Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Кафедра ММП, компьютерный практикум

Отчет о выполненном задании «Линейные модели»

Травникова Арина Сергеевна, 317 группа

Введение.

Задание заключается в реализации алгоритма линейной классификации с градиентным методом обучения и исследовании его работы на датасете англоязычных коммментариев из раздела обсуждений Википедии. В процессе работы проведены различные эксперименты для сравнения эффективности и качества работы алгоритмов с разными параметрами и подбора оптимальных гиперпараметров. Также выполнено предсказание для тестовой выборки.

Постановка задачи линейной классификации.

Бинарная классификация.

Линейный метод бинарной классификации заключается в построении разделяющей гиперплоскости в пространстве признаков. Класс объекта определяется его положением относительно проведенной плоскости.

Пусть дана обучающая выборка с известными ответами $\{(x_i, y_i)\}_{i=1}^N$, где $x_i \in \mathbb{R}^{d+1}$, $y_i \in \mathbb{Y} = \{-1, +1\}$, и выборка объектов $\{z_i\}_{i=1}^M$, $z_i \in \mathbb{R}^{d+1}$, для которых необходимо предсказать ответ. Предполагается, что на множестве объектов введен константный признак, принимающий значение 1 на каждом объекте.

Предсказание класса для произвольного объекта z определяется формулой $a(z) = \text{sign}\,\langle w, z\rangle$, где $w \in \mathbb{R}^{d+1}$- вектор весов классификатора, настраеваемый в процессе обучения.

Обучение модели заключается в численной минимизации функционала эмпирического риска:

$$Q(w) = \frac{1}{N}\sum_{i=1}^{N} L(M_i(w)) + \frac{\lambda}{2}\|w\|_2^2 \to \min_w,$$

где $L(M_i)$ - функция потерь, $M_i = w^{\mathrm{T}} x_i y_i$ - отступ для объекта x_i, λ - коэффициент регуляризации.

Далее в работе используется логистическая функция потерь $L(M_i) = \log(1 + e^{-M_i})$.

Помимо предсказания метки класса для объекта логистическая регрессия позволяет корректно предсказывать его вероятность:

$$P(y = +1|x) = \frac{1}{1 + e^{-w^{\mathrm{T}} x}} = \sigma(w^{\mathrm{T}} x), P(y = -1|x) = 1 - P(y = +1|x) = \sigma(-w^{\mathrm{T}} x)$$

Для решения задачи минимизации функционала Q градиентным методом требуется вычисление градиента $\nabla_w Q$. Выведем формулы для вычисления градиента функций L и Q:

$\frac{\partial Q(w)}{\partial w_i} = \frac{1}{N} \sum\limits_{j=1}^{N} \frac{\partial L_j}{\partial w_i} + \lambda w_i$

$\frac{\partial L_j(w)}{\partial w_i} = \frac{\partial L_j}{\partial M} \frac{\partial M_j(w)}{\partial w_i} = -\frac{y_j x_j^i}{e^{M_j} + 1} = -y_j x_j^i \sigma(-M_j)$, здесь $L_j = L(M_j) = \log(1 + e^{-w^{\mathrm{T}} x_j y_j})$

$\nabla_w Q = \left(-\frac{1}{N} \sum\limits_{j=1}^{N} y_j x_j^i \sigma(-w^{\mathrm{T}} x_j y_j) + \lambda w_i \right)_{i=1}^{d+1}$.

Многоклассовая классификация.

В случае C классов вероятность принадлежности классу $k \in \mathbb{Y} = \{1, \ldots, C\}$ для произвольного объекта $x \in \mathbb{R}^{d+1}$ определяется выражением $P(y = k|x) = \frac{e^{w_k^{\mathrm{T}} x}}{\sum\limits_{i=1}^{C} e^{w_i^{\mathrm{T}} x}}$, где $w_i \in \mathbb{R}^{d+1}$ - вектор весов, соответствующий бинарному классификатору $a_i(x) = \operatorname{sign} \langle w_i, x \rangle$, который оценивает вероятности принадлежности объекта x классу i. Положим $W = \left(w_1, \ldots, w_c \right)^{\mathrm{T}} \in \mathbb{R}^{C \times d+1}$ - матрица весов.

Аналогично модели бинарной классификации обучение заключается в численной минимизации функционала эмпирического риска:

$$Q(W) = -\frac{1}{N} \sum_{i=1}^{N} \log P(y = y_i|x_i) + \frac{\lambda}{2} \sum_{k=1}^{C} \|w_k\|_2^2 \to \min_{w_1, \ldots, w_c}.$$

Выведем формулы для градиента фунционала Q:

Пусть $L(W, x_i, y_i) = -\log P(y = y_i|x_i) = -\log \frac{e^{w_{y_i}^{\mathrm{T}} x_i}}{\sum\limits_{k=1}^{C} e^{w_k^{\mathrm{T}} x_i}} = -\log e^{w_{y_i}^{\mathrm{T}} x_i} + \log \sum\limits_{k=1}^{C} e^{w_k^{\mathrm{T}} x_i} =$

$-w_{y_i}^{\mathrm{T}} x_i + + \log \sum\limits_{k=1}^{C} e^{w_k^{\mathrm{T}} x_i}$

$\frac{\partial L(W, x_i, y_i)}{\partial w_l^j} = -\mathbb{1}[y_i = l] x_i^j + \frac{x_i^j e^{w_l^{\mathrm{T}} x_i}}{\sum\limits_{k=1}^{C} e^{w_k^{\mathrm{T}} x_i}}, l \in [1, C], j \in [1, d+1]$

$$\frac{\partial Q(W)}{\partial w_l^j} = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial L(W,x_i,y_i)}{\partial w_l^j} + \lambda w_l^j = \frac{1}{N}\sum_{i=1}^{N}(-\mathbb{1}[y_i = l]x_i^j + \frac{x_i^j e^{w_l^{\mathrm{T}}x_i}}{\sum\limits_{k=1}^{C} e^{w_k^{\mathrm{T}}x_i}}) + \lambda w_l^j$$

Покажем, что при $C = 2$ мультиномиальная логистическая регрессия сводится к бинарной:

$$P(y = 1|x) = \frac{e^{w_1^{\mathrm{T}}x}}{e^{w_1^{\mathrm{T}}x} + e^{w_2^{\mathrm{T}}x}} = \frac{1}{1 + e^{(w_2 - w_1)^{\mathrm{T}}x}} = \sigma((w_1 - w_2)^{\mathrm{T}}x)$$

Выражение для вероятности класса 1 совпадает с соответствующей формулой для бинарной логистической регрессии с вектором весов $w = w_1 - w_2$.

Эксперимент 1, 2.

Датасет для исследования состоит из англоязычных комментариев раздела обусждений Википедии: 52061 объектов выделено для обучения и 20676 - для теста. Задача бинарной классификации заключается в определении, является ли некоторый комментарий токсичным или нет.

Эксперименты состоят в предварительной обработке текстовых данных. Для этого выполнено приведение текста к нижнему регистру, удаление всех символов, кроме букв английского алфавита, цифр и пробелов.

С помощью преобразования BagOfWords текстовые данные переведены в числовые: каждому слову некоторого документа ставится в соответствие число, сколько раз оно встречается в этом документе. С целью уменьшения размерности признакового пространства в работе учитываются только те слова, которые втречаются не менее, чем в 5 комментариях. Итоговое число признаков после описанных преобразований - 18254.

Эксперимент 3.

Для решения задачи численной минимизации функционала эмпирического риска $Q(w)$ используется метод градиентного спуска, основанный на итеративном алгоритме. На первом шаге выбирается начальное приближение вектора весов, а далее на каждой итерации он изменяется в направлении антиградиента $Q(w)$: $w_{k+1} = w_k - \frac{\alpha}{k^\beta}\nabla_w Q, k \geq 0$.
Итеративный процесс останавливается либо по истечении максимального времени работы (количества итераций), либо при слишком малом изменении фунционала $Q(w)$ за итерацию - сходимости метода.

Эксперимент состоит в исследовании зависимости точности предсказания и значения функции потерь от номера итерации и времени работы градиентного спуска при различных настраеваемых гиперпараметрах:

- $\alpha \in [0.001, 0.01, 0.1, 0.8, 1, 2]$ - размер шага градиентного спуска
- $\beta \in [0, 0.01, 0.1, 0.5, 1]$ - размер шага градиентного спуска
- $w_0 \in [(0, \ldots, 0), (random), (1, \ldots, 1)]$- начальное приближение вектора весов

Под точностью алгоритма будем понимать долю правильно предсказанных ответов. Градиентный спсук проводится по обучающей выборке, измерение точности - по тестовой.

Исследование зависимости поведения работы градиентного спуска при различных α проводится при заданных $w_0 = 0, \beta = 0$. Результаты приведены на рис. 1.

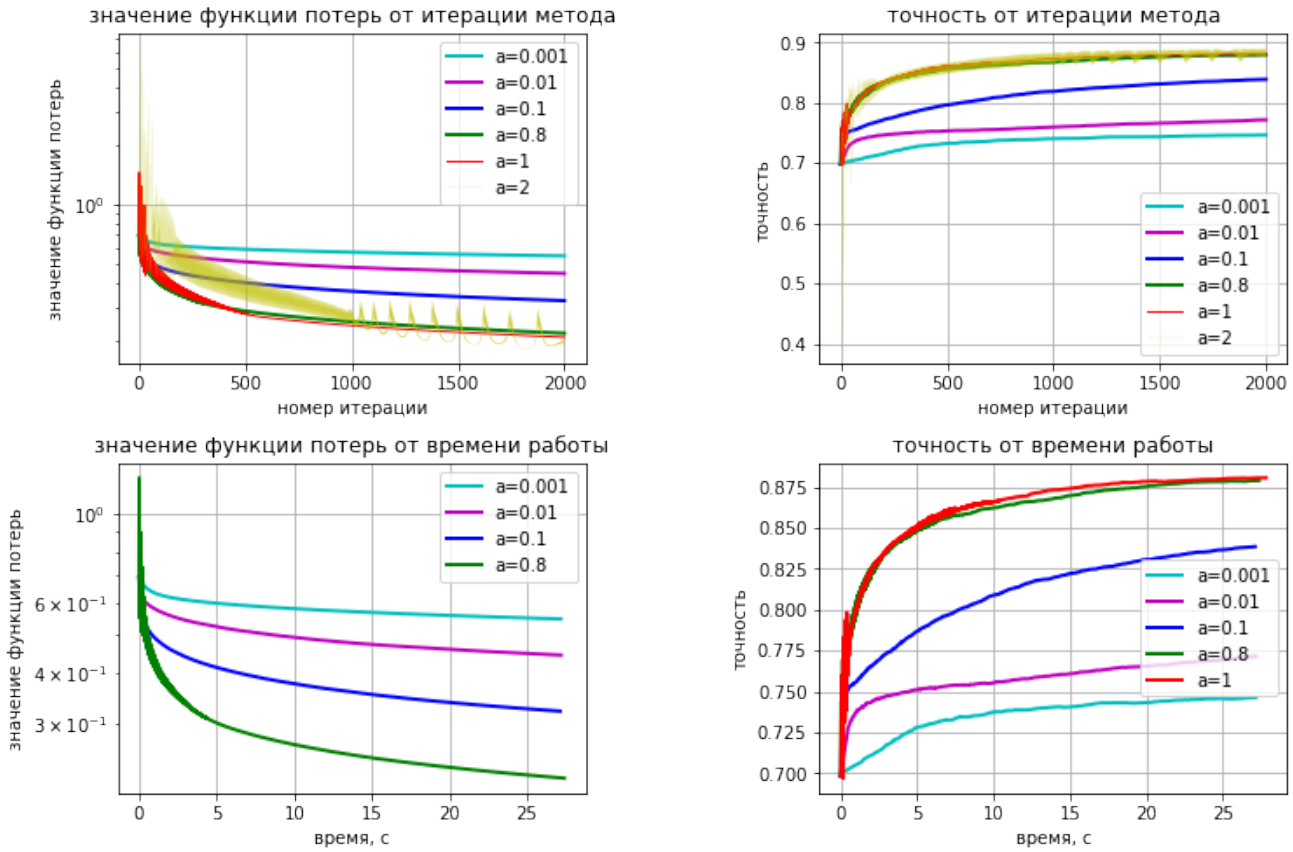


Рис. 1: Поведение функции потерь и точности алгоритма логистической регрессии с обучением методом градиентного спуска при различных значениях шага α.

Графики показывают, что с ростом $\alpha \leq 1$ значение функционала ожидаемых потерь уменьшается, а точность модели на тестовых данных, соответственно, увеличивается. При малых параметрах α алгоритм сходится очень медленно, так как на каждом шаге слабо учитывается значение градиента $Q(w)$, то есть алгоритм неэффективен, а при $\alpha \geq 1$ значение функции потерь колеблется и не достигает своего минимума – метод расходится. Это объясняется тем, что из-за большого коэффициента при антиградиенте, функционал каждую итерацию может перешагивать через свой минимум, так и не достигая его.

Можно заметить, что лучшая точность достигается при выборе $\alpha = 1$. Однако при количестве итераций до 500 оптимизируемый функционал сильно осциллирует, и в таком случае имеет смысл выбрать меньшее α, например, 0.8. Так как далее в экспериментах максимальное число итераций устанавливается большим, выбран параметр $\alpha = 1$ с целью достижения лучшей точности.

Исследование зависимости поведения работы градиентного спуска от параметра β проводится при заданных $w_0 = 0, \alpha = 1$. Результаты приведены на рис. 2.

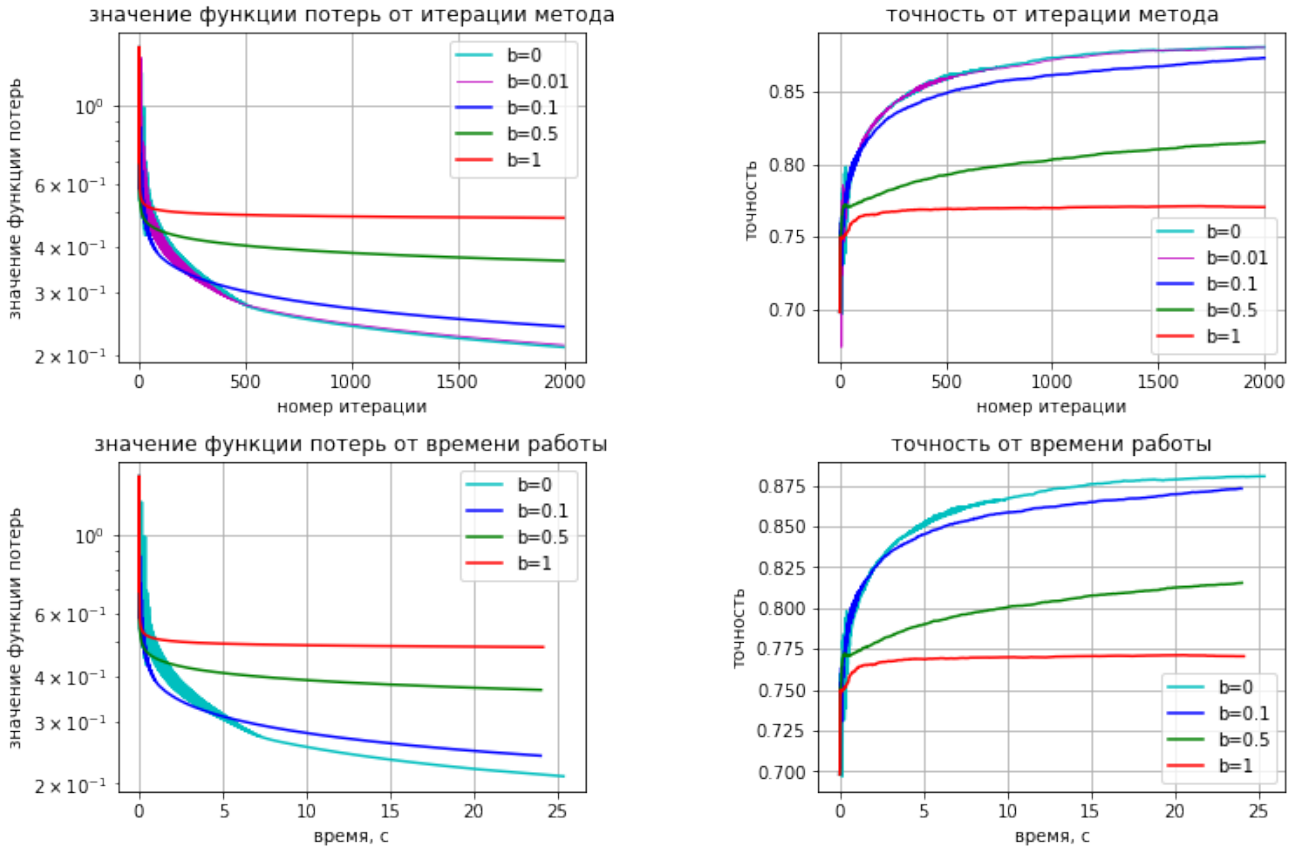


Рис. 2: Поведение функции потерь и точности алгоритма логистической регрессии с обучением методом градиентного спуска при различных параметрах шага β.

Результаты показывают, что при числе итераций $k \geq 500$ с увеличением значения β точность алгоритма уменьшается, а значение функции эмпиричекого риска увеличивается. Это возникает из-за того, что при больших k, β множитель $\frac{1}{k^\beta}$ становится слишком мал и сходимость метода существенно замедляется, так как на каждом шаге величина градиента учитывается слишком слабо. В это случае оптимален выбор $\beta = 0$.

При небольшом количестве итераций алгоритма для $\beta < 0.1$ наблюдаются сильные колебания оптимизируемого функционала, что объясняется выбором $\alpha = 1$: при малых k, β коэффициент $\frac{\alpha}{k^\beta} \approx 1$ (см. рис. 1 при $\alpha = 1, \beta = 0$). При росте β : $\frac{\alpha}{k^\beta} < 1$ и функция перестает колебаться, то есть метод сходится даже при небольшом числе итераций.

Для достижения лучшей точности в дальнейших экспериментах выбран параметр $\beta = 0$, так как число итераций устанавливается большим.

Зависимость поведения градиентного спуска от выбора начального приближения вектора весов w_0 исследуется при фиксированных параметрах $\alpha = 1, \beta = 0$. Результаты работы показаны на рис. 3.

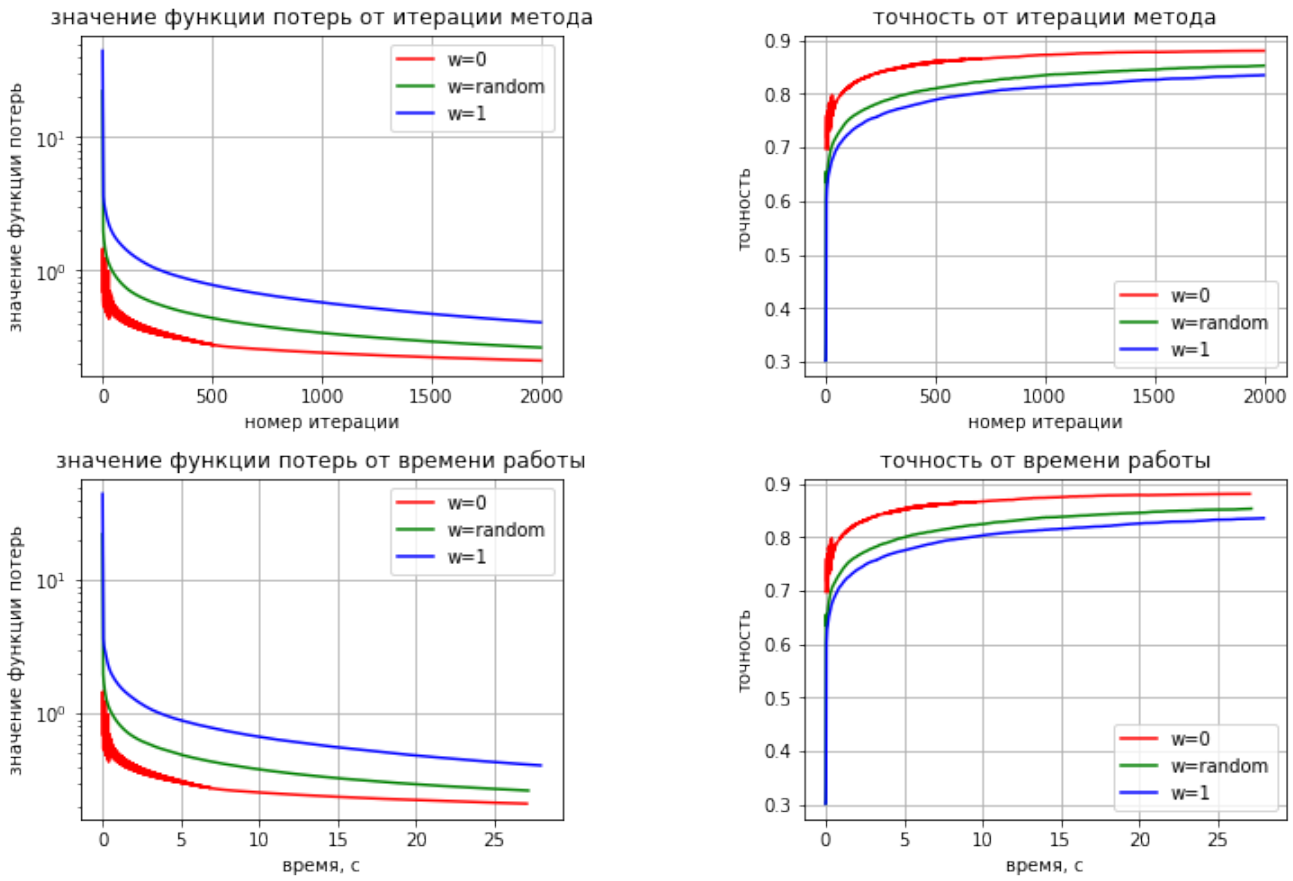


Рис. 3: Поведение функции потерь и точности алгоритма логистической регрессии с обучением методом градиентного спуска при различных параметрах начального приближения весов.

Эксперимент показывает, что при большом количестве итераций выбор начального приближения влияет только на скорость сходимости: при фиксированном числе шагов лучшую точность дает выбор нулевого начального приближения весов.

Применяя лучшие подобранные параметры: $alpha = 1$, $beta = 0$, $w_i = 0$ - для предсказания на тестовой выборке, получаем точность 88.80%

Эксперимент 4.

Модификацией метода градиентного спуска является стохастический градиентный спуск (SGDC), в котором на каждой итерации алгоритма градиент $Q(w)$ вычисляется не по всем объектам обучающей выборки, а лишь по ее случайно выбранному подмножеству фиксированного размера $batch\_size$, что уменьшает вычислительную сложность метода и время его работы. Одной итерации градиентного спуска сопоставим *эпоху* стохастического градиентного спуска - совокупность итераций обновления весов, которые нужно выполнить,

чтобы пройти по всей выборке.

Эксперимент состоит в исследовании зависимости точности предсказания и значения функции потерь от номера эпохи и времени работы стохастического градиентного спуска при различных настраеваемых гиперпараметрах:

- $\alpha \in [0.001, 0.01, 0.1, 0.5, 1, 2]$ - размер шага градиентного спуска
- $\beta \in [0, 0.01, 0.1, 0.5, 1, 5]$ - размер шага градиентного спуска
- $w_0 \in [(0, \ldots, 0), (random), (1, \ldots, 1)]$- начальное приближение вектора весов
- $batch\_size \in [50, 100, 1000, 10000]$ - размер подвыборки

Измерение точности алгоритма на тестовых данных и значения функции потерь при различных α производится при прочих фиксированных параметрах $\beta = 0, w_0 = 0$, $batch\_size = 1000$. Результаты представлены на рис. 4. Далее в работе приведены только графики зависимости функционалов от времени, так как соответствующие им графики по эпохам выглядят аналогично. Максимальное время работы - время выполнения 150 эпох SGDC.

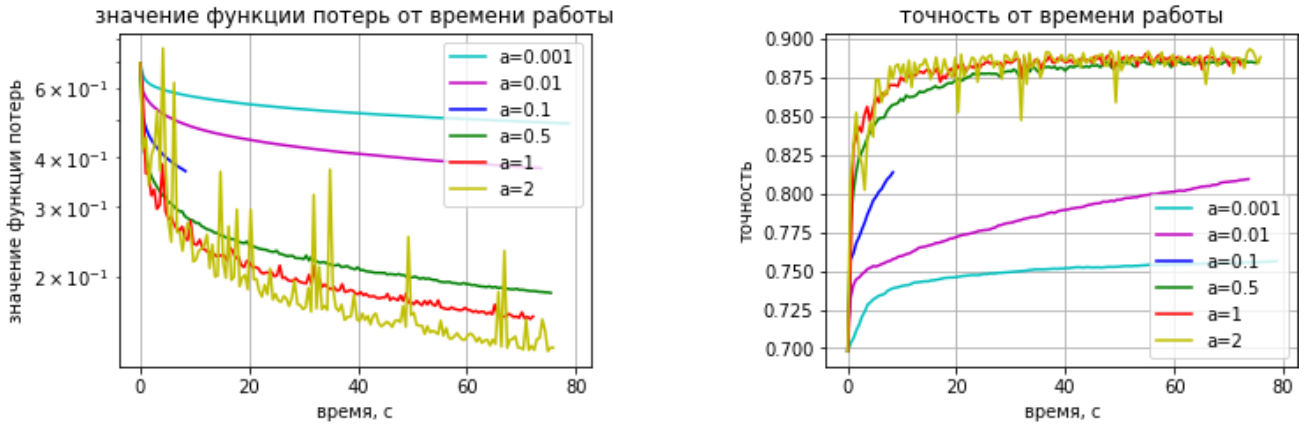


Рис. 4: Поведение функции потерь и точности алгоритма логистической регрессии с обучением методом SGDC при различных значениях шага α.

Эксперимент показывает, что поведение оптимизирумемых функций при стохастическом градиентном спуске в зависимости от параметра α схоже с их поведением при градиентном спуске, однако вид график более пилообразный. Это связано с тем, что на каждом шаге в градиентом спуске ошибка уменьшается на всех объектах выборки, а в стохастическом - лишь на части фиксированного размера, поэтому суммарная ошибка может возрастать на некоторых итерациях, но, тем не менее, уменьшается в среднем. С увеличением параметра α улучшается точность, но метод хуже сходится. При $\alpha > 1$, как и в градиентном спуске, сходимости нет. Для дальнейшей работы выбрано значение $\alpha = 1$.

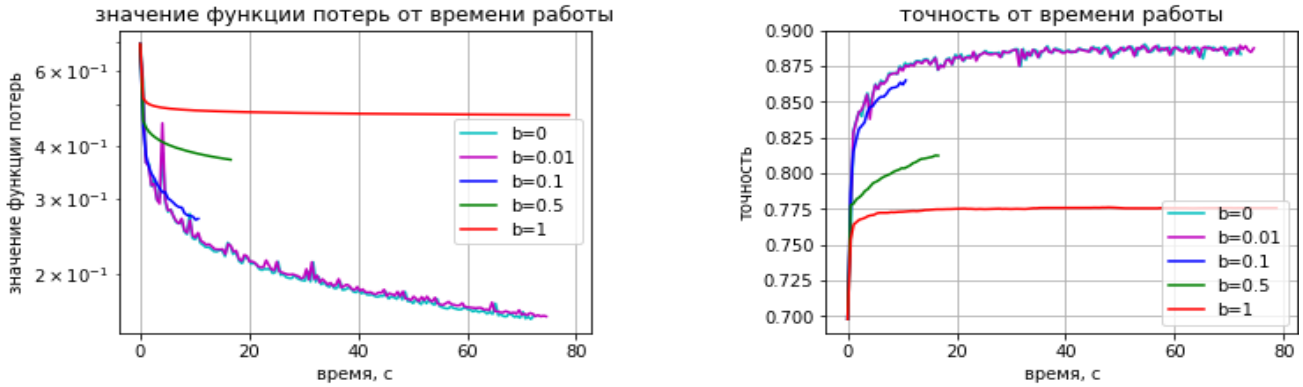


Рис. 5: Поведение функции потерь и точности алгоритма логистической регрессии с обучением методом SGDC при различных значениях шага β.

Исследование поведения алгоритма при различных значениях параметра β производится при фиксированных параметрах $\alpha = 1, w_0 = 0$, $batch\_size = 1000$. Результаты представлены на рис. 5.

Так же, как и для градиентного спуска, параметр β влияет на сходимость метода и точность работы. При больших $\beta > 0.5$ метод сходится очень медленно из-за того, что коэффицент $\frac{1}{k^\beta}$ мал. При малых $\beta < 0.1$ оптимизируемые функционалы колеблются в окрестности точки минимума в силу выбора α (см. рис. 4). При других значениях β метод останавливает свою работу до достижения минимума по критерию слишком малого изменения функции потерь за одну эпоху. Далее используется $\beta = 0.01$, так такое значение дает высокую точность, а метод более устойчив, чем при $\beta = 0$.

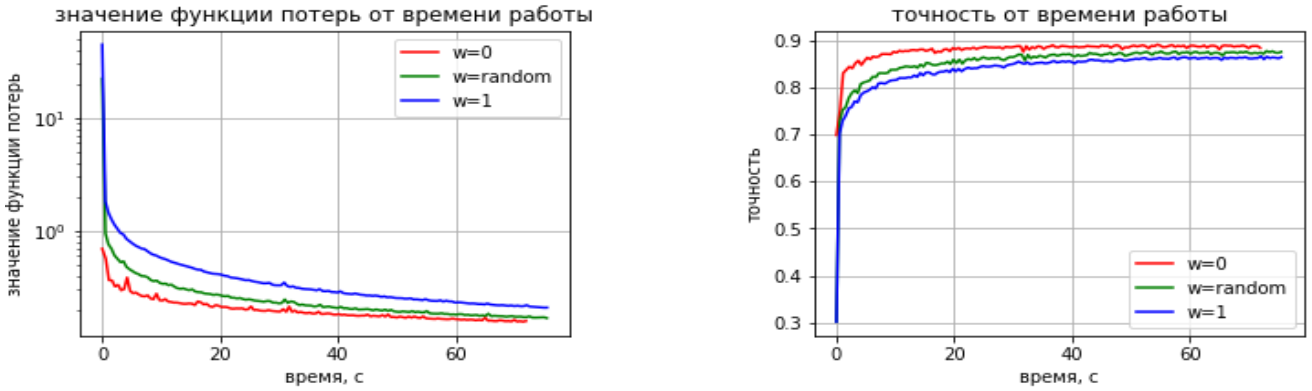


Рис. 6: Поведение значения функции потерь и точности алгоритма логистической регрессии с обучением методом SGDC при разных параметрах начального приближения весов.

Рис. 6 показывает зависимость значения функции потерь и точности алгоритма от времени работы стохастического градиентного спуска при различных значениях начального приближения вектора весов. Остальные настраеваемые параметры фиксированы:

$\alpha = 1, \beta = 0.01, batch\_size = 1000$. Как и в случае обычного градиентного спуска, лучшим начальным приближением вектора весов является нулевое.

На рис. 7 изображены графики зависимости функции потерь и точности алгоритма в зависимости от номера эпохи и времени работы SGDC при различных значениях $batch\_size$. Результаты показывают, что чем больше установлен размер подвыборки, тем быстрее работает метод в пределах одной эпохи. Это происходит благодаря векторизованными вычислениям градиента с помощью библиотеки *numpy*, эффективно реализующей матричные операции: чем больше размер матрицы, тем эффективнее вычисления. Также стоить отметить, что при большом $batch\_size$ метод проигрывает в точности и функционалы сходятся медленнее, а стохастический градиентный спуск переходит в градиентный спуск. Таким образом, оптимальный выбор $batch\_size = 1000$, так как он минимизирует время работы без потери в точности.

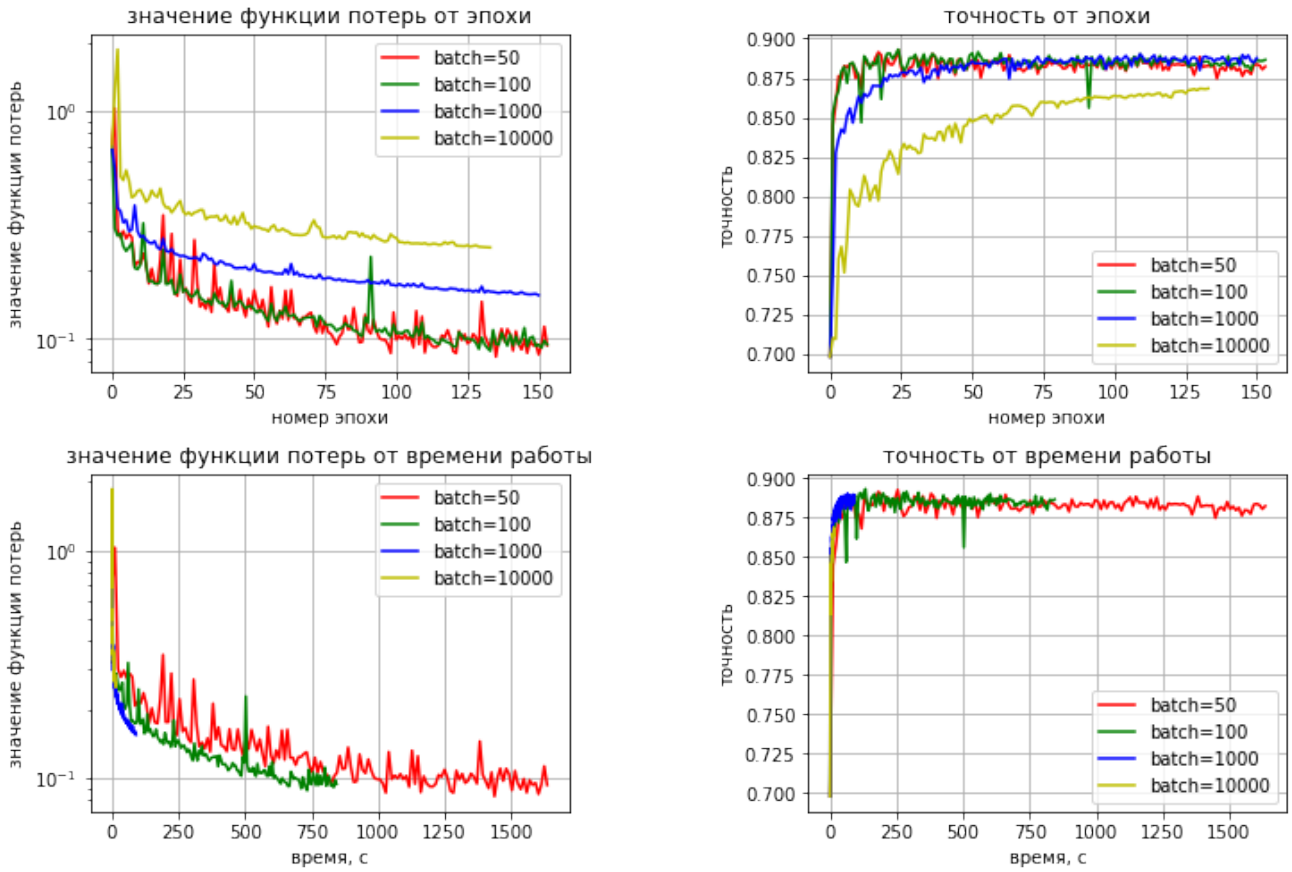


Рис. 7: Поведение функции потерь и точности алгоритма логистической регрессии с обучением методом SGDC при различных значениях $batch\_size$.

Применяя лучшие подобранные параметры: $alpha = 1, beta = 0.01, w_i = 0, batch\_size = 1000$ - для предсказания на тестовой выборке, получаем точность 89.06%

Эксперимент 5.

Эксперимент заключается в сравнении работы методов градиентного спуска и стохастического градиентного спуска при лучших подобранных параметрах из экспериментов 4, 5. Поведение функции потерь и точности метода в зависимости от времени работы изображено на рис. 8.

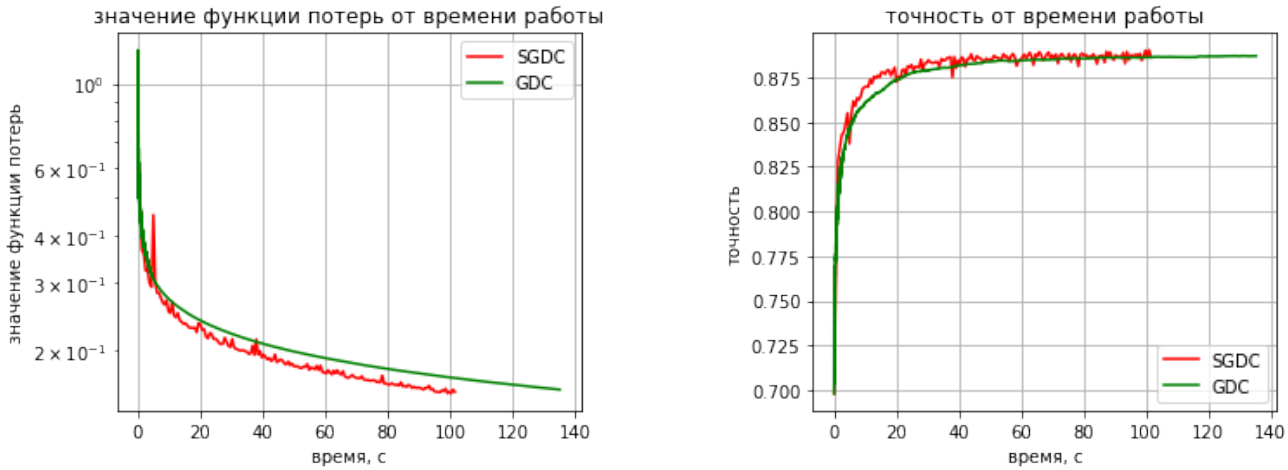


Рис. 8: Поведение значения функции потерь и точности алгоритма логистической регрессии с обучением методом GDC и SGDC при лучших подобранных параметрах.

Результаты показывают, что стохастический градиентный спуск дает значительный выигрыш по времени по сравнению с обычным градиентным спуском из-за того, что обновление весов на каждом шаге стохастического градиентного спуска требует гораздо меньшего времени, хотя вместе с этим и большего числа шагов, так как градиент считается лишь по части обучающей выборки.

Кроме того, можно заметить, что графики функций для градиентного спуска гладкие, а для стохастического - пилообразные. Это объясняется тем, что на каждой итерации градиентного спуска ошибка уменьшается на каждом объекте и функция потерь монотонно убывает, а на каждом шаге стохастического - ошибка уменьшается лишь на выбранном подмножестве объектов, по которым считается градиент, но она может возрасти на других объектах выборки, что и вызывает колебания функции потерь. Тем не менее, для стохастического градиентного спуска ошибка в целом уменьшается и модель достигает высокой точности. Это подтверждает и лучшая полученная точность: 88.80% для GDC и 89.06% для SGDC.

Эксперимент 6.

Эксперимент состоит в преобразовании обучающей и тестовой выборки путем применения лемматизации и удаления стоп-слов, а также изучении влияния преобразования к документам на точность предсказания классификатора.

После применения указанных преобразований размерность признакового пространства уменьшилась до 16234, то есть на 11%.

Лучшая точность алгоритма при обучении методом градиентного спуска при подобранных выше гиперпараметрах $\alpha = 1, \beta = 0, w_0 = 0$ составила 88.81%, что всего на 0.01% выше, чем на выборке без описанной обработки. При стохастическом градиентном спуске с параметрами $\alpha = 1, \beta = 0.001, w_0 = 0, batch\_size = 1000$ точность стала 88.92%, то есть уменьшилась на 0.14% относительно его применения к данным без обработки, но метод все равно остался точнее градиентного спуска. Возможно, незначительное падение точности для стохастического градиентного спуска связано со смысловыми особенностями задачи - для определения токсичности комментария употребление некоторых стоп-слов или же форм слова может оказаться важным признаком.

На рис. 9 изображены графики зависимости точности алгоритмов градиентного и стохастического градиентного спуска от времени работы для преобразованной и непреобразованной коллекции документов.

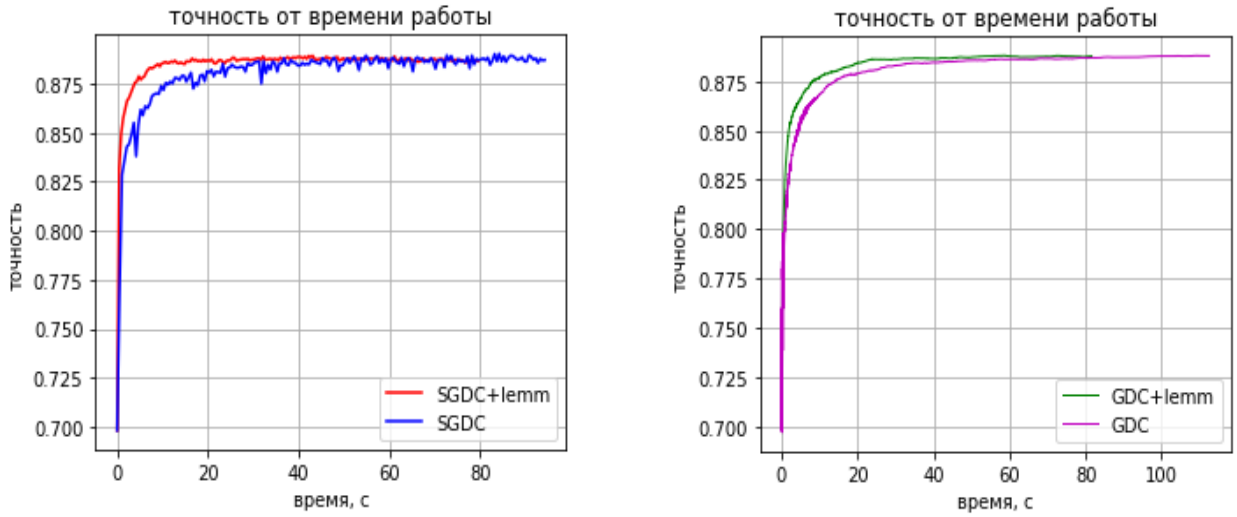


Рис. 9: График зависимости точности алгоритма логистической регрессии с обучением методоми GDC и SGDC для преобразованной *(+lemm)* и непреобразованной выборки от времени работы.

Результаты показывают, что после лемматизации коллекции и удалении стоп-слов время работы алгоритмов заметно уменьшилось. Это объясняется уменьшением размерности признакового пространства.

Эксперимент 7.

Эксперимент состоит в исследовании зависимости времени работы и точности классификатора в зависимости от следующих параметров преобразования исходной коллекции текстов в числовой вид:

- использование модели BagOfWords или Tfidf для преобразования текстов . Представление BagOfWords преобразует текстовые данные в числовые: каждому слову некоторого документа ставится в соответствие число, сколько раз оно встречается в этом документе. Tfidf также учитывает частоту слова во всей коллекции документов, что позволяет судить о важности слов.

- значение параметров min_df, max_df. Значение $min\_df(max\_df) = k$ означает, что из выборки отбрасываются те слова, которые встречаются менее(более), чем в k документах.

Применяя к выборке, полученной после удаления стоп-слов и лемматизации, представление Tfidf, найдем лучшую точность алгоритмов GDC и SGDC. При оптимальных подобранных параметрах: $\alpha = 5, \beta = 0, w_0 = 0, batch\_size = 1000$ - точность модели GDC составляет 89.19%, а SGDC - 89.27%, то есть по сравнению с представлением BagOfWords точность алгоритмов выросла на 0.38% и 0.22% соответственно.

Результаты измерения точности моделей SGD и SGDC для представлений текстовой коллекции в виде BagOfwords и Tfidf представлены на рис. 10. Графики подтверждают полученный прирост лучшей точности для метода TfIdf относительно BagOfWords.

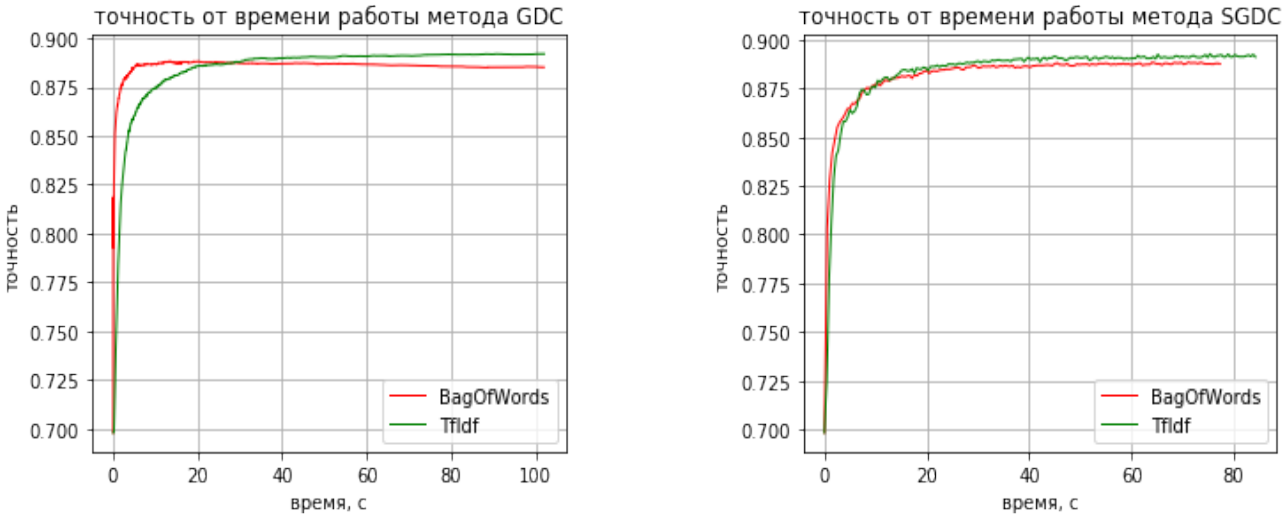
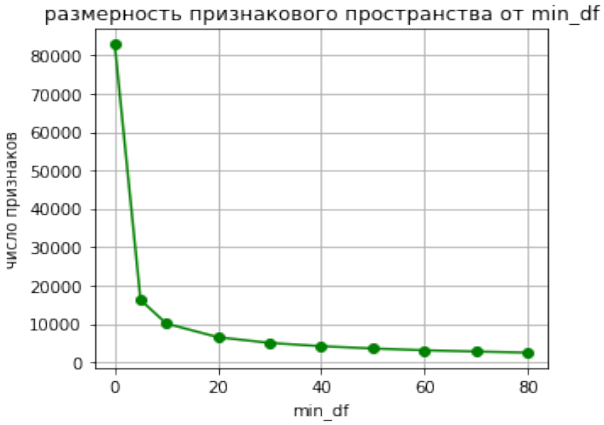


Рис. 10: График зависимости точности алгоритмов от времени при представлениях текста Tfidf, BagOfWords.

На рис. 11 изображена зависимость размерности пространства признаков от параметров min_df, max_df преобразования. Вид графиков объясняется определением величин min_df, max_df. Результаты показывают, что после удаления стоп-слов большенство слов коллекции являются уникальными - при большом max_df количество признаков почти не уменьшается и рост функции замедляется.



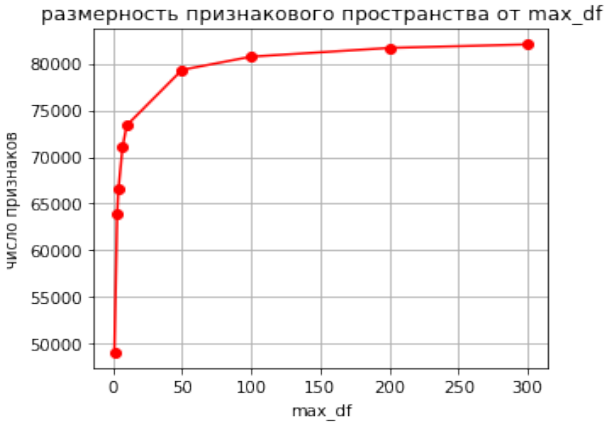
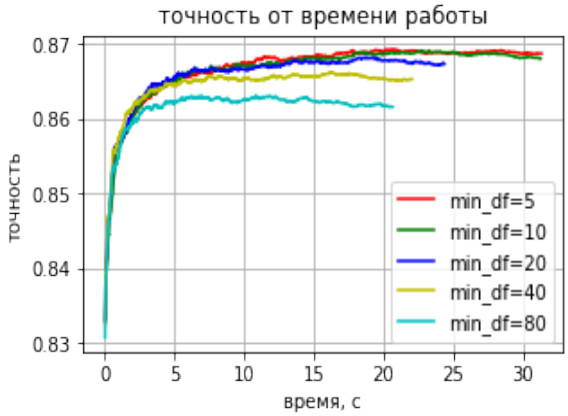


Рис. 11: График зависимости числа слов в коллекции документов от параметров min_df, max_df преобразования.



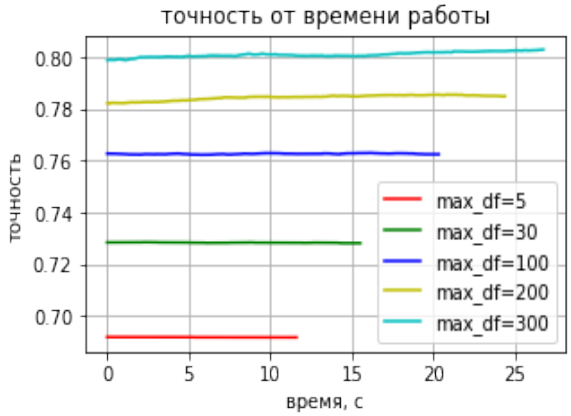


Рис. 12: График зависимости точности метода GDC с представлением TFidf от параметров min_df, max_df преобразования.

Рис. 12 иллюстрирует зависимость точности метода GDC с представлением Tfidf текстовой коллекции от времени работы алгоритма.

Результаты показывают, что с ростом значения min_df сокращается время работы метода из-за уменьшения числа признаков (см. рис. 10), а также уменьшается точность, так как отбрасываются не только уникальные слова документов, но и слова, встречающиеся в большом количестве документов, которые могут оказаться важными признаками. Аналогично с уменьшением max_df сокращается время работы метода и падает точность.

Таким образом, установленный параметр $min\_df = 5$ является оптимальным.

Эксперимент 8.

Эксперимент состоит в применении лучшего метода к тестовой выборке и анализе ошибок алгоритма. Метод, показавший наиболее высокое качество при проведении экспериментов - стохастический градиентный спуск с параметрами $\alpha = 5, \beta = 0, batch\_size = 1000, w_0 = 0$ при выполнении следующей обработки текстовой выборки:

- приведение к нижнему регистру
- удаление всех символов, кроме букв английского алфавита, цифр и пробелов
- лемматизация
- удаление стоп-слов и слов, которые встречаются менее, чем в 5 документах
- преобразование Tfidf коллекции документов

Оптимальный коэффициент регуляризации $\lambda = 0$, данные почти линейно разделимы.

Итоговая точность 89.27% Для повышения качества предсказания алгортма следует проанализировать матрицу ошибок, представленную ниже в таблице 1. Элемент в позиции (i, j) означает количество комментариев класса i, распознанных как класс j.

	Not toxic	Toxic
Not toxic	13094	1339
Toxic	880	5363

Таблица 1: Матрица ошибок предсказания для тестовой выборки.

Рассмотрим комментарии, на которых модель дает ошибку:

1. модель принимает нетоксичный комментарий за токсичный
   (a) ultima linux talk page what the hell
   (b) ...parasite nibbles pieces off its host sucks the host blood...
   (c) n lnkfjjkghdfjgfjgfjhgiofj iojfoijio ioioh....
   (d) 22 hl en num 100 lr lang en ft i cr safe off tbs lr lang 1en fuck you money
2. модель принимает токсичный комментарий за нетоксичный
   (a) you are dead wrong the falcons were 14 2 do your research before you edit something on wikipedia 11
   (b) moi ego i am mortified that you could say such a thing poor old mona i always thiught was a miserable looking woman probably hormonal
   (c) ip address uunet technologies inc uunet1996b net 208 192 0 0 1 208 192 0 0 208 255 255 255

Приведенные комментарии обладают некоторыми общими чертами, как-то: опечатки в словах, употребление слова другого языка или даже быссмысленных комбинаций букв, наличие большого количества цифр. Для первой группы ошибок также стоит отметить, что в комментариях употребляются слова с негативным смыслом: *hell, blood, parasite, fuck* и при этом длина предложения невелика, что объясняет классификацию комменатрия в группу токсичных.

Дополнительное удаление цифр из текста дало повышение точности на 0.01% - точность стала 89.28%. Также было сделано предположение, что регистр может оказывать влияние на токсичность комментария, однако сохранение исходного регистра не дало улучшения качества: точность 89.23%. Применение n-грамм также не показало прироста в точности и даже привело к ее падению: 87.54%

Замечания.

Вычисление градиента функционала потерь $Q(w)$ является основным действием в реализации градиентных методов. В работе для его расчета используется собственная функция $grad$, векторизованно вычисляющая значение $\nabla_w Q$ по ее аналитическому представлению при указанных X, y, w, λ. Таким образом, возникает необходимость проверки корректности ее реализации.

Для проверки правильности подсчета градиента использован метод конечных приращений для вычисления приближенного значения градиента:

$$\frac{\partial Q(w)}{\partial w_i} \approx \frac{Q(w + \varepsilon e_i) - Q(w)}{\varepsilon}, \varepsilon > 0, e_i = (0, \ldots, 1, \ldots, 0).$$

Для проведения тестирования выполнено покрытие области $x, w \in \mathbb{R}^3$, где $x_i, w_i \in [0, 1]$, сеткой с шагом 0.01, $y \in \mathbb{Y} = \{-1, +1\}$, параметр $\lambda \in [0, 1]$ генерируется случайным образом. На каждом наборе значений параметров градиент, подсчитанный методом конечных разностей сравнивается со значением функции $grad$ с заданной величиной погрешности. Реализация приведена в приложениях к отчету.

Вывод.

Проведенные эксперименты показали, что алгоритм логистической регрессии с использованием некоторых эвристик дает хорошую точность для бинарной классификации комментариев: токсичные и не токсичные комментарии оказались почти линейно разделимы.

В ходе работы показано, что при правильно подобранных гиперпараметрах модели метод стохастического градиентного спуска оказывается эффективнее и точнее метода градиентного спуска. Предобработка текстовой коллекции позволяет повысить качество предсказания классификатора и уменьшить время работы алгоритма за счет снижения размерности пространства признаков.