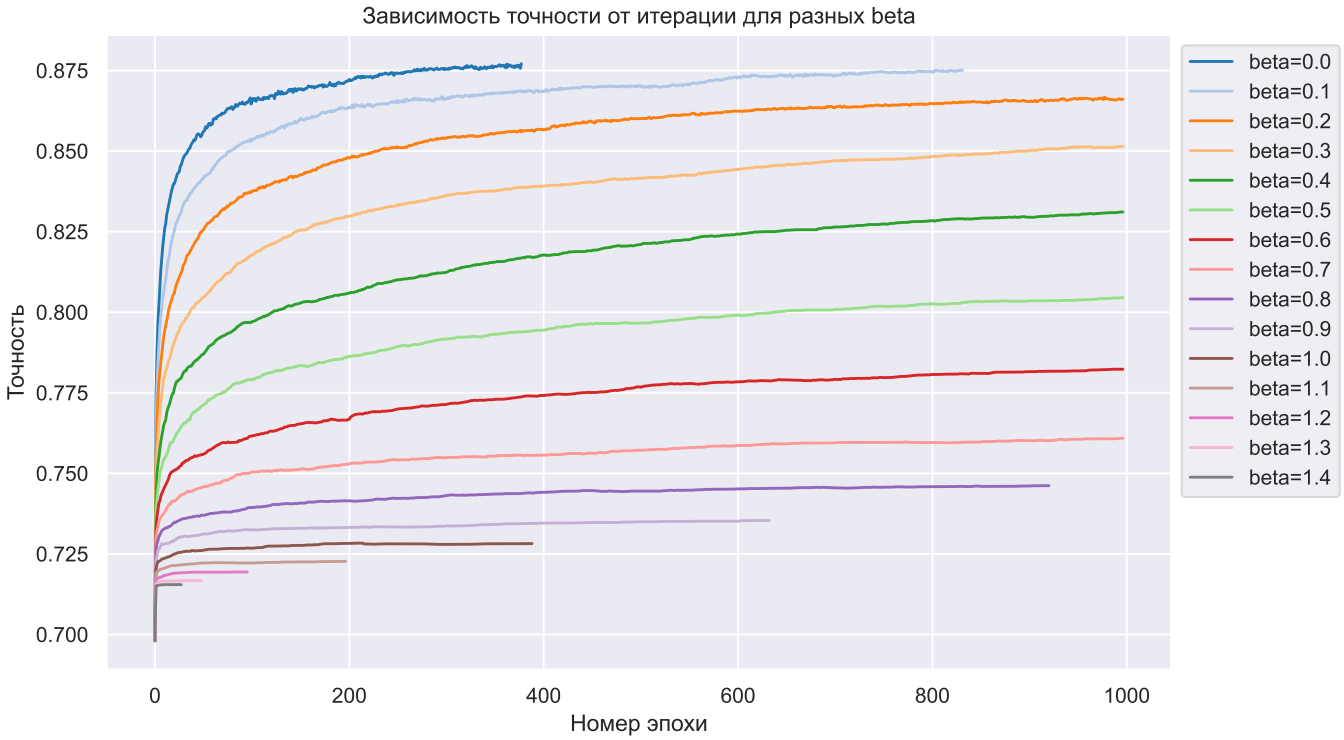


Рис. 14: Точность от эпохи для различных β

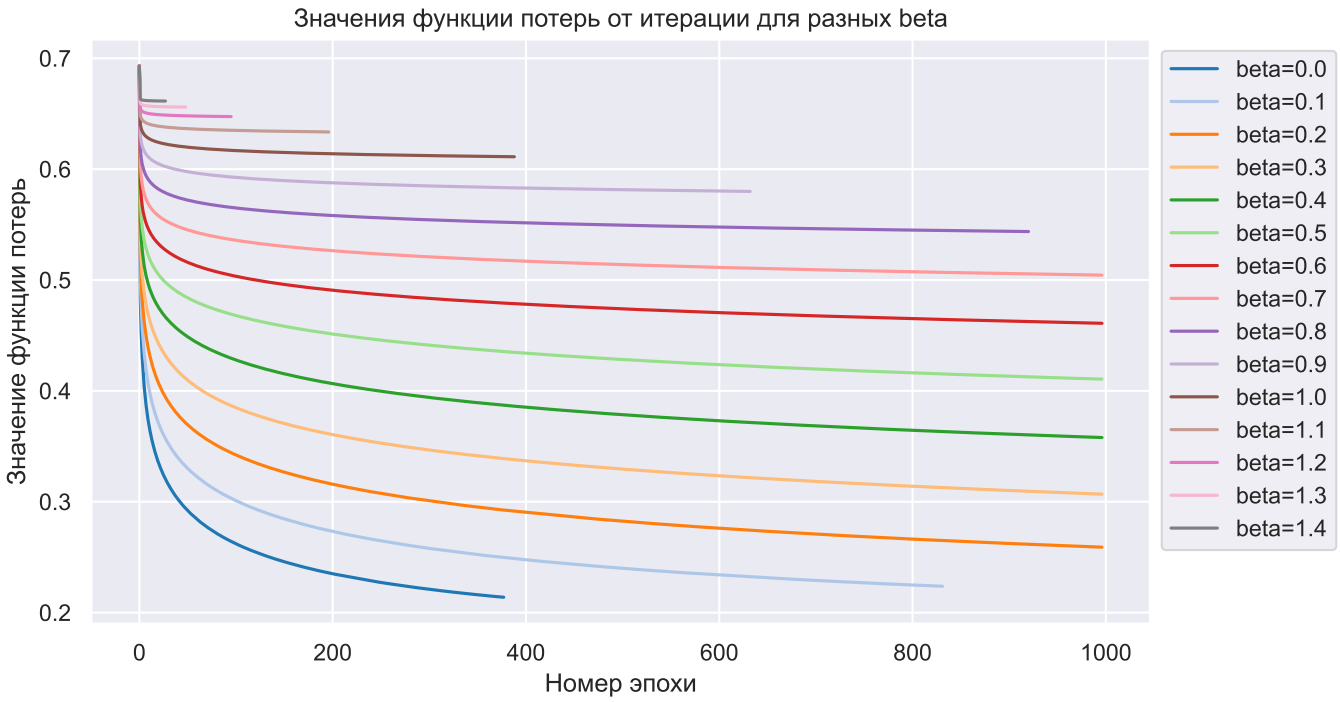
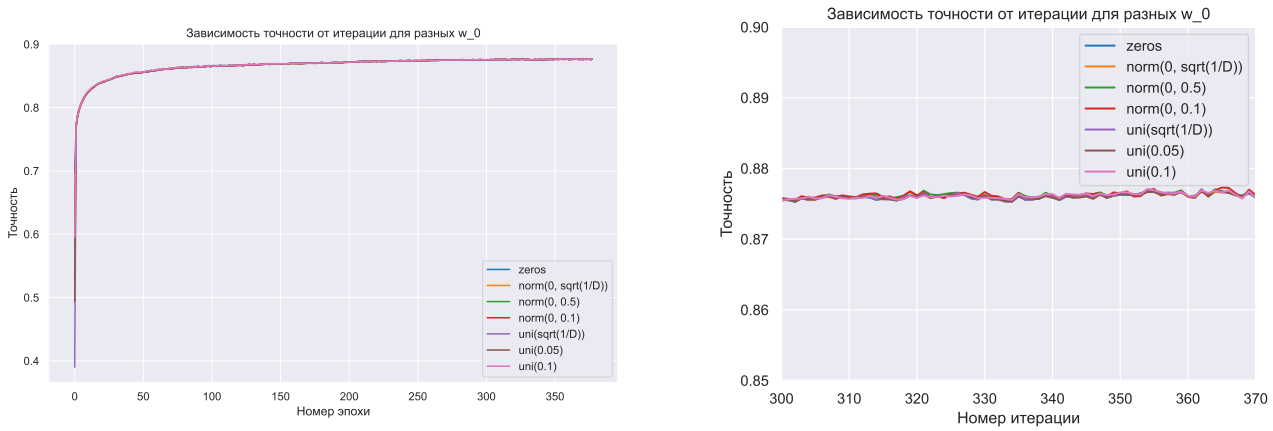


Рис. 15: Значение функции потерь от эпохи для различных β

На графиках зависимости точности от номера эпохи 14 и зависимости значения функции потерь от номера эпохи 15 заметно выделяется значение $\beta = 0$. При этом достигается максимальная точность за относительно небольшое количество эпох, а также минимальное значение функции потерь. Хорошая точность и стабильная сходимость также наблюдаются при $\beta = 0.1$, но для этого требуется значительно большее количество эпох. При дальнейшем увеличении β сходимость ухудшается. Особенно при больших значениях параметра стохастический градиентный спуск быстро останавливается по условию tolerance, что происходит по аналогичной причине, как и в эксперименте с градиентным спуском 3.4.2.

3.5.3 Подбор начального приближения

Аналогично эксперименту с градиентным спуском 3.4.3 рассмотрим следующие варианты начального приближения:
[zeros, $\mathcal{N}(0, \sqrt{1/D})$, $\mathcal{N}(0, 0.5)$, $\mathcal{N}(0, 0.1)$, $\mathrm{Uni}(\sqrt{1/D})$, $\mathrm{Uni}(0.05)$, $\mathrm{Uni}(0.1)$],
D - количество признаков.



(a) Основной график (b) Приближенный график

Рис. 16: Зависимость точности от эпохи при различных начальных приближениях

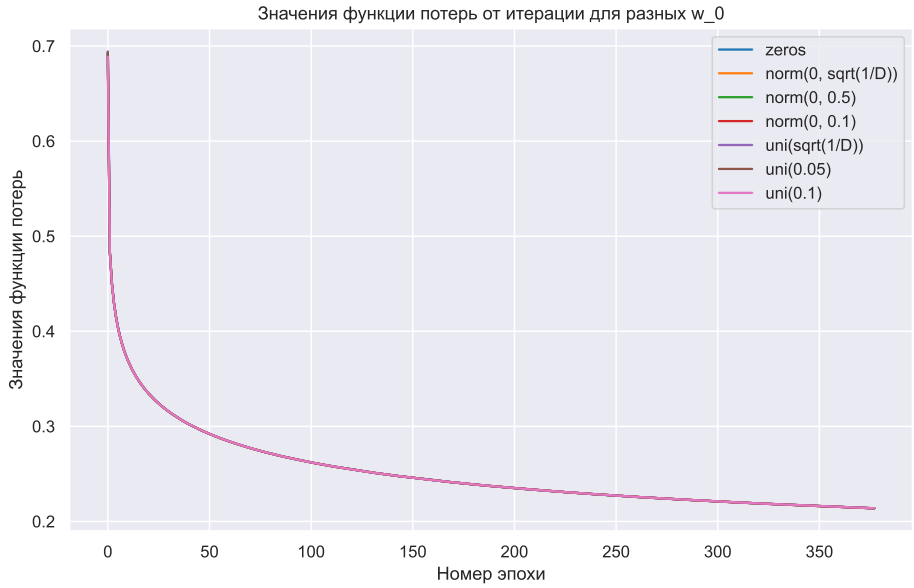


Рис. 17: Зависимость значения функции потерь от эпохи при различных начальных приближениях

Данные графики 16 и 17 показывают, что выбор начального приближения не влияет на качество и скорость сходимости стохастического градиентного спуска. В отличие от эксперимента с градиентным спуском 3.4.3, даже на приближенных графиках не наблюдается существенной разницы в поведении алгоритма при различных начальных приближениях.

Таким образом, можно сделать вывод, что начальное приближение не оказывает существенного влияния на качество и скорость обучения модели в рамках нашего эксперимента. В дальнейшем для экспериментов со стохастическим градиентным спуском будет использован нулевой вектор в качестве начальных весов,

так же как и в случае с градиентным спуском. Возможно, отсутствие различий в поведении методов при различных начальных приближениях связано с нормализацией выборки.

3.5.4 Подбор размера подвыборки

Рассмотрим следующие значения размеров подвыборки `batch_size`: [5, 25, 50, 100, 250, 500, 1000, 1500, 3000]

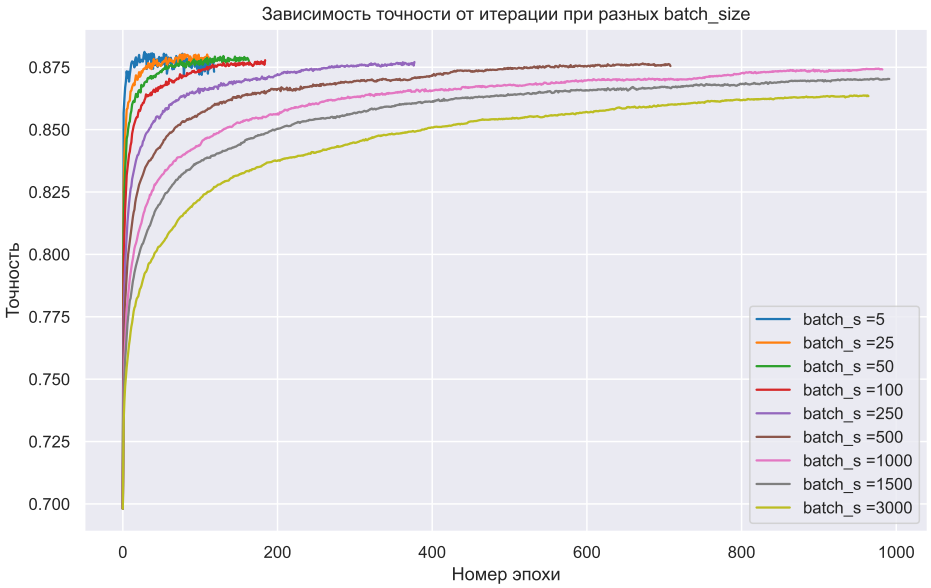


Рис. 18: Зависимость точности от эпохи при размерах подвыборки

График 18 показывает, что наилучшая точность и скорость сходимости достигается при следующих значениях размера подвыборки: 5, 25, 50, 100.

Рассмотрим график их точности от номера эпохи отдельно 19. На нём видно, что при `batch_size` $= 100$ сходимость имеет наиболее стабильный характер.

График зависимости значения функции потерь от эпохи 20 подтверждает предыдущий вывод. И, хотя при размере, равном 100, не достигается минимальное значение функции потерь, в дальнейшем мы будем использовать именно это значение в связи с необходимостью ускорения экспериментов. При уменьшении размера батча упрощаются вычисления на одной итерации, но при этом растёт число этих итераций за одну эпоху, что может замедлять метод в целом. Значение 100 будет оптимальным по точности, стабильности сходимости и времени выполнения стохастического градиентного спуска.

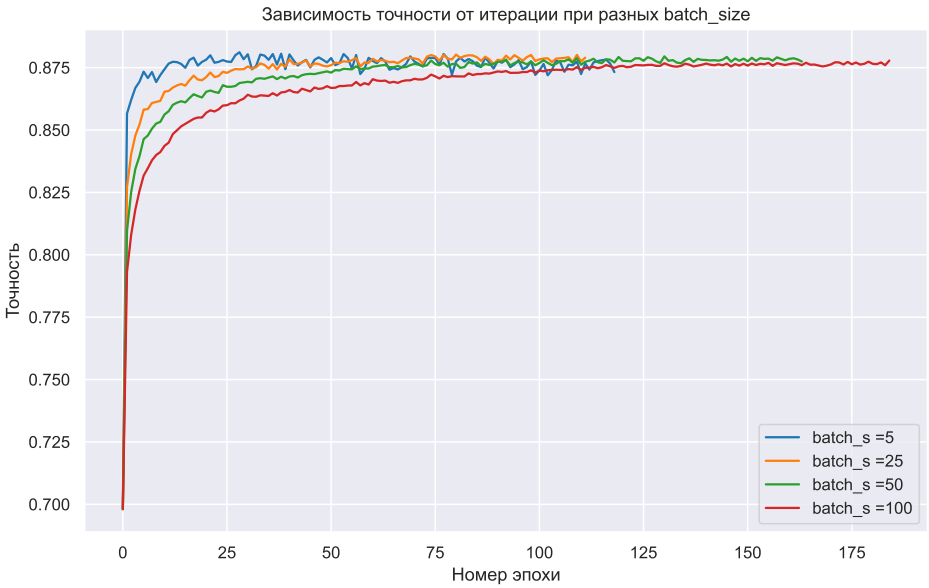


Рис. 19: Зависимость точности от эпохи при размерах 5,25,50,100

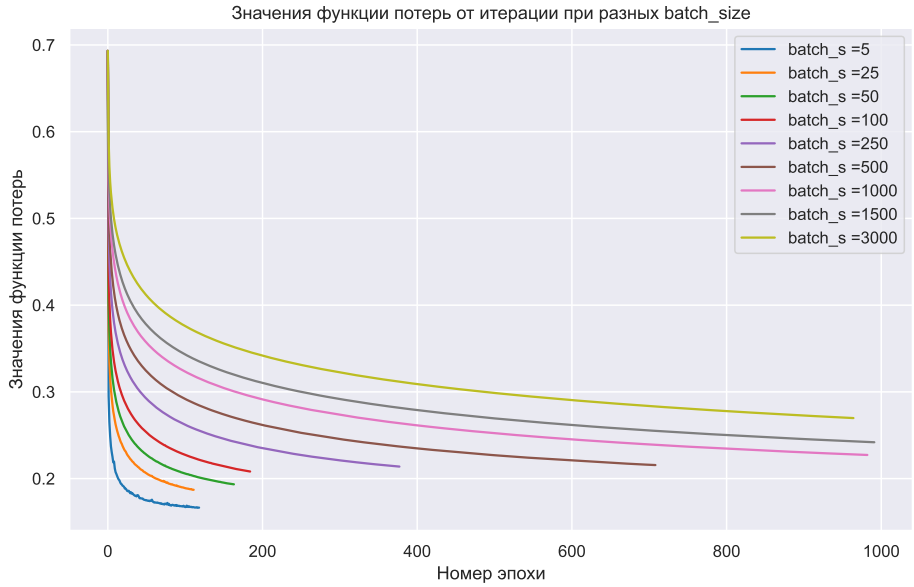


Рис. 20: Зависимость значения функции потерь от эпохи при различных размерах подвыборки

3.6 Сравнение градиентного и стохастического градиентного спуска

Для сравнения двух методов обучим модели с полученными наилучшими параметрами. Наилучшие полученные характеристики для метода градиентного спус-

ка:

- step_alpha = 90,
- step_beta = 0,
- $\mathbf{w}_0 = \mathbf{zeros}$.

Наилучшие полученные характеристики для метода стохастического градиентного спуска:

- step_alpha = 10,
- step_beta = 0,
- $\mathbf{w}_0 = \mathbf{zeros}$,
- batch_size = 100.

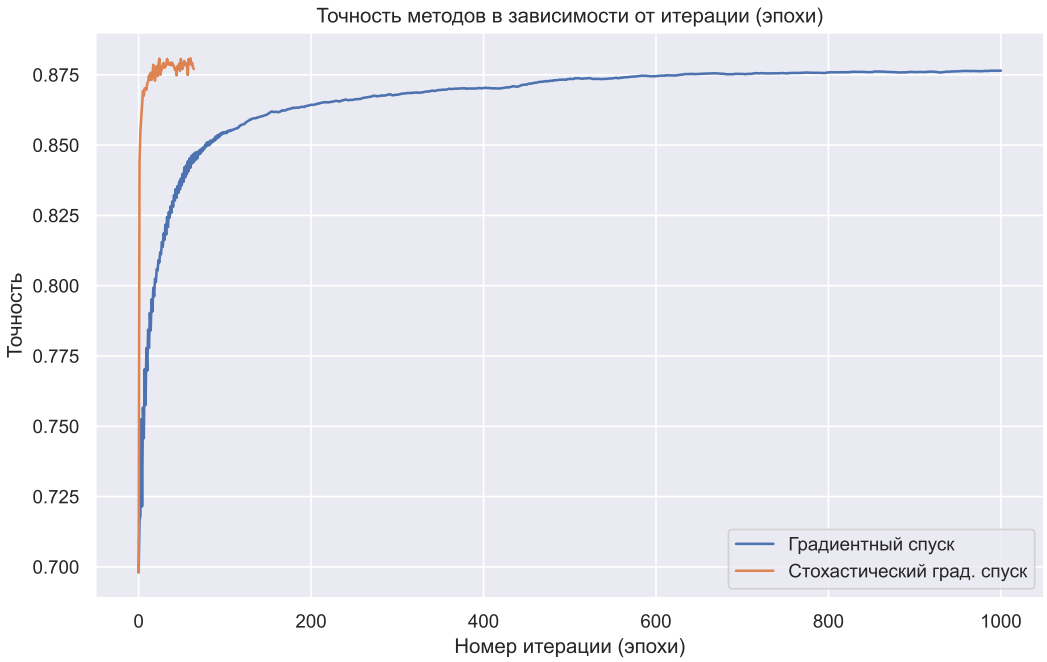


Рис. 21: Зависимость точности от номера итерации(эпохи)

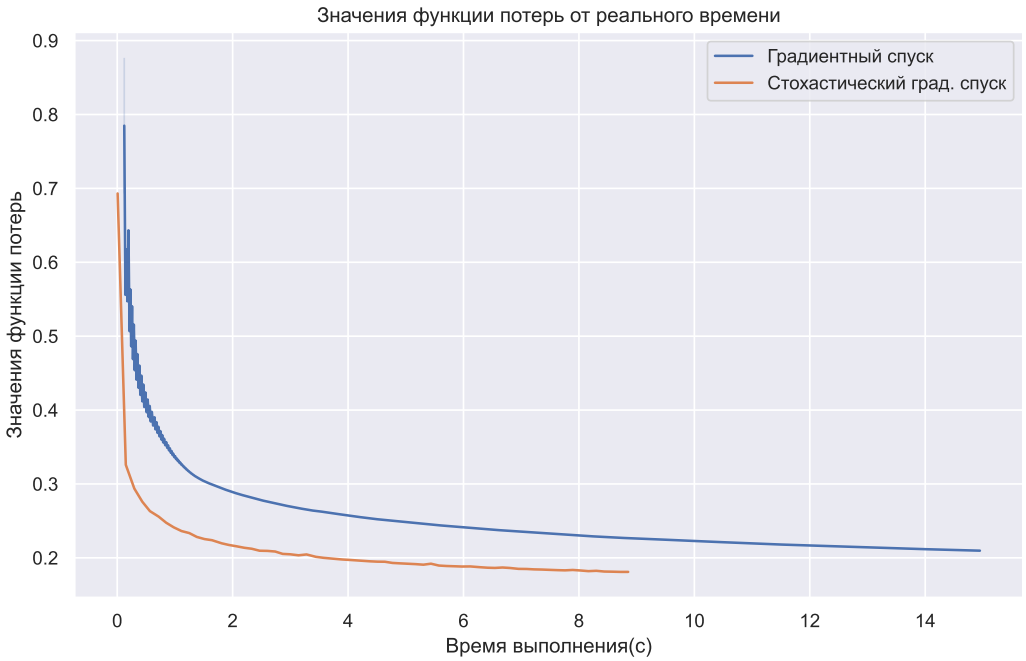


Рис. 22: Зависимость значения функции потерь от реального времени

График, представленный на рисунке 21, показывает, что стохастический градиентный спуск сходится точнее и за меньшее количество эпох (итераций) в сравнении с градиентным спуском. Однако сходимость стохастического градиентного спуска имеет менее стабильный характер по точности. Это связано со стохастичность (случайностью) выбора признаков для подвыборки.

Кроме того, проанализировав график 22, можно сделать вывод, что стохастический градиентный спуск обеспечивает более быстрое и качественное достижение минимального значения функции потерь по сравнению с градиентным спуском.

В связи с такими наблюдениями при обучении лучших моделей будем использовать именно стохастический градиентный спуск.

3.7 Лемматизация

Для повышения точности анализа текста, а также снижения размерности данных часто используют лемматизацию текста. Произведём её с помощью WordNetLemmatize из библиотеки ntlk, а именно выполним следующие действия:

- Приведём все слова в начальную форму;
- Удалим стоп-слова из данных.

Также нормализуем выборку.

После данной предобработки размерность признакового пространства уменьшилась с 3736 до 3399.

Метод	Время до лематизации (с.)	Время после лематизации (с.)
Стох. град. спуск	8.75	9.35
Градиентный спуск	14.83	14.76

В данном эксперименте, как и во всех остальных время работы алгоритма 3 раза измеряется и считается среднее.

Из приведённой таблицы можно сделать вывод, что лемматизация немного увеличивает время выполнения стохастического градиентного спуска, но практически не изменяет время выполнения градиентного спуска (небольшое уменьшение).

Скорее всего, это связано с тем, что лемматизация изменяет структуру признакового пространства, что сказывается на количестве итераций за одну эпоху и выборе случайных векторов для подвыборки в случае стохастического градиентного спуска. А градиентный спуск всегда обрабатывает всю выборку целиком, и для него снижение размерности признакового пространства оказывает положительное влияние на эффективность.

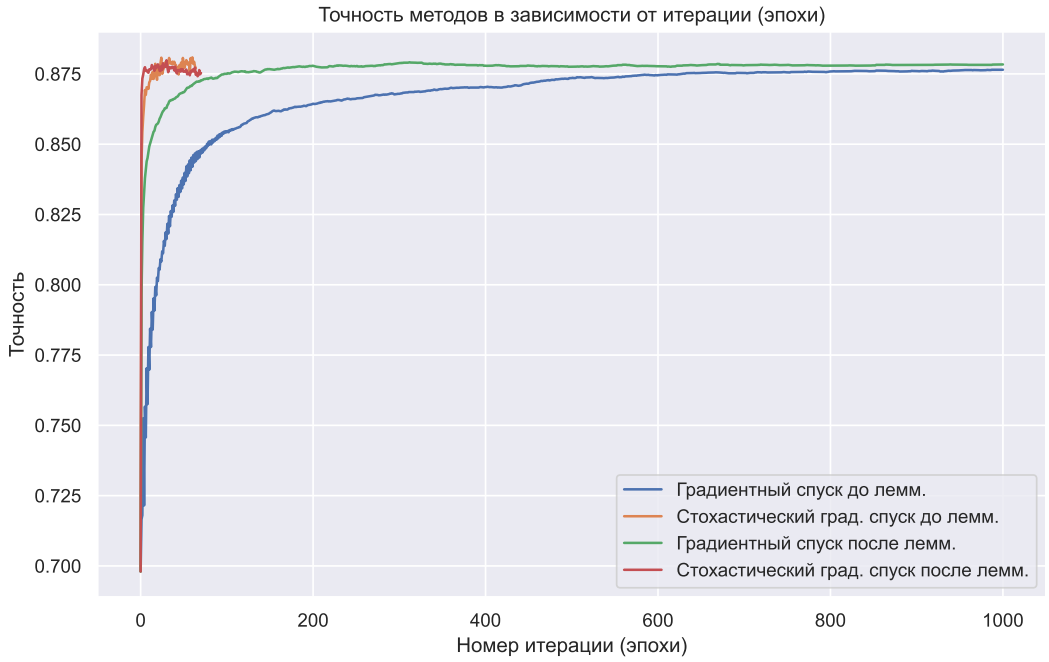


Рис. 23: Зависимость точности от номера итерации(эпохи)

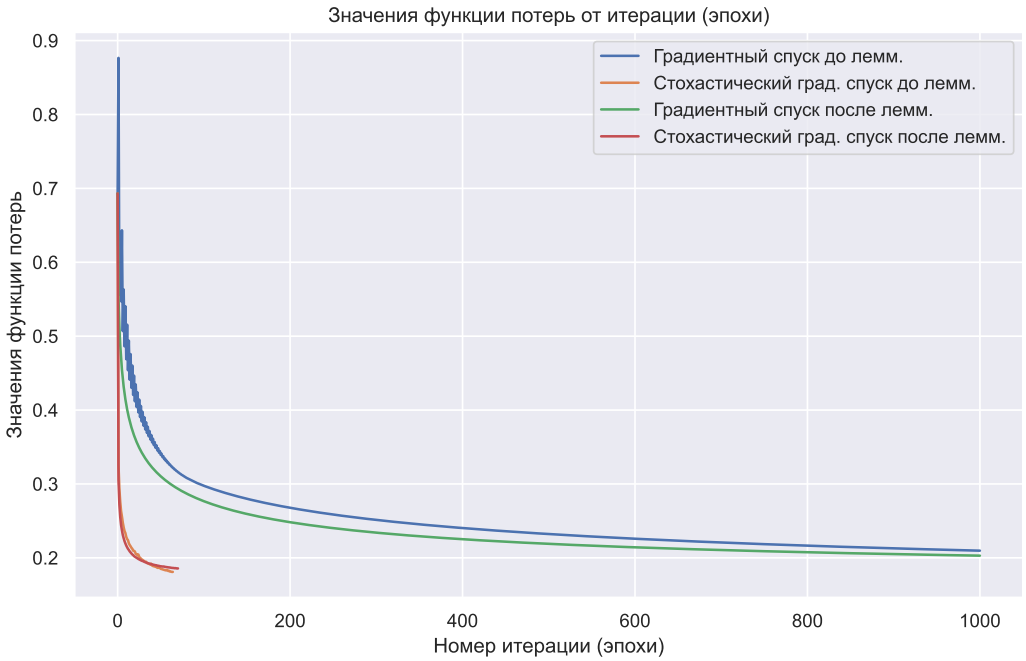


Рис. 24: Зависимость значения функции потерь от итерации(эпохи)

По графикам 23 и 24 заметно, что использование лемматизации значительно улучшило сходимость и точность градиентного спуска уже с первых итераций.

Что касается стохастического градиентного спуска, после применения лемматизации он стал более стабильным, хотя и немного потерял в точности. Однако в целом его результаты практически не изменились.

В итоге можно сказать, что лематизация положительно влияет на модель.

3.8 Представления BagOfWords и Tfidf

Подберём коэффициент регуляризации для каждого из представлений, остальные параметры модели оставим без изменений 3.6.

3.8.1 BagOfWords

Подбор коэффициента регуляризации:

l2_coef	Точность SGD	Точность GD
0.0005	0.853985296962662	0.8563068291739214
0.001	0.848906945250532	0.8484716579609208
0.01	0.8114722383439736	0.8177597214161346
0.1	0.7659605339524086	0.6980557167730702

Следовательно, будем использовать коэффициент 0.0005, дающий наибольшую точность.

Будем перебирать следующие значения для `min_df` и `max_df`:

`min_df = [10^(-6), 10^(-4), 10^(-3), 10^(-2), 10^(-1)]`

`max_df = [0.3, 0.5, 0.7, 0.8, 0.9]`

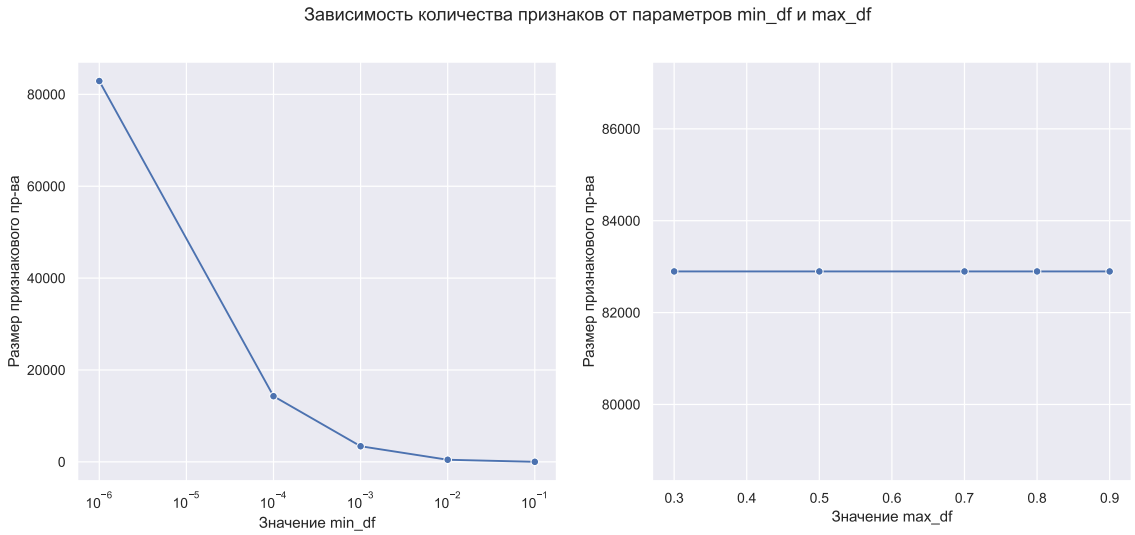


Рис. 25: Зависимость количества признаков от `min_df` и `max_df`

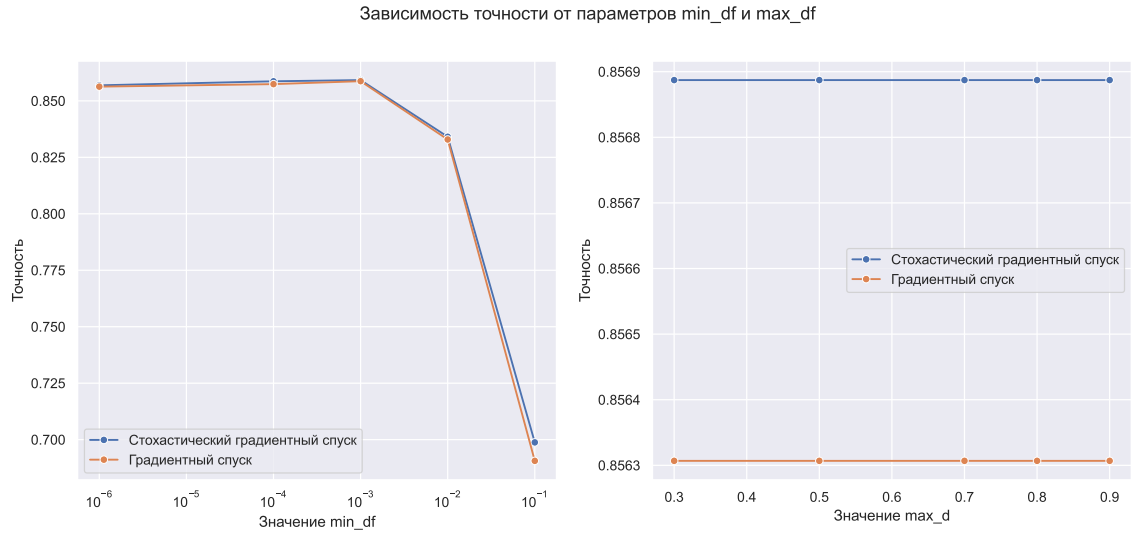


Рис. 26: Зависимость точности от `min_df` и `max_df`

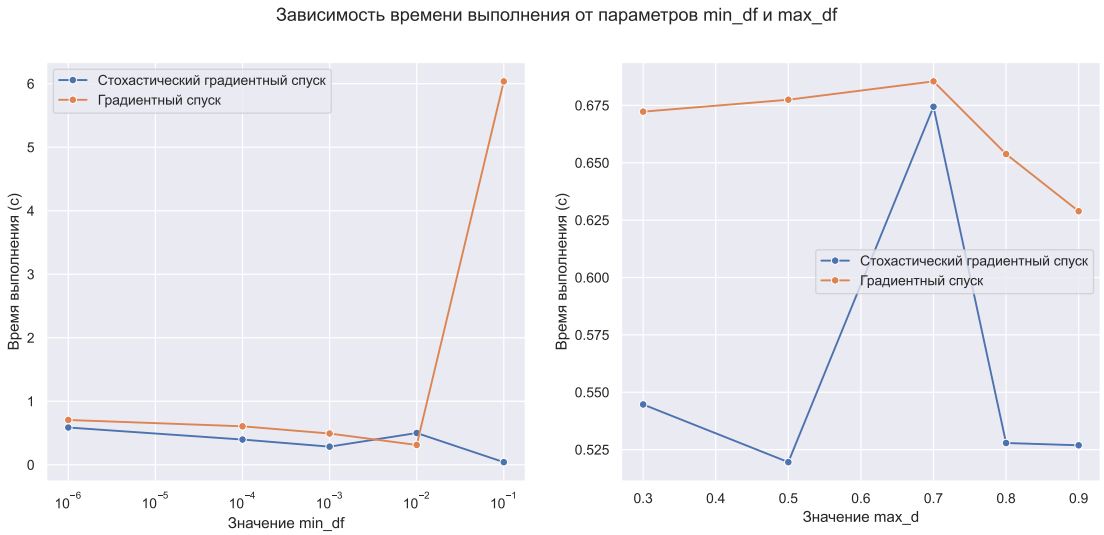


Рис. 27: Зависимость времени выполнения от `min_df` и `max_df`

По графику 25 заметно, как сильно влияет параметр `min_df` на размерность признакового пространства, и то, что выбранные значения `max_df` никак не повлияли на размерность. Это говорит о том, что в наших данных нет слов, встречающихся чаще, чем в 30% документов.

График 26 показывает, что при больношм значении `min_df`, то есть при сильном сужении признакового пространства точность сильно падает. Изменение `max_df` никак не влияет на результат по причине, описанной выше.

На графике 27 представлена зависимость времени выполнения от значений параметров. Анализируя его и предыдущие 2 графика, можно сделать вывод, что оптимальными по точности, размеру признакового пространства и времени выполнения для представления BagOfWords являются значения `min_df` $= 10^{-3}$, `max_df` $= 0.5$.

3.8.2 Tfidf

Подбор коэффициента регуляризации:

l2_coef	Точность SGD	Точность GD
0.005	0.839765912168698	0.838556780808667
0.001	0.8587250918939834	0.8609982588508416
0.01	0.8306732443412652	0.829609208744438
0.1	0.7980266976204294	0.6980557167730702

Следовательно, будем использовать коэффициент 0.001, дающий наибольшую точность.

Будем перебирать аналогично эксперименту с BagOfWords следующие значения для `min_df` и `max_df`:

```
min_df = [10^(-6), 10^(-4), 10^(-3), 10^(-2), 10^(-1)]
max_df = [0.3, 0.5, 0.7, 0.8, 0.9]
```

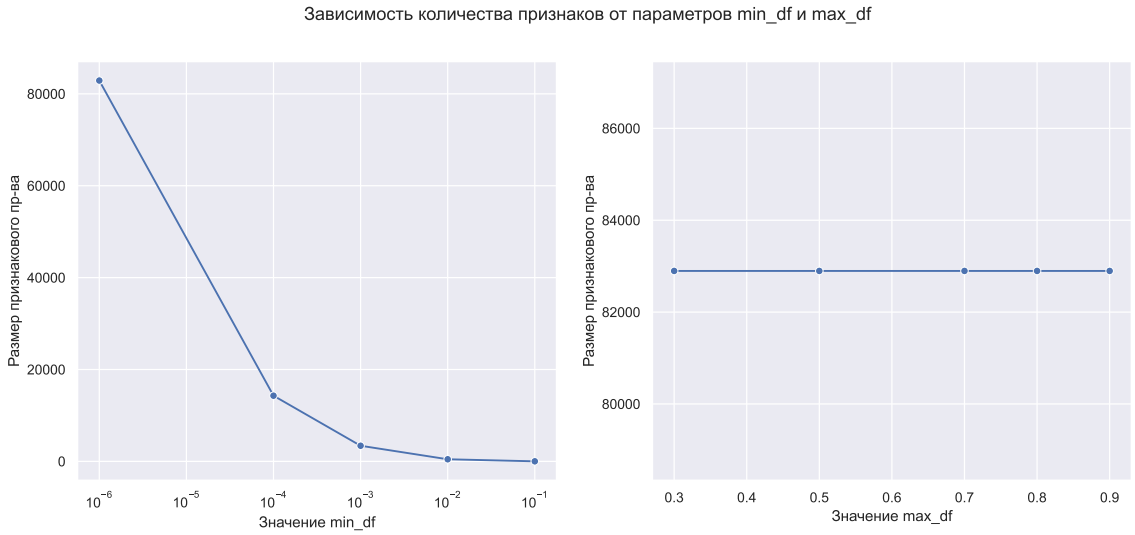


Рис. 28: Зависимость количества признаков от `min_df` и `max_df`

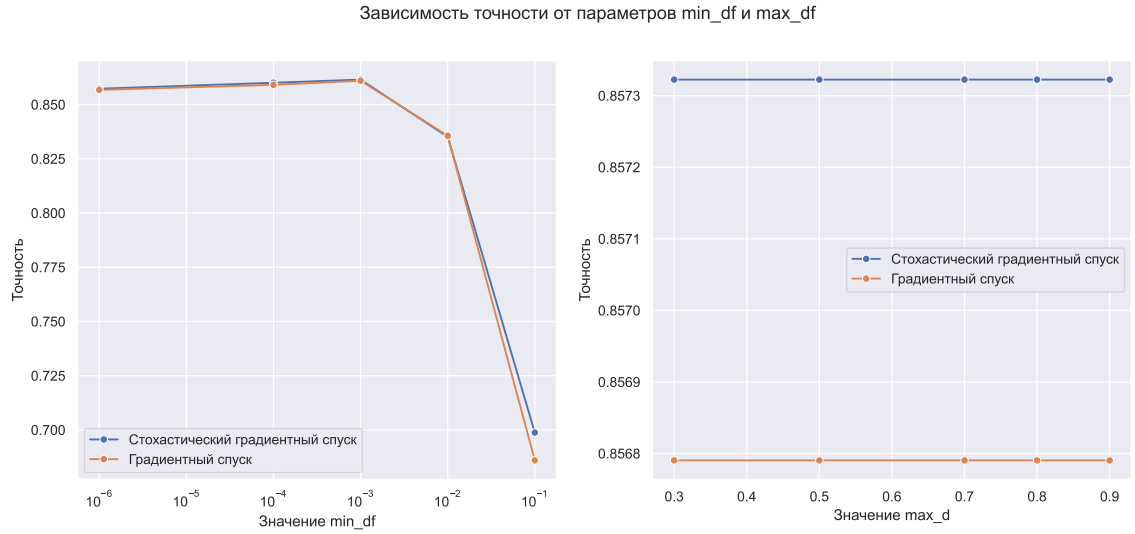


Рис. 29: Зависимость точности от `min_df` и `max_df`

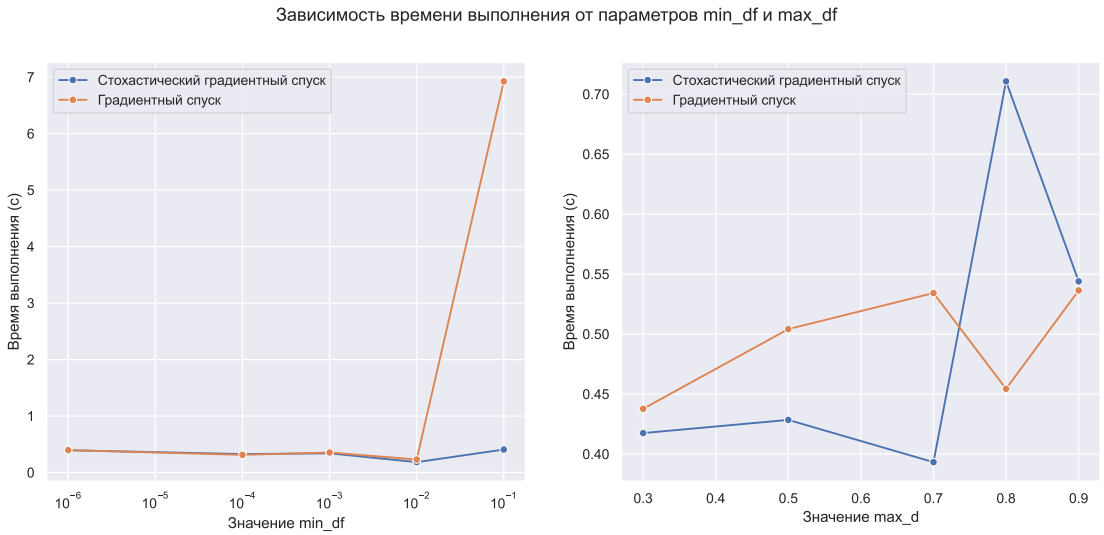


Рис. 30: Зависимость времени выполнения от `min_df` и `max_df`

Графики 28, 29 и 30 схожи с графиками из эксперимента с представлением BagOfWords. Проанализировав их, можно сделать вывод, что оптимальными значениями для параметров являются `min_df` $= 10^{-3}$, `max_df` $= 0.7$.

3.8.3 Сравнение BagOfWords и Tfidf

Проведем сравнение моделей, обученных на выборках, основанных на двух различных представлениях данных. В ходе эксперимента получили следующие результаты:

Метод и представление	Точность
SGD BoW	0.8593538402011994
GD BoW	0.8587250918939834
SGD Tfidf	0.8587250918939834
GD Tfidf	0.8609982588508416

Все модели показали очень близкие результаты, что говорит о целесообразности использования любого из данных представлений, но наилучшую точность показала модель, обученная градиентным спуском на представлении Tfidf.

3.9 Выбор лучшей модели и анализ ошибок

3.9.1 Перебор моделей

Обучение всех моделей осуществлялось с помощью стохастического градиентного спуска. Выборки обоих представлений были нормализованы. Всего было проверно 96 моделей, все комбинации следующих параметров:

- `alphas` = `[5, 10, 50, 90]`
- `beta` = `[0, 0.1, 0.5, 0.7]`
- `l2_coefs` = `[0.001, 0.0005, 0.01]`
- представления данных: BagOfWords, Tfidf с оптимальными параметрами из предыдущего эксперимента.

alpha	**beta**	**l2_coef**	**Представление**	**Точность**
10	0.1	0.0005	tfidf	0.869317
10	0	0.0005	tfidf	0.869027
5	0.1	0.0005	tfidf	0.868833
90	0.5	0.0005	tfidf	0.868592
90	0.1	0.0005	tfidf	0.868495
50	0.1	0.0005	tfidf	0.868350
5	0	0.0005	tfidf	0.868156
50	0.5	0.0005	tfidf	0.868156
90	0.7	0.0005	tfidf	0.867576
50	0	0.0005	tfidf	0.866802

Таблица 1: Топ 10 лучших моделей

Как видно из результатов всё-таки наилучшая точность достигается при представлении Tfidf.

3.9.2 Анализ ошибок

Проанализируем ошибки лучшей модели. Для этого построим облака слов.

Облако слов для ложных отрицательных предсказаний

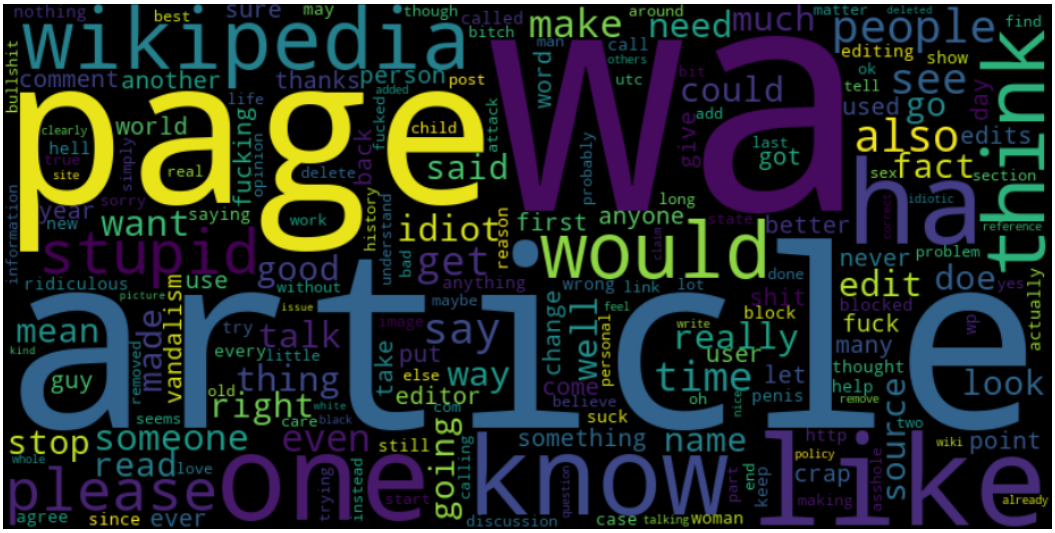


Рис. 31: Облако ложно отрицательных предсказаний

Облако слов для ложных положительных предсказаний

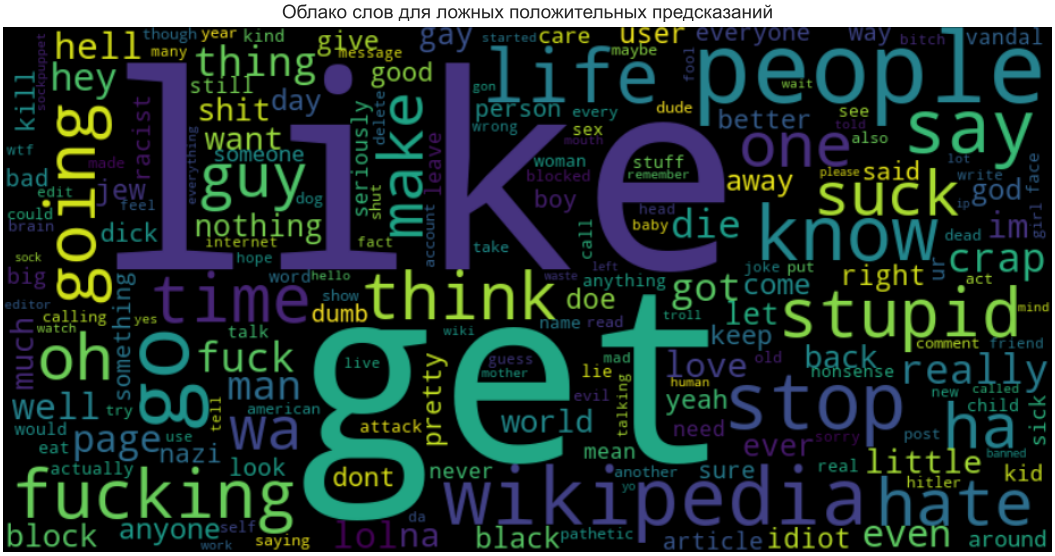


Рис. 32: Облако ложно положительных предсказаний

Из визуализаций ошибок, представленных в 31 и 32, можно сделать вывод, что модель в основном ошибается на часто используемых в фразах глаголах, таких как like, get, know, а также на часто встречающихся нейтральных словах (например, wikipedia, page, article, people). Также стоит отметить, что среди ошибок

выделяются некоторые междометия, которые не удалось эффективно удалить с помощью лематизации.

3.10 Добавление в признаковое пространство n-грамм

N-граммы — это последовательности из n подряд идущих слов или символов в тексте. Например, 1-граммы (униграммы) представляют собой отдельные слова, биграммы — пары соседних слов, а триграммы — тройки соседних слов. Использование n-грамм полезно в анализе текстов, потому что они позволяют учитывать контекст слов, что важно для более полного понимания текста.

Попробуем добавить в признаковое пространство:

- только униграммы
- униграммы и биграммы
- униграммы, биграммы и триграммы
- только биграммы

Проведём эксперимент на лемматизированных данных.

Время (с)	**Точность модели**	**n-граммы**
0.9343	0.8653	1-граммы
50.9574	0.8509	1-граммы, 2-граммы
123.0131	0.8424	1-граммы, 2-граммы, 3-граммы
17.0383	0.7998	2-граммы

Таблица 2: Время, точность модели и n-граммы

Из приведённой таблицы видно, что размер максимальных n-грамм, а также количество их видов значительно увеличивают время работы алгоритма, при этом улучшений в точности не происходит. Возможно, это связано с тем, что токсичность в тексте вносят главным образом отдельные слова, а не целые фразы. Следовательно, использование более высоких n-грамм в данной задаче может не приносить значительных улучшений, а наоборот, лишь усложнять вычисления.

Проведём аналогичный эксперимент, но на не лемматизированных данных. Возможно, отсутствие улучшений связано с удалением стоп-слов и приведением всех остальных слов к начальной форме.

Время (с)	**Точность модели**	**n-граммы**
1.1831	0.8405	1-граммы
13.3189	0.816	2-граммы
118.4265	0.7998	3-граммы
13.4780	0.79164	4-граммы

Таблица 3: Результаты эксперимента на не лемматизированных данных

Исходя из приведённых результатов эксперимента, можно сделать вывод, что лемматизация данных улучшает работу алгоритма. В то время как использование N-грамм не приводит к улучшению результатов, и может оказаться излишним.

4 Заключение

В ходе выполнения практического задания были теоретически обоснованы формулы градиентов, реализованы градиентный и стохастический градиентные спуски, проведено множество экспериментов по подбору оптимальных параметров алгоритмов. Результаты экспериментов показали, что стохастический градиентный спуск, несмотря на свою более высокую вариативность, обеспечивает значительное ускорение при обработке больших объемов данных, что делает его более эффективным для задач с большими выборками, а также может показывать наибольшую точность. В то время как градиентный спуск требует большего времени на вычисления, но демонстрирует более стабильную сходимость. В целом, обе методики показали свою применимость для задачи классификации токсичных текстов, и выбор между ними зависит от конкретных условий задачи.